Penerapan Algoritma Arima dalam Harga Saham BBRI

Yudha H Tejaningrat

6 Desember 2023

Contents

Pengantar

Laporan singkat ini merupakan eksplorasi penggunaan analisis runtun waktu dengan algoritma Arima dalam menghitung harga saham BBRI.

Data

Berikut *summary* data yang digunakan dalam penyusunan.

```
summary(BBRI.JK)
```

```
##      Index              BBRI.JK.Open   BBRI.JK.High     BBRI.JK.Low
##  Min.   :2006-12-31   Min.   : 340   Min.   : 412.5   Min.   : 240
```

```
##  1st Qu.:2011-03-07   1st Qu.:1089   1st Qu.:1272.5   1st Qu.:1008
##  Median :2015-05-15   Median :2190   Median :2315.0   Median :2035
##  Mean   :2015-05-16   Mean   :2384   Mean   :2536.7   Mean   :2239
##  3rd Qu.:2019-07-23   3rd Qu.:3705   3rd Qu.:3902.5   3rd Qu.:3463
##  Max.   :2023-09-30   Max.   :5650   Max.   :5750.0   Max.   :5450
##  BBRI.JK.Close  BBRI.JK.Volume      BBRI.JK.Adjusted
##  Min.   : 340   Min.   :1.081e+09   Min.   : 225.2
##  1st Qu.:1142   1st Qu.:2.374e+09   1st Qu.: 800.1
##  Median :2195   Median :2.872e+09   Median :1665.4
##  Mean   :2397   Mean   :3.094e+09   Mean   :1996.5
##  3rd Qu.:3690   3rd Qu.:3.595e+09   3rd Qu.:3093.8
##  Max.   :5650   Max.   :7.239e+09   Max.   :5650.0
```

```r
head(BBRI.JK, n=5)
```

```
##            BBRI.JK.Open BBRI.JK.High BBRI.JK.Low BBRI.JK.Close BBRI.JK.Volume
## 2006-12-31        515.0          545         450           530     2866205000
## 2007-01-31        520.0          530         440           475     3298620000
## 2007-02-28        482.5          515         470           505     2408000000
## 2007-03-31        520.0          565         505           525     2941200000
## 2007-04-30        530.0          630         520           610     2404560000
##            BBRI.JK.Adjusted
## 2006-12-31         329.9703
## 2007-01-31         295.7281
## 2007-02-28         314.4057
## 2007-03-31         326.8574
## 2007-04-30         379.7772
```

```r
dataset = Cl(BBRI.JK) %>% as.data.frame()
dataset_close = data.frame(tanggal=as.Date(row.names(dataset)),
                           tutup=dataset[,1])
head(dataset_close)
```

```
##      tanggal tutup
## 1 2006-12-31   530
## 2 2007-01-31   475
## 3 2007-02-28   505
## 4 2007-03-31   525
## 5 2007-04-30   610
## 6 2007-05-31   575
```

```r
training = as_tbl_time(dataset_close, index = tanggal) %>%   filter_time('2007' ~ '2022')
training
```

```
## # A time tibble: 192 x 2
## # Index:         tanggal
##    tanggal    tutup
##    <date>     <dbl>
##  1 2007-01-31   475
##  2 2007-02-28   505
##  3 2007-03-31   525
##  4 2007-04-30   610
##  5 2007-05-31   575
##  6 2007-06-30   630
##  7 2007-07-31   625
##  8 2007-08-31   660
```

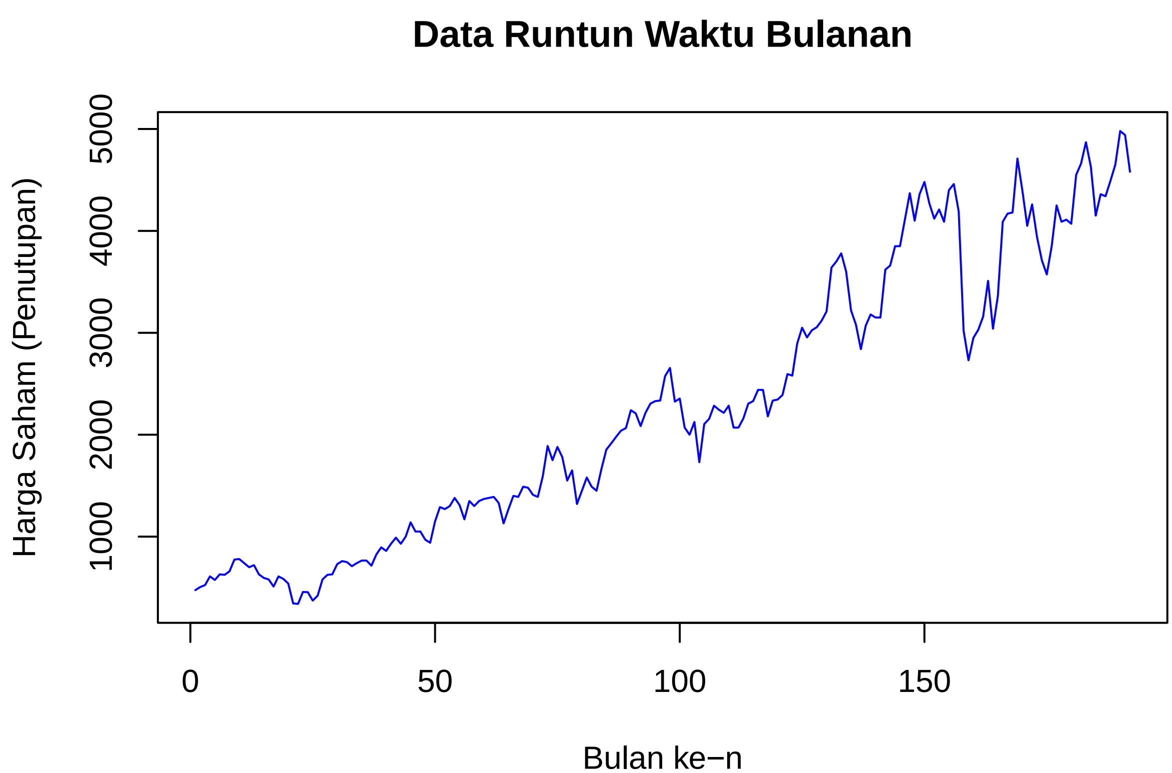
```
## 9 2007-09-30   775
## 10 2007-10-31   780
## # i 182 more rows
```

```
testing = as_tbl_time(dataset_close, index = tanggal) %>%   filter_time('2023' ~ '2023')
testing
```

```
## # A time tibble: 9 x 2
## # Index:         tanggal
##   tanggal    tutup
##   <date>     <dbl>
## 1 2023-01-31  4670
## 2 2023-02-28  4730
## 3 2023-03-31  5100
## 4 2023-04-30  5575
## 5 2023-05-31  5425
## 6 2023-06-30  5650
## 7 2023-07-31  5550
## 8 2023-08-31  5225
## 9 2023-09-30  4960
```

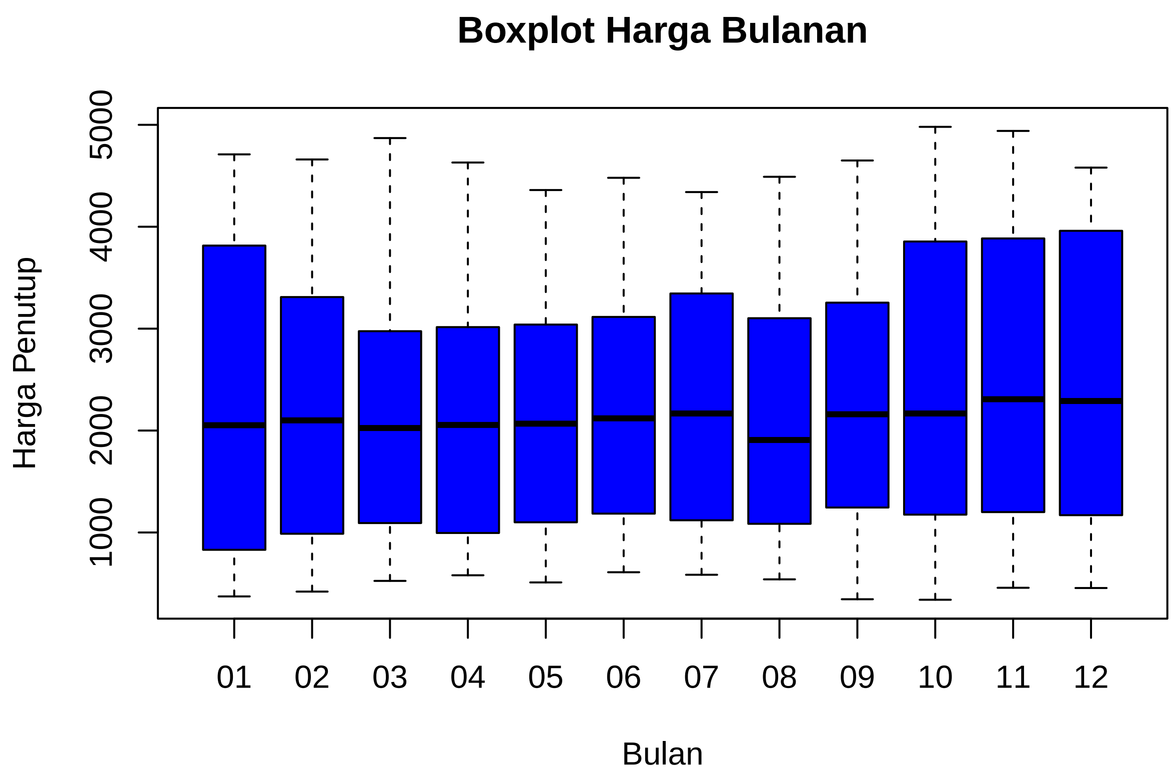
Dan berikut diagram *plot* dari data tersebut.

```
ts.plot(training$tutup, col="blue", ylab="Harga Saham (Penutupan)", xlab="Bulan ke-n",
        main="Data Runtun Waktu Bulanan")
```

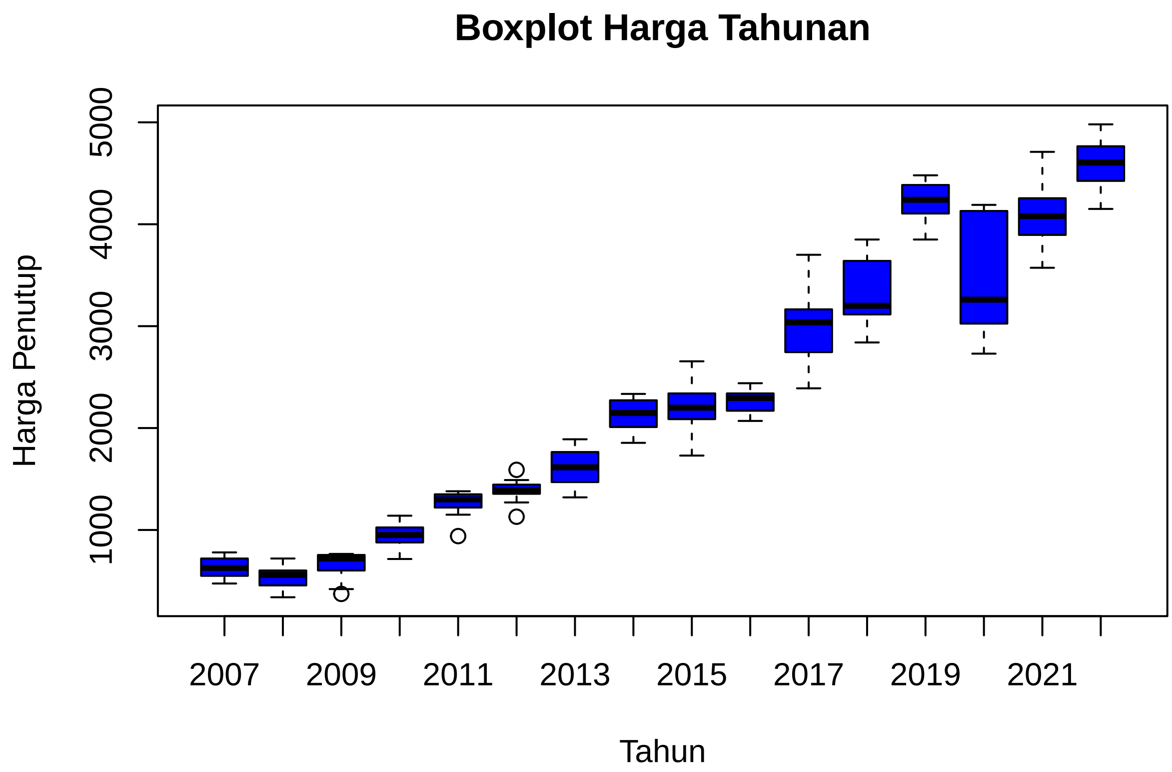



```
# Ubah tanggal menjadi bulan
training$bulan <- format(training$tanggal, "%m")
```

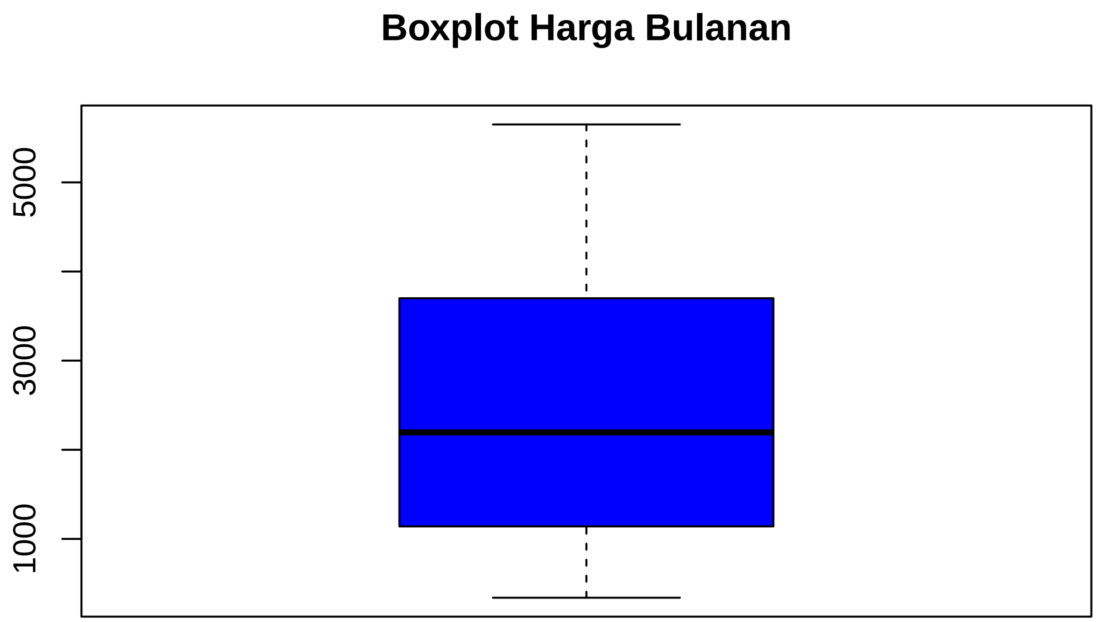
```r
# Ubah tanggal menjadi tahun
training$tahun <- format(training$tanggal, "%Y")
# Buat boxplot bulanan
boxplot(tutup ~ bulan, data=training, col="blue",
        main="Boxplot Harga Bulanan", xlab="Bulan", ylab="Harga Penutup")
```




```r
# Buat boxplot tahunan
boxplot(tutup ~ tahun, data=training, col="blue",
        main="Boxplot Harga Tahunan", xlab="Tahun", ylab="Harga Penutup")
```




```r
boxplot(dataset_close$tutup, col="blue", title=FALSE, cex=0.5, pch=19,
        main="Boxplot Harga Bulanan")
```



Pengujian terhadap Data

Sebelum dilakukan estimasi, dilakukan beberapa pengujian terhadap data.

Dickey-Fuller Unit-Root Test

Berikut hasil pengujian menggunakan Augmented Dickey-Fuller dengan fungsi **adf.test** dari paket **tseries**.

```
adf.test(training$tutup)
```

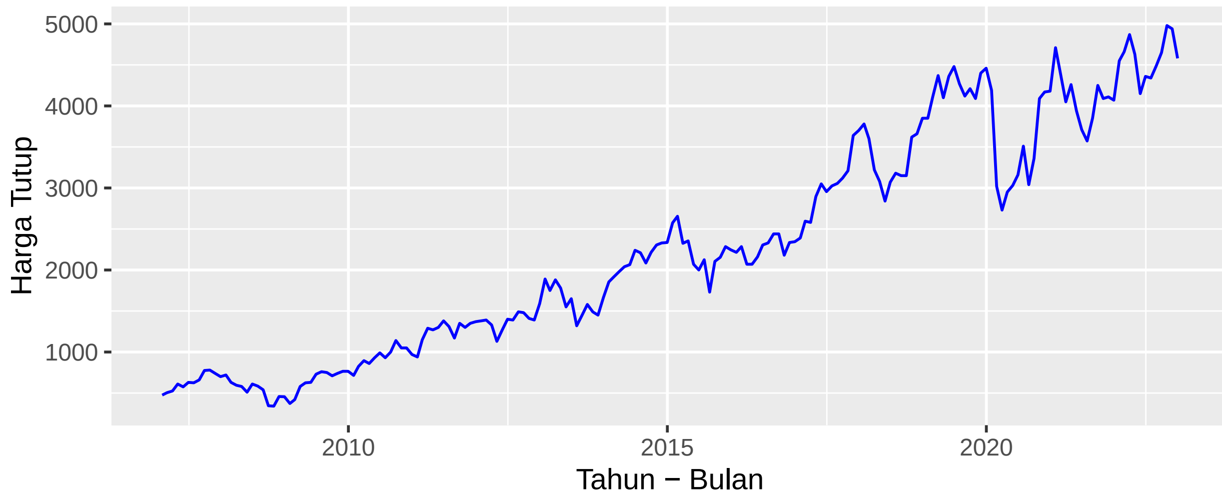
```
##
##  Augmented Dickey-Fuller Test
##
## data:  training$tutup
## Dickey-Fuller = -4.0048, Lag order = 5, p-value = 0.01024
## alternative hypothesis: stationary
```

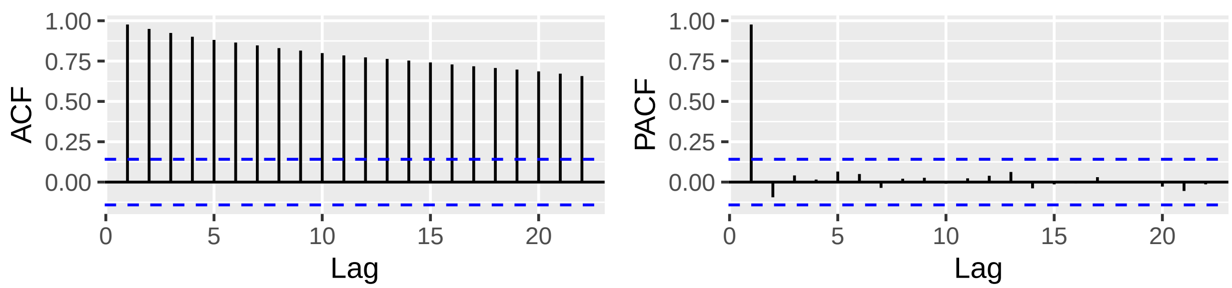
Uji Autokolerasi

Berikut hasil pengujian menggunakan Autokorelasi menggunakan fungsi **Acf** dan **Pacf** dari paket **forecast**.

```
# Plot time series using ggplot2
ts_plot <- ggplot(training, aes(x = tanggal, y = tutup)) +
  geom_line(color = "blue") +
  labs(x = "Tahun - Bulan", y = "Harga Tutup")
acf_plot <- ggAcf(training$tutup) + labs(title="")
```

```
pacf_plot <- ggPacf(training$tutup) + labs(title="")
grid.arrange(ts_plot, grid.arrange(acf_plot, pacf_plot, ncol = 2, heights = 1), ncol = 1, heights = c(3
```





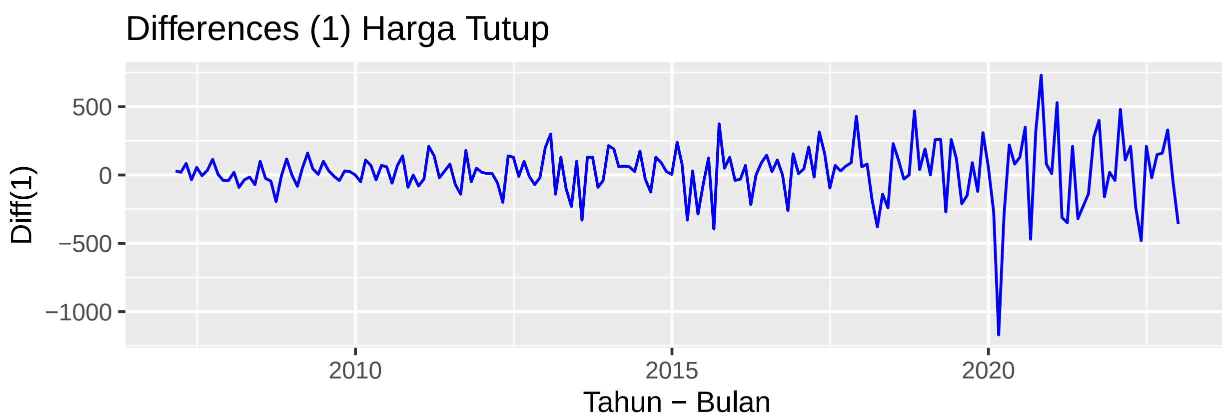
Dalam contoh ini, kita menggunakan dataset yang dihasilkan secara acak dan mengidentifikasi cut off pada ACF. Kami kemudian menentukan ordo MA (q) berdasarkan lag tempat cut off tersebut terjadi.

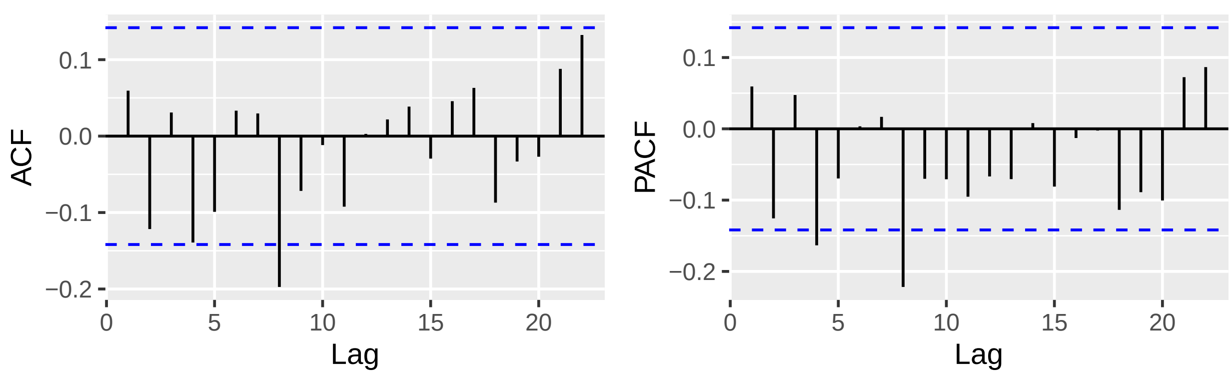
Pastikan untuk mengganti dataset dengan data deret waktu yang sesuai dengan analisis Anda.

Differences

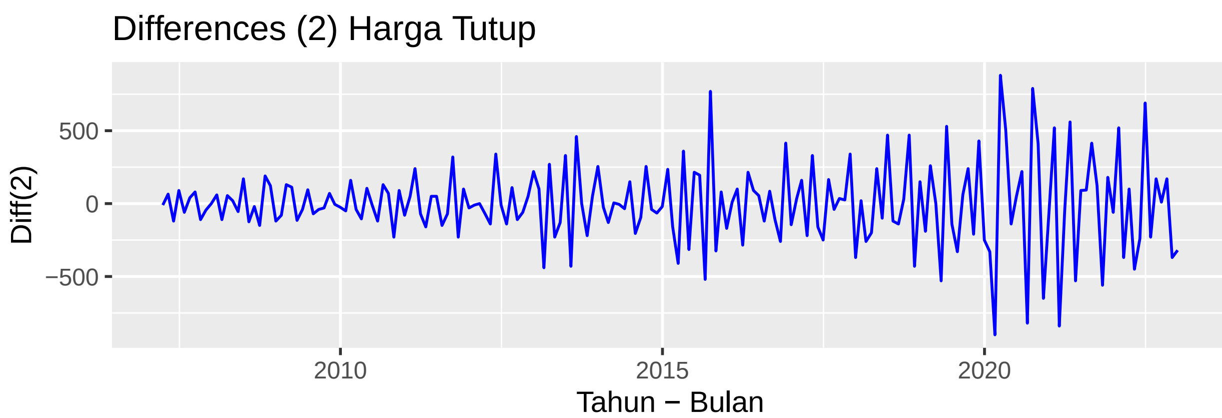
Berikut diagram transformasi *differences* menggunakan fungsi **diff** dari data dan log natural data.

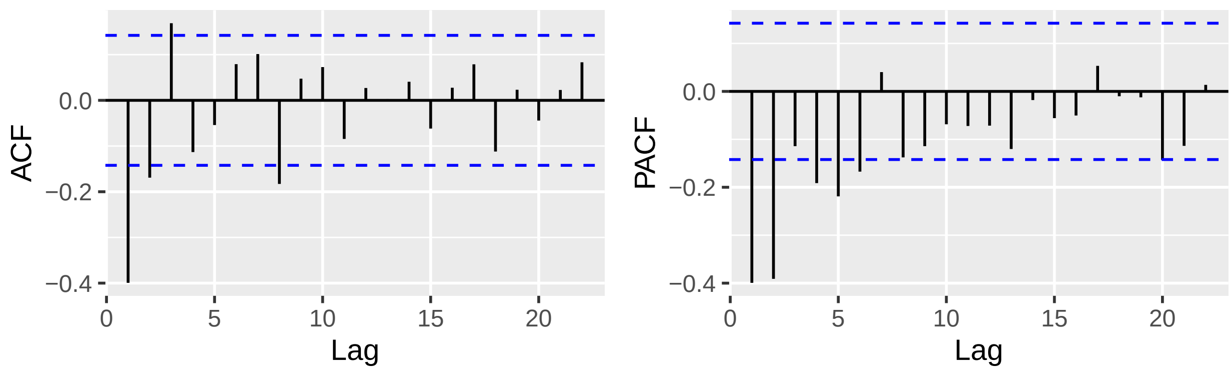
```
monthly_diff = data.frame(tanggal=as.Date(training$tanggal[2:nrow(training)]),
                          tutup=as.numeric(diff(training$tutup, differences = 1)))
# Plot time series using ggplot2
ts_plot <- ggplot(monthly_diff, aes(x = tanggal, y = tutup)) +
  geom_line(color = "blue") +
  labs(title="Differences (1) Harga Tutup", x="Tahun - Bulan", y="Diff(1)")
acf_plot <- ggAcf(monthly_diff$tutup) + labs(title="")
pacf_plot <- ggPacf(monthly_diff$tutup) + labs(title="")
grid.arrange(ts_plot, grid.arrange(acf_plot, pacf_plot, ncol = 2, heights = 1), ncol = 1, heights = c(3
```






```
monthly_diff2 = data.frame(tanggal=as.Date(training$tanggal[3:nrow(training)]),
                          tutup=as.numeric(diff(training$tutup, differences = 2)))
# Plot time series using ggplot2
ts_plot <- ggplot(monthly_diff2, aes(x = tanggal, y = tutup)) +
  geom_line(color = "blue") +
  labs(title="Differences (2) Harga Tutup", x="Tahun - Bulan", y="Diff(2)")
acf_plot <- ggAcf(monthly_diff2$tutup) + labs(title="")
pacf_plot <- ggPacf(monthly_diff2$tutup) + labs(title="")
grid.arrange(ts_plot, grid.arrange(acf_plot, pacf_plot, ncol = 2, heights = 1), ncol = 1, heights = c(3
```

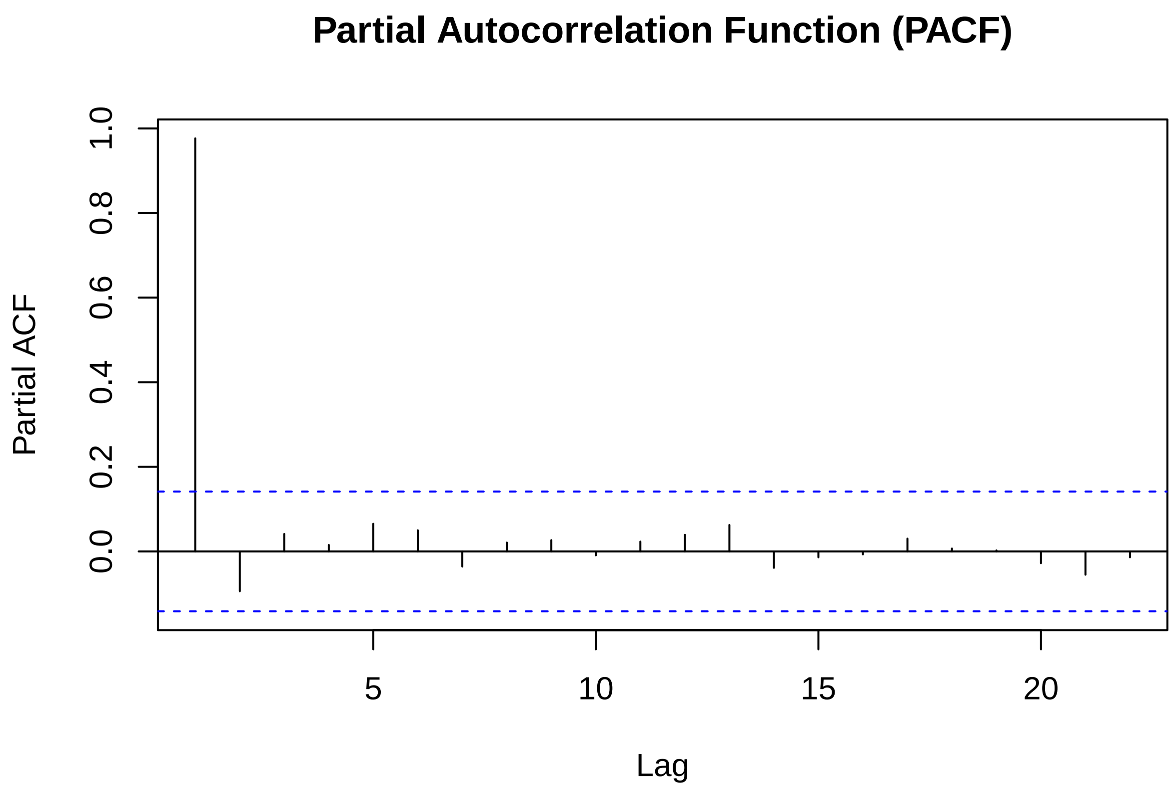




Identifikasi ARMA

Tanpa Differencing

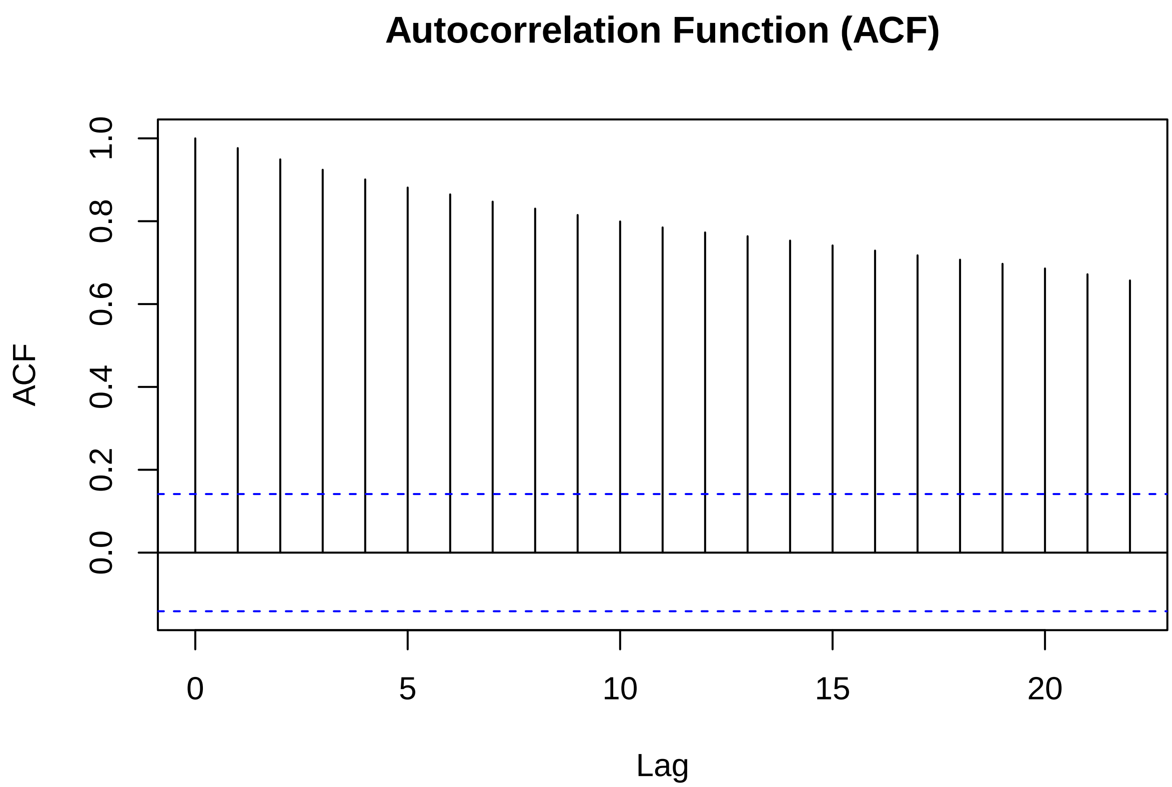
```
# Plot PACF
pacf_plot <- pacf(training$tutup, main = "Partial Autocorrelation Function (PACF)")
```




```r
# Tentukan cut off
cut_off <- 2 / sqrt(length(dataset))
# Identifikasi lag pertama di mana PACF melewati cut off
lag_with_cut_off <- which(abs(pacf_plot$acf) < cut_off)[1]
# Tentukan ordo AR (p) berdasarkan lag dengan cut off
order_ar <- lag_with_cut_off - 1
#cat("Order of AR (p):", order_ar, "\n")
```

Karena pada PACF *cut-off* pada 2 dan *lag* adalah 1, maka ordo untuk AR adalah 0.

```r
# Plot ACF
acf_plot <- acf(training$tutup, main = "Autocorrelation Function (ACF)")
```



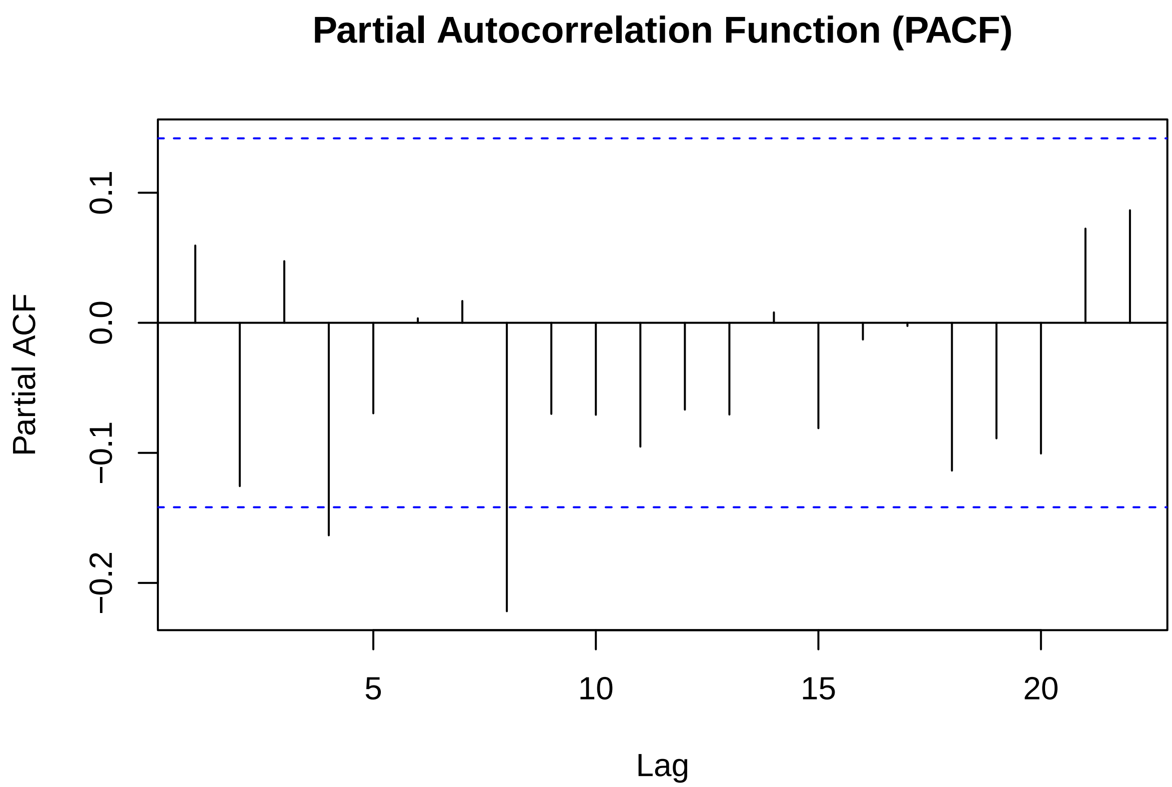

```r
# Tentukan cut off
cut_off <- 2 / sqrt(length(training))
# Identifikasi lag pertama di mana ACF melewati cut off
lag_with_cut_off <- which(abs(acf_plot$acf) < cut_off)[1]
# Tentukan ordo MA (q) berdasarkan lag dengan cut off
order_ma <- lag_with_cut_off - 1
#cat("Order of MA (q):", order_ma, "\n")
```

Karena pada ACF *cut-off* pada 1 dan *lag* adalah 2, maka ordo untuk MA adalah 1.

Sehingga saran model ARIMA adalah ARIMA(0 , 0, 1).

Differencing 1

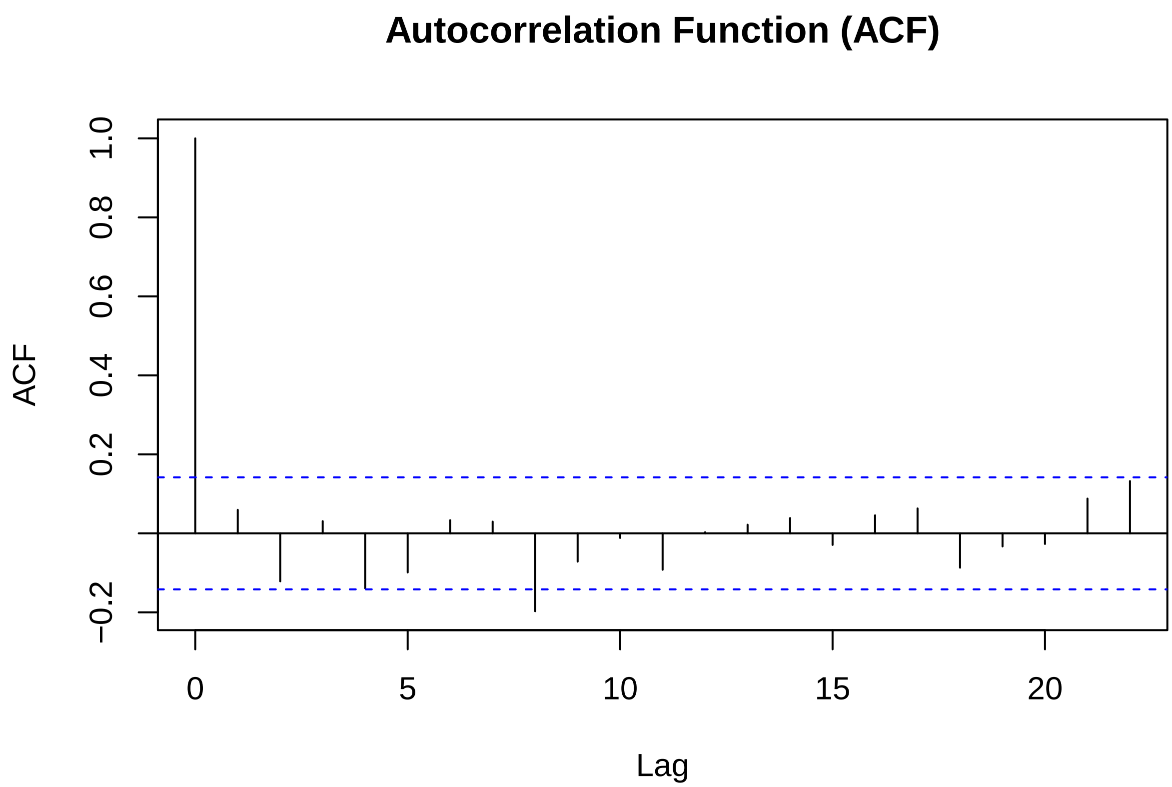
```r
# Plot PACF
pacf_plot <- pacf(monthly_diff$tutup, main = "Partial Autocorrelation Function (PACF)")
```




```
# Tentukan cut off
cut_off <- 2 / sqrt(length(monthly_diff))
# Identifikasi lag pertama di mana PACF melewati cut off
lag_with_cut_off <- which(abs(pacf_plot$acf) < cut_off)[1]
# Tentukan ordo AR (p) berdasarkan lag dengan cut off
order_ar <- lag_with_cut_off - 1
#cat("Order of AR (p):", order_ar, "\n")
```

Karena pada PACF *cut-off* pada 1.4142136 dan *lag* adalah 1, maka ordo untuk AR adalah 0.

```
# Plot ACF
acf_plot <- acf(monthly_diff$tutup, main = "Autocorrelation Function (ACF)")
```



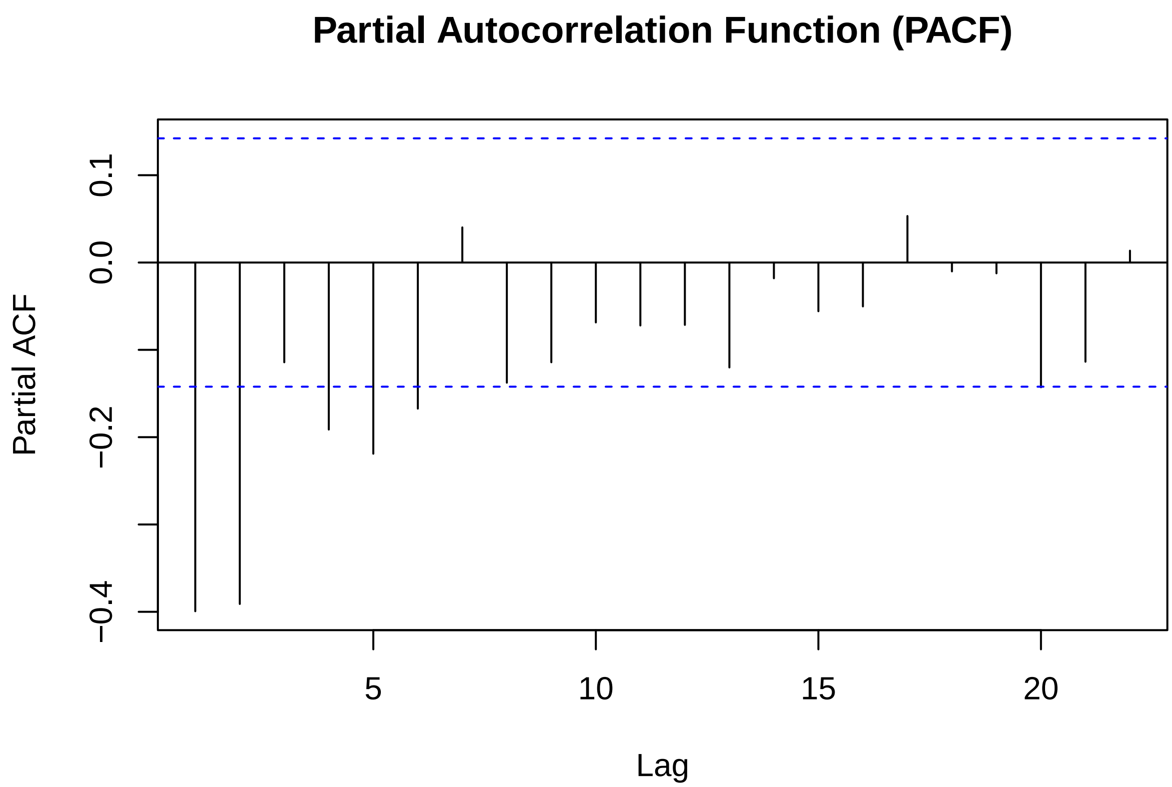

```r
# Tentukan cut off
cut_off <- 2 / sqrt(length(monthly_diff))
# Identifikasi lag pertama di mana ACF melewati cut off
lag_with_cut_off <- which(abs(acf_plot$acf) < cut_off)[1]
# Tentukan ordo MA (q) berdasarkan lag dengan cut off
order_ma <- lag_with_cut_off - 1
#cat("Order of MA (q):", order_ma, "\n")
```

Karena pada ACF *cut-off* pada 1.4142136 dan *lag* adalah 1, maka ordo untuk MA adalah 0.

Sehingga saran model ARIMA adalah ARIMA(0 , 1, 0).

Differencing 2

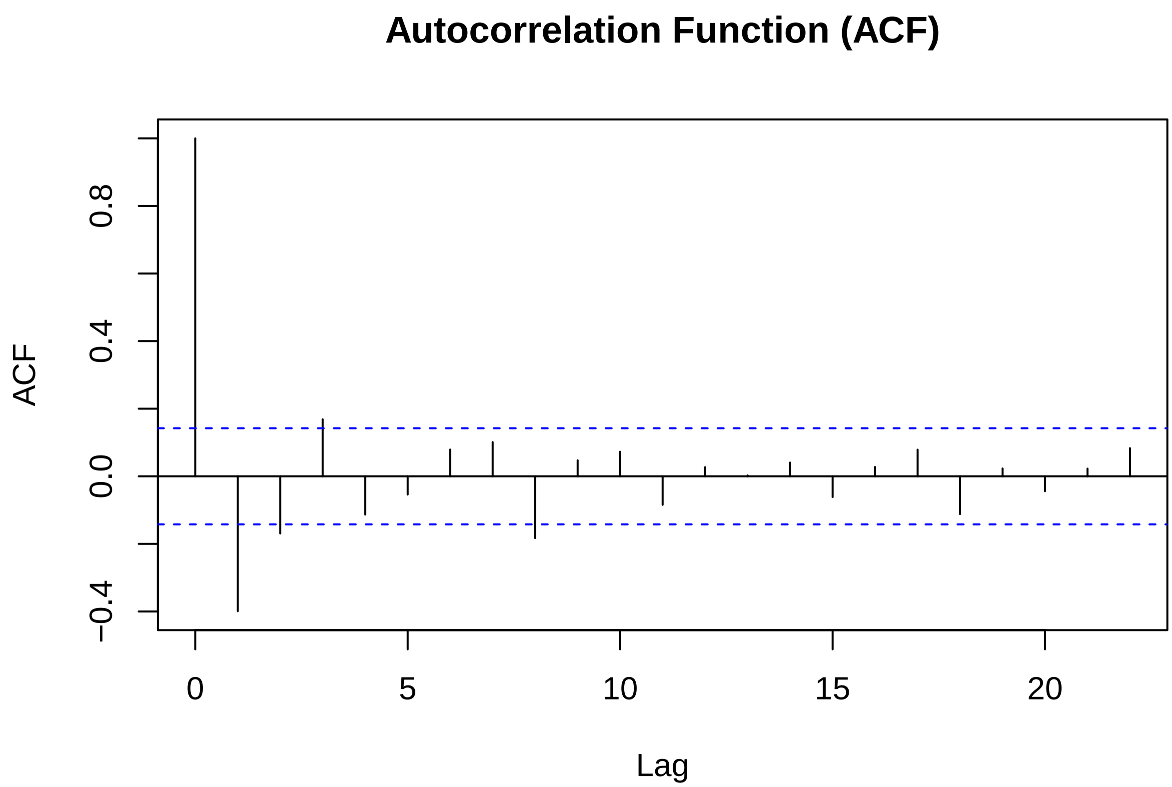
```r
# Plot PACF
pacf_plot <- pacf(monthly_diff2$tutup, main = "Partial Autocorrelation Function (PACF)")
```




```
# Tentukan cut off
cut_off <- 2 / sqrt(length(monthly_diff2))
# Identifikasi lag pertama di mana PACF melewati cut off
lag_with_cut_off <- which(abs(pacf_plot$acf) < cut_off)[1]
# Tentukan ordo AR (p) berdasarkan lag dengan cut off
order_ar <- lag_with_cut_off - 1
#cat("Order of AR (p):", order_ar, "\n")
```

Karena pada PACF *cut-off* pada 1.4142136 dan *lag* adalah 1, maka ordo untuk AR adalah 0.

```
# Plot ACF
acf_plot <- acf(monthly_diff2$tutup, main = "Autocorrelation Function (ACF)")
```




```
# Tentukan cut off
cut_off <- 2 / sqrt(length(monthly_diff2))
# Identifikasi lag pertama di mana ACF melewati cut off
lag_with_cut_off <- which(abs(acf_plot$acf) < cut_off)[1]
# Tentukan ordo MA (q) berdasarkan lag dengan cut off
order_ma <- lag_with_cut_off - 1
#cat("Order of MA (q):", order_ma, "\n")
```

Karena pada ACF *cut-off* pada 1.4142136 dan *lag* adalah 1, maka ordo untuk MA adalah 0.

Sehingga saran model ARIMA adalah ARIMA(0 , 2, 0).

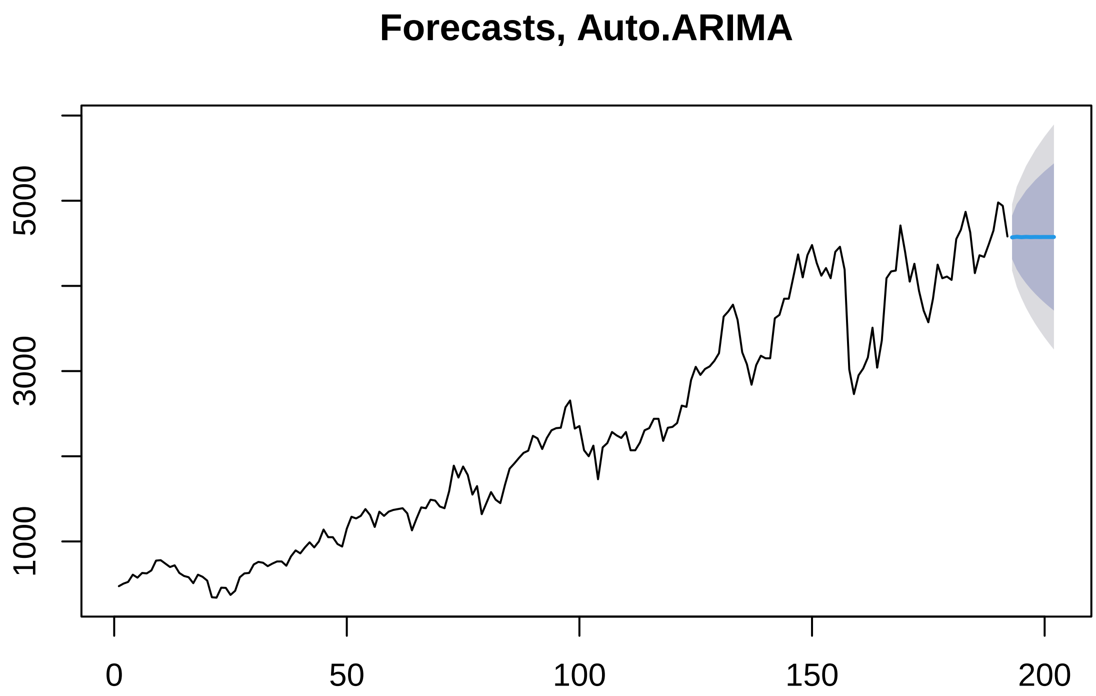
Estimasi menggunakan Arima

Berikut data akhir yang akan di test.

```
actual <- testing
model_cv <- data.frame(model=character(), aic=numeric(), mape_test=numeric(),
                       mae_test=numeric(), rmse_test=numeric())
```

Auto.ARIMA

```
mfitauto <- auto.arima(training$tutup, seasonal = FALSE, allowdrift = FALSE,
                       max.p = 10, max.d = 3, max.q = 10)
mfcastauto <- forecast(mfitauto, h=10)
plot(mfcastauto, main="Forecasts, Auto.ARIMA")
```




```r
summary(mfitauto)
```

```
## Series: training$tutup
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1     ma1
##       -0.6872  0.8134
## s.e.   0.1657  0.1311
##
## sigma^2 = 39628:  log likelihood = -1281.14
## AIC=2568.28   AICc=2568.41   BIC=2578.04
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE        ACF1
## Training set 19.86473 197.5066 134.8988 0.698043 6.728314 0.975246 -0.03762037
```

```r
summary(mfcastauto)
```

```
##
## Forecast method: ARIMA(1,1,1)
##
## Model Information:
## Series: training$tutup
## ARIMA(1,1,1)
##
## Coefficients:
```

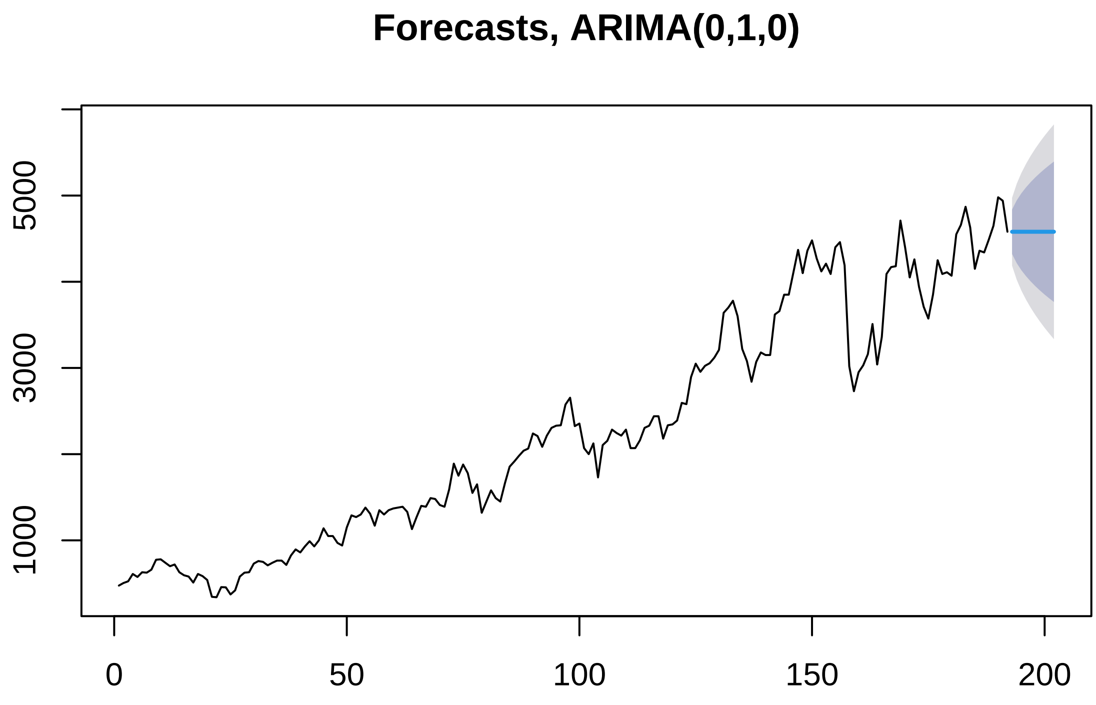
```
##           ar1     ma1
##       -0.6872  0.8134
## s.e.   0.1657  0.1311
##
## sigma^2 = 39628:  log likelihood = -1281.14
## AIC=2568.28   AICc=2568.41   BIC=2578.04
##
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE         ACF1
## Training set 19.86473 197.5066 134.8988 0.698043 6.728314 0.975246 -0.03762037
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 193       4569.875 4314.759 4824.991 4179.709 4960.041
## 194       4576.833 4192.599 4961.067 3989.197 5164.469
## 195       4572.051 4105.188 5038.914 3858.046 5286.057
## 196       4575.337 4030.743 5119.932 3742.452 5408.223
## 197       4573.079 3965.252 5180.906 3643.488 5502.670
## 198       4574.631 3906.609 5242.653 3552.980 5596.282
## 199       4573.565 3852.217 5294.912 3470.359 5676.770
## 200       4574.297 3802.101 5346.494 3393.324 5755.271
## 201       4573.794 3754.675 5392.912 3321.060 5826.527
## 202       4574.140 3710.138 5438.142 3252.762 5895.518
```

```r
m1 = data.frame(model="auto.arima / arima(1,1,1)", aic=mfitauto$aic,
                mape_test=mape(mfcastauto$mean, actual$tutup),
                mae_test=mae(mfcastauto$mean, actual$tutup),
                rmse_test = rmse(mfcastauto$mean, actual$tutup))
print(t(m1[,2:5]))
```

```
##                   [,1]
## aic       2568.2823381
## mape_test    0.1271844
## mae_test   581.7397324
## rmse_test  687.6708607
```

ARIMA (0,1,0)

```r
mfit010 <- arima(training$tutup, order=c(0,1,0))
mfcast010 <- forecast(mfit010, h=10)
plot(mfcast010, main="Forecasts, ARIMA(0,1,0)")
```




```r
summary(mfit010)
```

```
##
## Call:
## arima(x = training$tutup, order = c(0, 1, 0))
##
##
## sigma^2 estimated as 40419:  log likelihood = -1283.99,  aic = 2569.98
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE    MAPE      MASE       ACF1
## Training set 21.38268 200.5205 137.6049 0.7362257 6.80796 0.9948096 0.05939905
```

```r
summary(mfcast010)
```

```
##
## Forecast method: ARIMA(0,1,0)
##
## Model Information:
##
## Call:
## arima(x = training$tutup, order = c(0, 1, 0))
##
##
## sigma^2 estimated as 40419:  log likelihood = -1283.99,  aic = 2569.98
##
## Error measures:
```

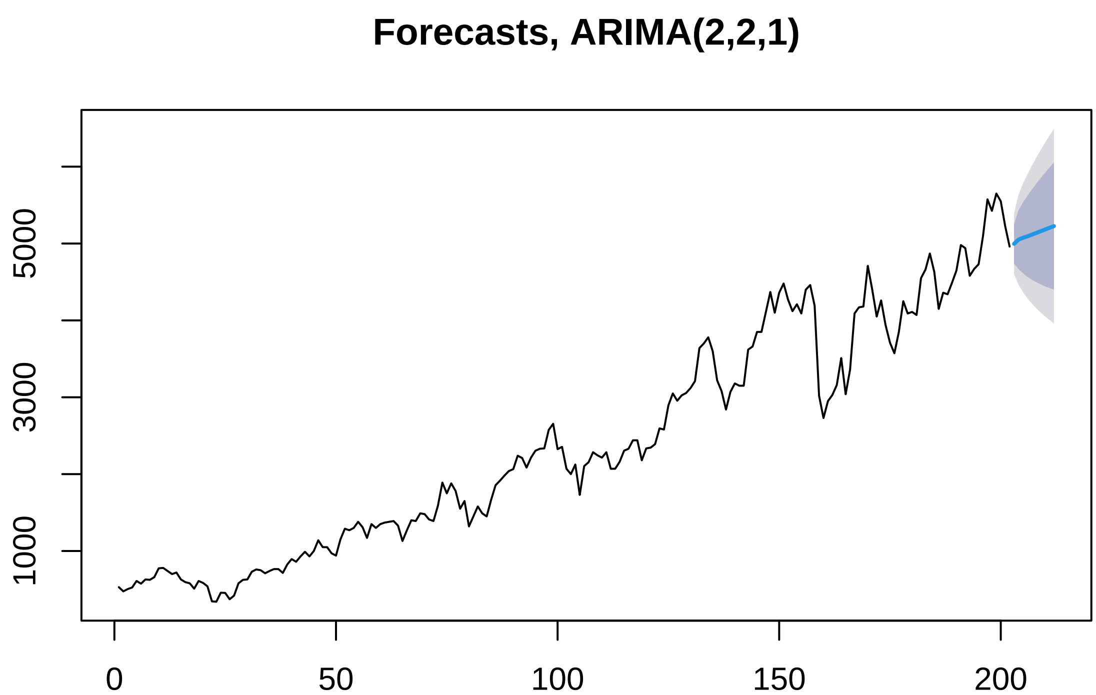
```
##                    ME     RMSE      MAE       MPE    MAPE      MASE       ACF1
## Training set 21.38268 200.5205 137.6049 0.7362257 6.80796 0.9948096 0.05939905
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 193           4580 4322.351 4837.649 4185.960 4974.040
## 194           4580 4215.629 4944.371 4022.743 5137.257
## 195           4580 4133.739 5026.261 3897.502 5262.498
## 196           4580 4064.702 5095.298 3791.919 5368.081
## 197           4580 4003.879 5156.121 3698.899 5461.101
## 198           4580 3948.891 5211.109 3614.802 5545.198
## 199           4580 3898.324 5261.676 3537.467 5622.533
## 200           4580 3851.258 5308.742 3465.485 5694.515
## 201           4580 3807.052 5352.948 3397.879 5762.121
## 202           4580 3765.242 5394.758 3333.935 5826.065
```

```
m2 = data.frame(model="arima(0,1,0)", aic=mfit010$aic,
                mape_test=mape(mfcast010$mean, actual$tutup),
                mae_test=mae(mfcast010$mean, actual$tutup),
                rmse_test = rmse(mfcast010$mean, actual$tutup))
print(t(m2[,2:5]))
```

```
##                   [,1]
## aic        2569.981978
## mape_test     0.125655
## mae_test    575.500000
## rmse_test   682.596147
```

ARIMA (2,2,1)

```
mfit221 <- arima(dataset_close$tutup, order=c(2,2,1))
mfcast221 <- forecast(mfit221, h=10)
plot(mfcast221, main="Forecasts, ARIMA(2,2,1)")
```




```r
summary(mfit221)
```

```
##
## Call:
## arima(x = dataset_close$tutup, order = c(2, 2, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.0853  -0.1098  -1.0000
## s.e.  0.0706   0.0709   0.0139
##
## sigma^2 estimated as 40537:  log likelihood = -1347.48,  aic = 2702.95
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 7.377299 200.3394 138.5992 0.07350105 6.631401 0.9763014
##                      ACF1
## Training set -0.001192547
```

```r
summary(mfcast221)
```

```
##
## Forecast method: ARIMA(2,2,1)
##
## Model Information:
##
## Call:
```

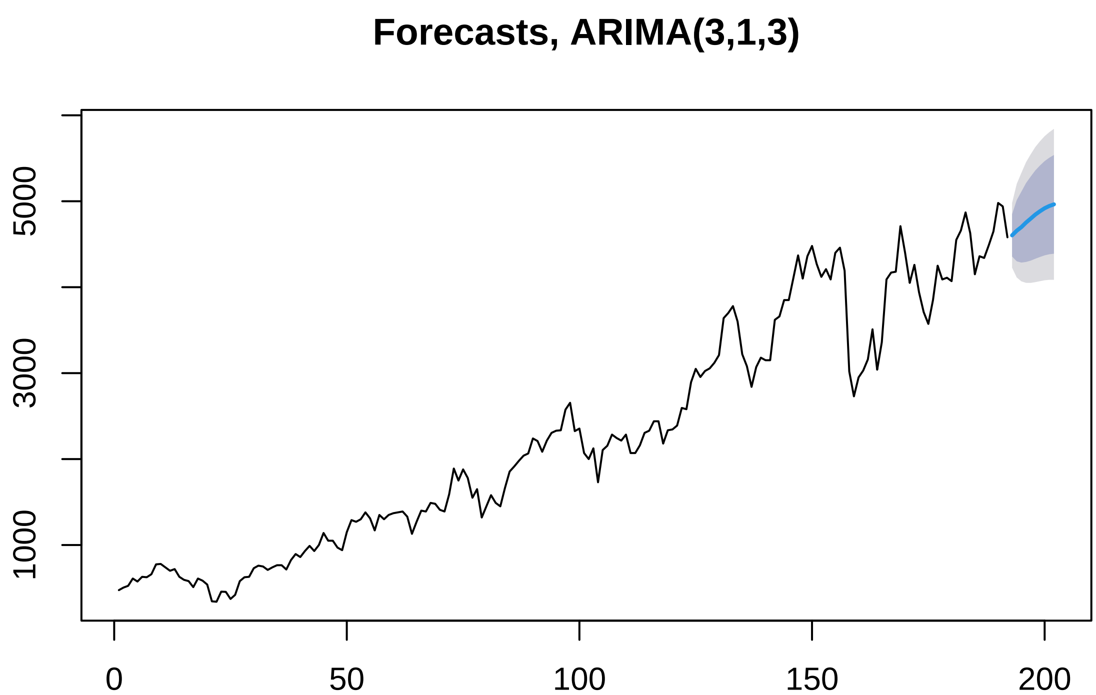
```
## arima(x = dataset_close$tutup, order = c(2, 2, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.0853  -0.1098  -1.0000
## s.e.  0.0706   0.0709   0.0139
##
## sigma^2 estimated as 40537:  log likelihood = -1347.48,  aic = 2702.95
##
## Error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 7.377299 200.3394 138.5992 0.07350105 6.631401 0.9763014
##                      ACF1
## Training set -0.001192547
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 203       4995.881 4737.213 5254.549 4600.283 5391.479
## 204       5050.834 4668.165 5433.503 4465.593 5636.076
## 205       5074.391 4613.500 5535.283 4369.518 5779.264
## 206       5093.178 4567.201 5619.156 4288.765 5897.591
## 207       5115.004 4529.524 5700.483 4219.591 6010.417
## 208       5137.612 4497.357 5777.867 4158.427 6116.797
## 209       5159.954 4468.876 5851.031 4103.041 6216.866
## 210       5182.187 4443.382 5920.991 4052.283 6312.090
## 211       5204.440 4420.402 5988.477 4005.357 6403.522
## 212       5226.706 4399.520 6053.892 3961.634 6491.778
```

```r
m3 = data.frame(model="arima(2,2,1)", aic=mfit221$aic,
                mape_test=mape(mfcast221$mean, actual$tutup),
                mae_test=mae(mfcast221$mean, actual$tutup),
                rmse_test = rmse(mfcast221$mean, actual$tutup))
print(t(m3[,2:5]))
```

```
##                    [,1]
## aic        2.702951e+03
## mape_test  6.260691e-02
## mae_test   3.210536e+02
## rmse_test  3.638986e+02
```

ARIMA (3,1,3)

```r
mfit313 <- arima(training$tutup, order=c(3,1,3))
mfcast313 <- forecast(mfit313, h=10)
plot(mfcast313, main="Forecasts, ARIMA(3,1,3)")
```




```r
summary(mfit313)
```

```
##
## Call:
## arima(x = training$tutup, order = c(3, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     ma3
##       1.2053  0.3181  -0.5986  -1.1688  -0.5682  0.7975
## s.e.  0.1716  0.3241   0.1659   0.1400   0.2701  0.1387
##
## sigma^2 estimated as 36528:  log likelihood = -1276.35,  aic = 2566.69
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 28.08433 190.6243 132.9419 0.9751304 6.619854 0.9610982
##                     ACF1
## Training set -0.03061251
```

```r
summary(mfcast313)
```

```
##
## Forecast method: ARIMA(3,1,3)
##
## Model Information:
##
## Call:
```

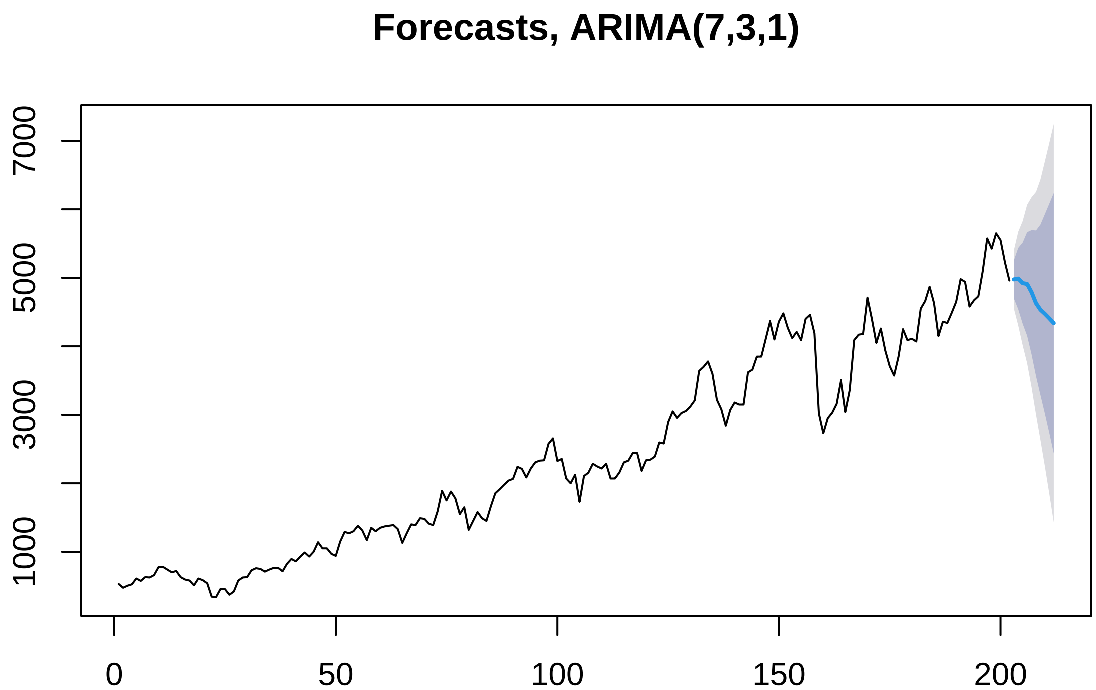
```
## arima(x = training$tutup, order = c(3, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     ma3
##       1.2053  0.3181  -0.5986  -1.1688  -0.5682  0.7975
## s.e.  0.1716  0.3241   0.1659   0.1400   0.2701  0.1387
##
## sigma^2 estimated as 36528:  log likelihood = -1276.35,  aic = 2566.69
##
## Error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 28.08433 190.6243 132.9419 0.9751304 6.619854 0.9610982
##                     ACF1
## Training set -0.03061251
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 193       4603.103 4357.054 4849.152 4226.804 4979.402
## 194       4657.193 4301.232 5013.153 4112.798 5201.588
## 195       4698.875 4286.258 5111.493 4067.831 5329.920
## 196       4752.493 4293.581 5211.405 4050.648 5454.339
## 197       4798.002 4308.861 5287.142 4049.926 5546.077
## 198       4844.959 4330.990 5358.928 4058.911 5631.007
## 199       4883.938 4351.608 5416.268 4069.809 5698.067
## 200       4918.617 4370.469 5466.764 4080.298 5756.936
## 201       4944.706 4383.145 5506.268 4085.873 5803.540
## 202       4963.851 4389.503 5538.200 4085.461 5842.242
```

```
m4 = data.frame(model="arima(3,1,3)", aic=mfit313$aic,
                mape_test=mape(mfcast313$mean, actual$tutup),
                mae_test=mae(mfcast313$mean, actual$tutup),
                rmse_test = rmse(mfcast313$mean, actual$tutup))
print(t(m4[,2:5]))
```

```
##                   [,1]
## aic        2.566693e+03
## mape_test 8.464003e-02
## mae_test  4.076965e+02
## rmse_test 5.012489e+02
```

ARIMA (7,3,1)

```
mfit731 <- arima(dataset_close$tutup, order=c(7,3,1))
mfcast731 <- forecast(mfit731, h=10)
plot(mfcast731, main="Forecasts, ARIMA(7,3,1)")
```




```r
summary(mfit731)
```

```
##
## Call:
## arima(x = dataset_close$tutup, order = c(7, 3, 1))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6     ar7      ma1
##       -0.7232  -0.6778  -0.4411  -0.4813  -0.3549  -0.1864  0.0265  -1.0000
## s.e.   0.0717   0.0887   0.0997   0.0980   0.1000   0.0895  0.0724   0.0133
##
## sigma^2 estimated as 45960:  log likelihood = -1355.18,  aic = 2728.36
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE    MAPE     MASE
## Training set -3.705903 212.7842 146.7055 -0.3483671 7.10977 1.033403
##                      ACF1
## Training set -0.004385079
```

```r
summary(mfcast731)
```

```
##
## Forecast method: ARIMA(7,3,1)
##
## Model Information:
##
## Call:
```

```
## arima(x = dataset_close$tutup, order = c(7, 3, 1))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6     ar7      ma1
##       -0.7232  -0.6778  -0.4411  -0.4813  -0.3549  -0.1864  0.0265  -1.0000
## s.e.   0.0717   0.0887   0.0997   0.0980   0.1000   0.0895  0.0724   0.0133
##
## sigma^2 estimated as 45960:  log likelihood = -1355.18,  aic = 2728.36
##
## Error measures:
##                     ME     RMSE      MAE        MPE    MAPE     MASE
## Training set -3.705903 212.7842 146.7055 -0.3483671 7.10977 1.033403
##                      ACF1
## Training set -0.004385079
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 203       4975.824 4700.382 5251.265 4554.572 5397.075
## 204       4988.082 4540.281 5435.882 4303.230 5672.933
## 205       4921.596 4328.783 5514.408 4014.967 5828.224
## 206       4910.482 4154.972 5665.993 3755.028 6065.936
## 207       4789.486 3883.077 5695.895 3403.253 6175.719
## 208       4628.398 3565.668 5691.129 3003.092 6253.705
## 209       4533.113 3289.270 5776.955 2630.820 6435.405
## 210       4473.646 3017.991 5929.302 2247.413 6699.879
## 211       4407.520 2732.431 6082.609 1845.692 6969.348
## 212       4337.138 2436.602 6237.674 1430.519 7243.757
```

```r
m5 = data.frame(model="arima(7,3,1)", aic=mfit731$aic,
                mape_test=mape(mfcast731$mean, actual$tutup),
                mae_test=mae(mfcast731$mean, actual$tutup),
                rmse_test = rmse(mfcast731$mean, actual$tutup))
print(t(m5[,2:5]))
```

```
##                  [,1]
## aic       2728.3620627
## mape_test    0.1232563
## mae_test   571.7527285
## rmse_test  639.9780550
```

Uji Model ARIMA Tambahan

```r
mfit211 <- arima(training$tutup, order=c(2,1,1))
mfcast211 <- forecast(mfit211, h=10)
m6 = data.frame(model="arima(2,1,1)", aic=mfit211$aic,
                mape_test=mape(mfcast211$mean, actual$tutup),
                mae_test=mae(mfcast211$mean, actual$tutup),
                rmse_test = rmse(mfcast211$mean, actual$tutup))
print(t(m6[,2:5]))
```

```
##                  [,1]
## aic       2569.8221520
## mape_test    0.1216543
## mae_test   559.2124407
## rmse_test  668.5353846
```

```r
mfit212 <- arima(training$tutup, order=c(2,1,2))
mfcast212 <- forecast(mfit212, h=10)
m7 = data.frame(model="arima(2,1,2)", aic=mfit212$aic,
                mape_test=mape(mfcast212$mean, actual$tutup),
                mae_test=mae(mfcast212$mean, actual$tutup),
                rmse_test = rmse(mfcast212$mean, actual$tutup))
print(t(m7[,2:5]))
```

```
##                    [,1]
## aic        2569.2077433
## mape_test     0.1324836
## mae_test    603.0008125
## rmse_test   706.7566043
```

```r
mfit213 <- arima(training$tutup, order=c(2,1,3))
mfcast213 <- forecast(mfit213, h=10)
m8 = data.frame(model="arima(2,1,3)", aic=mfit213$aic,
                mape_test=mape(mfcast213$mean, actual$tutup),
                mae_test=mae(mfcast213$mean, actual$tutup),
                rmse_test = rmse(mfcast213$mean, actual$tutup))
print(t(m8[,2:5]))
```

```
##                    [,1]
## aic        2571.1531702
## mape_test     0.1338072
## mae_test    608.3414262
## rmse_test   711.1405845
```

```r
mfit311 <- arima(training$tutup, order=c(3,1,1))
mfcast311 <- forecast(mfit311, h=10)
m9 = data.frame(model="arima(3,1,1)", aic=mfit311$aic,
                mape_test=mape(mfcast311$mean, actual$tutup),
                mae_test=mae(mfcast311$mean, actual$tutup),
                rmse_test = rmse(mfcast311$mean, actual$tutup))
print(t(m9[,2:5]))
```

```
##                    [,1]
## aic        2571.8197326
## mape_test     0.1214905
## mae_test    558.5567672
## rmse_test   667.8653268
```

```r
mfit312 <- arima(training$tutup, order=c(3,1,2))
mfcast312 <- forecast(mfit312, h=10)
m10 = data.frame(model="arima(3,1,2)", aic=mfit312$aic,
                mape_test=mape(mfcast312$mean, actual$tutup),
                mae_test=mae(mfcast312$mean, actual$tutup),
                rmse_test = rmse(mfcast312$mean, actual$tutup))
print(t(m10[,2:5]))
```

```
##                    [,1]
## aic        2571.1340052
## mape_test     0.1342026
## mae_test    609.9363113
## rmse_test   712.4380873
```

```r
mfit112 <- arima(training$tutup, order=c(1,1,2))
mfcast112 <- forecast(mfit112, h=10)
m11 = data.frame(model="arima(1,1,2)", aic=mfit112$aic,
                mape_test=mape(mfcast112$mean, actual$tutup),
                mae_test=mae(mfcast112$mean, actual$tutup),
                rmse_test = rmse(mfcast112$mean, actual$tutup))
print(t(m11[,2:5]))
```

```
##                   [,1]
## aic       2569.8123447
## mape_test    0.1213373
## mae_test   557.9361702
## rmse_test  667.2863804
```

```r
mfit113 <- arima(training$tutup, order=c(1,1,3))
mfcast113 <- forecast(mfit113, h=10)
m12 = data.frame(model="arima(1,1,3)", aic=mfit113$aic,
                mape_test=mape(mfcast113$mean, actual$tutup),
                mae_test=mae(mfcast113$mean, actual$tutup),
                rmse_test = rmse(mfcast113$mean, actual$tutup))
print(t(m12[,2:5]))
```

```
##                   [,1]
## aic       2571.7944642
## mape_test    0.1208398
## mae_test   555.9523041
## rmse_test  665.1793981
```

```r
mfit411 <- arima(training$tutup, order=c(4,1,1))
mfcast411 <- forecast(mfit411, h=10)
m13 = data.frame(model="arima(4,1,1)", aic=mfit411$aic,
                mape_test=mape(mfcast411$mean, actual$tutup),
                mae_test=mae(mfcast411$mean, actual$tutup),
                rmse_test = rmse(mfcast411$mean, actual$tutup))
print(t(m13[,2:5]))
```

```
##                   [,1]
## aic       2571.0614404
## mape_test    0.1319864
## mae_test   601.4686726
## rmse_test  701.4302620
```

```r
mfit412 <- arima(training$tutup, order=c(4,1,2))
mfcast412 <- forecast(mfit412, h=10)
m14 = data.frame(model="arima(4,1,2)", aic=mfit412$aic,
                mape_test=mape(mfcast412$mean, actual$tutup),
                mae_test=mae(mfcast412$mean, actual$tutup),
                rmse_test = rmse(mfcast412$mean, actual$tutup))
print(t(m14[,2:5]))
```

```
##                   [,1]
## aic       2569.7224580
## mape_test    0.1248977
## mae_test   572.7063650
## rmse_test  677.1782120
```

```
mfit413 <- arima(training$tutup, order=c(4,1,3))
mfcast413 <- forecast(mfit413, h=10)
m15 = data.frame(model="arima(4,1,3)", aic=mfit413$aic,
                mape_test=mape(mfcast413$mean, actual$tutup),
                mae_test=mae(mfcast413$mean, actual$tutup),
                rmse_test = rmse(mfcast413$mean, actual$tutup))
print(t(m15[,2:5]))
```

```
##                     [,1]
## aic       2567.9715908
## mape_test    0.1296406
## mae_test   589.8996739
## rmse_test  704.2066715
```

```
mfit414 <- arima(training$tutup, order=c(4,1,4))
mfcast414 <- forecast(mfit414, h=10)
m16 = data.frame(model="arima(4,1,4)", aic=mfit414$aic,
                mape_test=mape(mfcast414$mean, actual$tutup),
                mae_test=mae(mfcast414$mean, actual$tutup),
                rmse_test = rmse(mfcast414$mean, actual$tutup))
print(t(m16[,2:5]))
```

```
##                   [,1]
## aic       2568.974282
## mape_test    0.122425
## mae_test   560.699005
## rmse_test  679.076649
```

Kesimpulan

Model Terbaik

```
eval = rbind(model_cv, m1, m2, m3, m11, m12, m6, m7, m8,
             m9, m10, m4, m5, m13, m14, m15, m16)
kable(eval)
```

model	aic	mape_test	mae_test	rmse_test
auto.arima / arima(1,1,1)	2568.282	0.1271844	581.7397	687.6709
arima(0,1,0)	2569.982	0.1256550	575.5000	682.5961
arima(2,2,1)	2702.951	0.0626069	321.0536	363.8986
arima(1,1,2)	2569.812	0.1213373	557.9362	667.2864
arima(1,1,3)	2571.794	0.1208398	555.9523	665.1794
arima(2,1,1)	2569.822	0.1216543	559.2124	668.5354
arima(2,1,2)	2569.208	0.1324836	603.0008	706.7566
arima(2,1,3)	2571.153	0.1338072	608.3414	711.1406
arima(3,1,1)	2571.820	0.1214905	558.5568	667.8653
arima(3,1,2)	2571.134	0.1342026	609.9363	712.4381
arima(3,1,3)	2566.693	0.0846400	407.6965	501.2489
arima(7,3,1)	2728.362	0.1232563	571.7527	639.9781
arima(4,1,1)	2571.061	0.1319864	601.4687	701.4303
arima(4,1,2)	2569.722	0.1248977	572.7064	677.1782
arima(4,1,3)	2567.972	0.1296406	589.8997	704.2067
arima(4,1,4)	2568.974	0.1224250	560.6990	679.0766

```r
summary(mfcast313$model)
```

```
##
## Call:
## arima(x = training$tutup, order = c(3, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     ma3
##       1.2053  0.3181  -0.5986  -1.1688  -0.5682  0.7975
## s.e.  0.1716  0.3241   0.1659   0.1400   0.2701  0.1387
##
## sigma^2 estimated as 36528:  log likelihood = -1276.35,  aic = 2566.69
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 28.08433 190.6243 132.9419 0.9751304 6.619854 0.9610982
##                     ACF1
## Training set -0.03061251
```

Diagnostik

```r
#coeftest(mfcast313)
```

```r
residuals = mfit313$residuals
```

```r
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.92321, p-value = 1.72e-08
```

```r
adf.test(residuals)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  residuals
## Dickey-Fuller = -5.6762, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
model <- lm(residuals^2 ~ seq_along(residuals))
bptest(model)
```

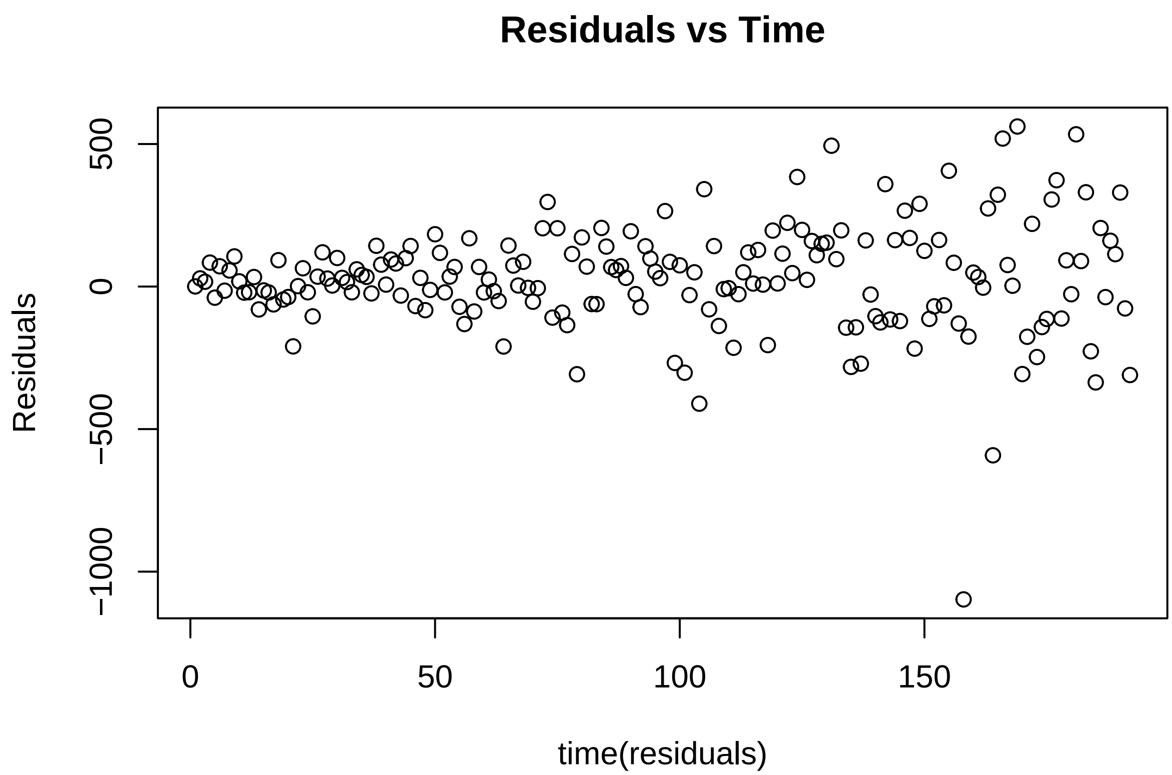
```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 2.2174, df = 1, p-value = 0.1365
```

```r
Box.test(residuals)
```

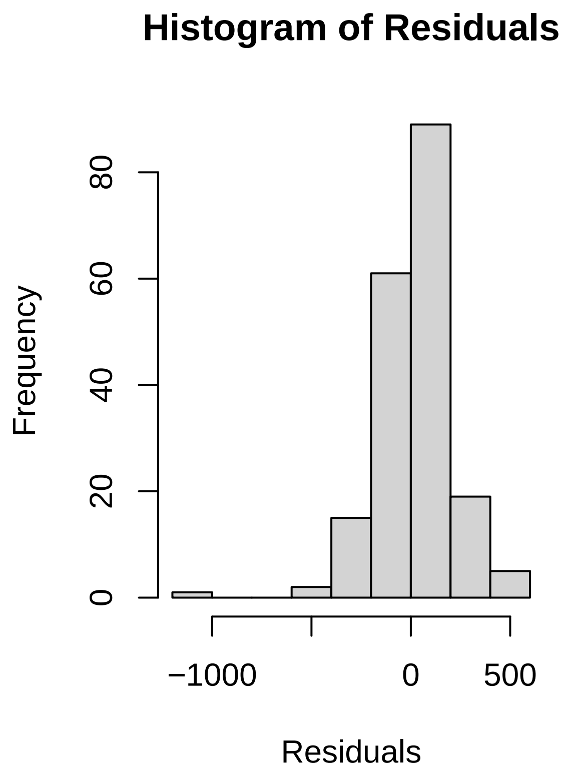
```
##
##  Box-Pierce test
```

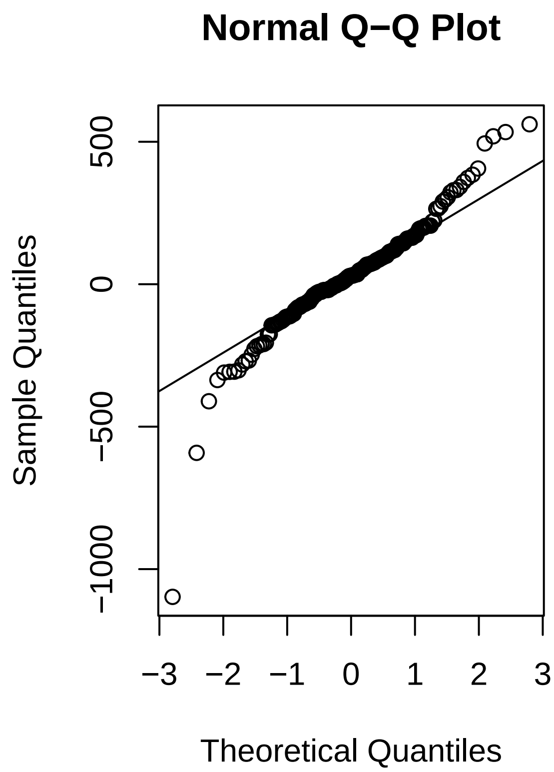
```
##
## data:  residuals
## X-squared = 0.17993, df = 1, p-value = 0.6714
```

```r
plot(residuals ~ time(residuals), main = "Residuals vs Time", ylab = "Residuals")
```




```r
par(mfrow = c(1,2))
hist(residuals, main = "Histogram of Residuals", xlab = "Residuals")
qqnorm(residuals)
qqline(residuals)
```






```r
influence.measures(lm(residuals ~ 1))
```

```
## Influence measures of
##   lm(formula = residuals ~ 1) :
##
##         dfb.1_     dffit cov.r   cook.d     hat inf
## 1   -1.06e-02 -1.06e-02 1.010 1.12e-04 0.00521
## 2    1.86e-04  1.86e-04 1.011 3.48e-08 0.00521
## 3   -4.27e-03 -4.27e-03 1.011 1.84e-05 0.00521
## 4    2.14e-02  2.14e-02 1.010 4.61e-04 0.00521
## 5   -2.57e-02 -2.57e-02 1.010 6.63e-04 0.00521
## 6    1.64e-02  1.64e-02 1.010 2.72e-04 0.00521
## 7   -1.63e-02 -1.63e-02 1.010 2.68e-04 0.00521
## 8    1.08e-02  1.08e-02 1.010 1.18e-04 0.00521
## 9    2.97e-02  2.97e-02 1.010 8.88e-04 0.00521
## 10  -3.78e-03 -3.78e-03 1.011 1.43e-05 0.00521
## 11  -1.89e-02 -1.89e-02 1.010 3.60e-04 0.00521
## 12  -1.84e-02 -1.84e-02 1.010 3.40e-04 0.00521
## 13   2.04e-03  2.04e-03 1.011 4.20e-06 0.00521
## 14  -4.15e-02 -4.15e-02 1.009 1.73e-03 0.00521
## 15  -1.60e-02 -1.60e-02 1.010 2.56e-04 0.00521
## 16  -1.87e-02 -1.87e-02 1.010 3.52e-04 0.00521
## 17  -3.47e-02 -3.47e-02 1.009 1.21e-03 0.00521
## 18   2.47e-02  2.47e-02 1.010 6.10e-04 0.00521
## 19  -2.82e-02 -2.82e-02 1.010 7.98e-04 0.00521
## 20  -2.48e-02 -2.48e-02 1.010 6.20e-04 0.00521
```

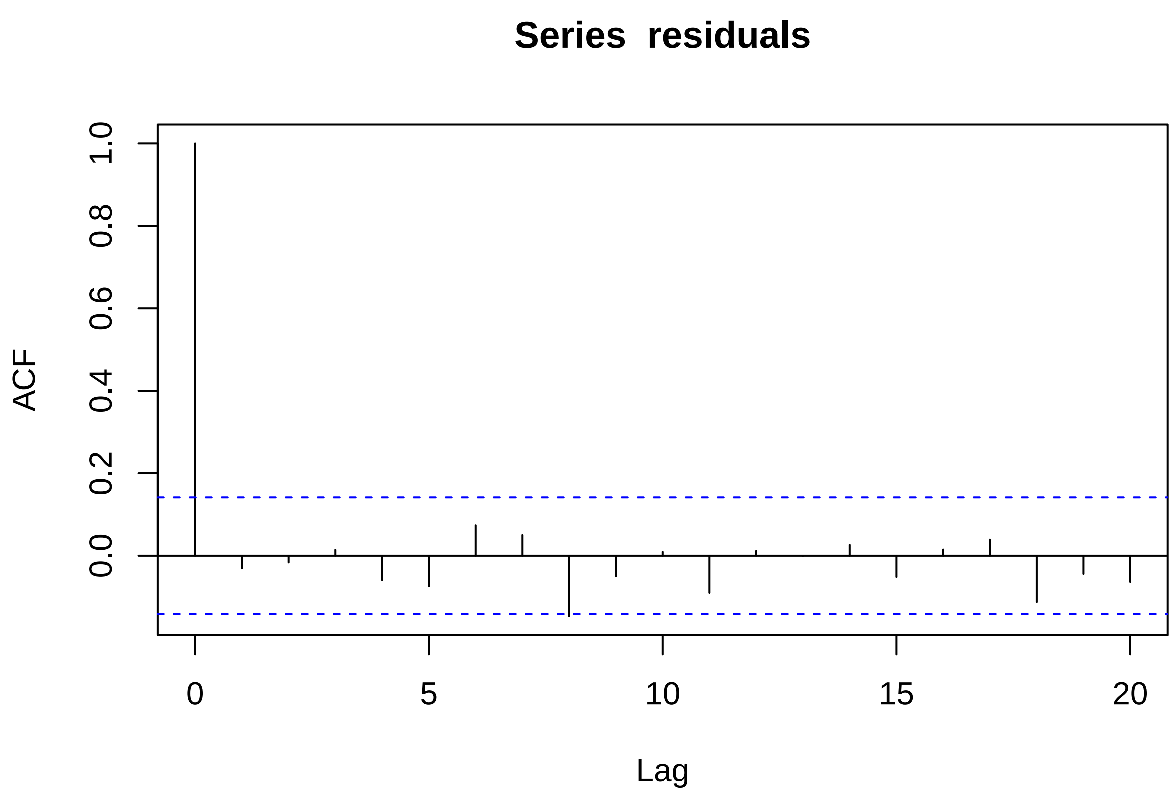
```
## 21  -9.15e-02 -9.15e-02 1.002 8.34e-03 0.00521
## 22  -1.02e-02 -1.02e-02 1.010 1.04e-04 0.00521
## 23   1.38e-02  1.38e-02 1.010 1.91e-04 0.00521
## 24  -1.85e-02 -1.85e-02 1.010 3.43e-04 0.00521
## 25  -5.08e-02 -5.08e-02 1.008 2.59e-03 0.00521
## 26   2.56e-03  2.56e-03 1.011 6.61e-06 0.00521
## 27   3.52e-02  3.52e-02 1.009 1.24e-03 0.00521
## 28  -3.18e-05 -3.18e-05 1.011 1.02e-09 0.00521
## 29  -9.37e-03 -9.37e-03 1.010 8.83e-05 0.00521
## 30   2.77e-02  2.77e-02 1.010 7.68e-04 0.00521
## 31   8.88e-04  8.88e-04 1.011 7.93e-07 0.00521
## 32  -4.75e-03 -4.75e-03 1.011 2.26e-05 0.00521
## 33  -1.85e-02 -1.85e-02 1.010 3.43e-04 0.00521
## 34   1.24e-02  1.24e-02 1.010 1.54e-04 0.00521
## 35   4.82e-03  4.82e-03 1.011 2.33e-05 0.00521
## 36   2.23e-03  2.23e-03 1.011 5.02e-06 0.00521
## 37  -2.00e-02 -2.00e-02 1.010 4.01e-04 0.00521
## 38   4.40e-02  4.40e-02 1.009 1.95e-03 0.00521
## 39   1.87e-02  1.87e-02 1.010 3.50e-04 0.00521
## 40  -8.13e-03 -8.13e-03 1.010 6.64e-05 0.00521
## 41   2.55e-02  2.55e-02 1.010 6.56e-04 0.00521
## 42   2.02e-02  2.02e-02 1.010 4.10e-04 0.00521
## 43  -2.29e-02 -2.29e-02 1.010 5.25e-04 0.00521
## 44   2.74e-02  2.74e-02 1.010 7.56e-04 0.00521
## 45   4.38e-02  4.38e-02 1.009 1.92e-03 0.00521
## 46  -3.69e-02 -3.69e-02 1.009 1.37e-03 0.00521
## 47   9.26e-04  9.26e-04 1.011 8.62e-07 0.00521
## 48  -4.25e-02 -4.25e-02 1.009 1.81e-03 0.00521
## 49  -1.53e-02 -1.53e-02 1.010 2.34e-04 0.00521
## 50   5.96e-02  5.96e-02 1.007 3.56e-03 0.00521
## 51   3.45e-02  3.45e-02 1.009 1.19e-03 0.00521
## 52  -1.86e-02 -1.86e-02 1.010 3.49e-04 0.00521
## 53   2.63e-03  2.63e-03 1.011 6.95e-06 0.00521
## 54   1.55e-02  1.55e-02 1.010 2.43e-04 0.00521
## 55  -3.79e-02 -3.79e-02 1.009 1.44e-03 0.00521
## 56  -6.12e-02 -6.12e-02 1.007 3.75e-03 0.00521
## 57   5.41e-02  5.41e-02 1.008 2.94e-03 0.00521
## 58  -4.43e-02 -4.43e-02 1.009 1.97e-03 0.00521
## 59   1.56e-02  1.56e-02 1.010 2.44e-04 0.00521
## 60  -1.87e-02 -1.87e-02 1.010 3.51e-04 0.00521
## 61  -1.36e-03 -1.36e-03 1.011 1.85e-06 0.00521
## 62  -1.69e-02 -1.69e-02 1.010 2.87e-04 0.00521
## 63  -3.02e-02 -3.02e-02 1.010 9.17e-04 0.00521
## 64  -9.16e-02 -9.16e-02 1.002 8.37e-03 0.00521
## 65   4.44e-02  4.44e-02 1.009 1.97e-03 0.00521
## 66   1.74e-02  1.74e-02 1.010 3.05e-04 0.00521
## 67  -9.36e-03 -9.36e-03 1.010 8.81e-05 0.00521
## 68   2.25e-02  2.25e-02 1.010 5.08e-04 0.00521
## 69  -1.26e-02 -1.26e-02 1.010 1.59e-04 0.00521
## 70  -3.12e-02 -3.12e-02 1.010 9.77e-04 0.00521
## 71  -1.29e-02 -1.29e-02 1.010 1.68e-04 0.00521
## 72   6.77e-02  6.77e-02 1.006 4.59e-03 0.00521
## 73   1.03e-01  1.03e-01 1.000 1.06e-02 0.00521
## 74  -5.25e-02 -5.25e-02 1.008 2.77e-03 0.00521
```

```
## 75   6.77e-02  6.77e-02 1.006 4.59e-03 0.00521
## 76  -4.58e-02 -4.58e-02 1.008 2.10e-03 0.00521
## 77  -6.26e-02 -6.26e-02 1.007 3.93e-03 0.00521
## 78   3.30e-02  3.30e-02 1.009 1.10e-03 0.00521
## 79  -1.30e-01 -1.30e-01 0.994 1.66e-02 0.00521
## 80   5.53e-02  5.53e-02 1.007 3.07e-03 0.00521
## 81   1.60e-02  1.60e-02 1.010 2.57e-04 0.00521
## 82  -3.42e-02 -3.42e-02 1.009 1.18e-03 0.00521
## 83  -3.44e-02 -3.44e-02 1.009 1.19e-03 0.00521
## 84   6.81e-02  6.81e-02 1.006 4.64e-03 0.00521
## 85   4.30e-02  4.30e-02 1.009 1.85e-03 0.00521
## 86   1.54e-02  1.54e-02 1.010 2.38e-04 0.00521
## 87   1.15e-02  1.15e-02 1.010 1.33e-04 0.00521
## 88   1.67e-02  1.67e-02 1.010 2.80e-04 0.00521
## 89   9.25e-04  9.25e-04 1.011 8.60e-07 0.00521
## 90   6.36e-02  6.36e-02 1.006 4.05e-03 0.00521
## 91  -2.09e-02 -2.09e-02 1.010 4.40e-04 0.00521
## 92  -3.84e-02 -3.84e-02 1.009 1.48e-03 0.00521
## 93   4.34e-02  4.34e-02 1.009 1.89e-03 0.00521
## 94   2.68e-02  2.68e-02 1.010 7.21e-04 0.00521
## 95   9.30e-03  9.30e-03 1.010 8.70e-05 0.00521
## 96   5.09e-04  5.09e-04 1.011 2.60e-07 0.00521
## 97   9.09e-02  9.09e-02 1.002 8.24e-03 0.00521
## 98   2.24e-02  2.24e-02 1.010 5.05e-04 0.00521
## 99  -1.14e-01 -1.14e-01 0.997 1.29e-02 0.00521
## 100  1.77e-02  1.77e-02 1.010 3.16e-04 0.00521
## 101 -1.27e-01 -1.27e-01 0.994 1.61e-02 0.00521
## 102 -2.22e-02 -2.22e-02 1.010 4.96e-04 0.00521
## 103  8.35e-03  8.35e-03 1.010 7.01e-05 0.00521
## 104 -1.70e-01 -1.70e-01 0.982 2.84e-02 0.00521    *
## 105  1.21e-01  1.21e-01 0.996 1.45e-02 0.00521
## 106 -4.13e-02 -4.13e-02 1.009 1.71e-03 0.00521
## 107  4.37e-02  4.37e-02 1.009 1.91e-03 0.00521
## 108 -6.38e-02 -6.38e-02 1.006 4.08e-03 0.00521
## 109 -1.42e-02 -1.42e-02 1.010 2.03e-04 0.00521
## 110 -1.29e-02 -1.29e-02 1.010 1.69e-04 0.00521
## 111 -9.33e-02 -9.33e-02 1.002 8.68e-03 0.00521
## 112 -2.10e-02 -2.10e-02 1.010 4.43e-04 0.00521
## 113  8.35e-03  8.35e-03 1.010 7.01e-05 0.00521
## 114  3.51e-02  3.51e-02 1.009 1.23e-03 0.00521
## 115 -6.70e-03 -6.70e-03 1.010 4.51e-05 0.00521
## 116  3.84e-02  3.84e-02 1.009 1.48e-03 0.00521
## 117 -8.13e-03 -8.13e-03 1.010 6.65e-05 0.00521
## 118 -8.97e-02 -8.97e-02 1.002 8.02e-03 0.00521
## 119  6.45e-02  6.45e-02 1.006 4.17e-03 0.00521
## 120 -6.62e-03 -6.62e-03 1.010 4.40e-05 0.00521
## 121  3.34e-02  3.34e-02 1.009 1.12e-03 0.00521
## 122  7.51e-02  7.51e-02 1.005 5.63e-03 0.00521
## 123  7.31e-03  7.31e-03 1.010 5.37e-05 0.00521
## 124  1.38e-01  1.38e-01 0.992 1.87e-02 0.00521
## 125  6.54e-02  6.54e-02 1.006 4.28e-03 0.00521
## 126 -1.75e-03 -1.75e-03 1.011 3.08e-06 0.00521
## 127  5.05e-02  5.05e-02 1.008 2.55e-03 0.00521
## 128  3.13e-02  3.13e-02 1.010 9.87e-04 0.00521
```

```
## 129  4.67e-02  4.67e-02 1.008 2.18e-03 0.00521
## 130  4.83e-02  4.83e-02 1.008 2.34e-03 0.00521
## 131  1.81e-01  1.81e-01 0.978 3.20e-02 0.00521   *
## 132  2.60e-02  2.60e-02 1.010 6.82e-04 0.00521
## 133  6.48e-02  6.48e-02 1.006 4.21e-03 0.00521
## 134 -6.61e-02 -6.61e-02 1.006 4.37e-03 0.00521
## 135 -1.19e-01 -1.19e-01 0.996 1.41e-02 0.00521
## 136 -6.56e-02 -6.56e-02 1.006 4.31e-03 0.00521
## 137 -1.15e-01 -1.15e-01 0.997 1.31e-02 0.00521
## 138  5.13e-02  5.13e-02 1.008 2.64e-03 0.00521
## 139 -2.14e-02 -2.14e-02 1.010 4.62e-04 0.00521
## 140 -5.06e-02 -5.06e-02 1.008 2.57e-03 0.00521
## 141 -5.90e-02 -5.90e-02 1.007 3.49e-03 0.00521
## 142  1.28e-01  1.28e-01 0.994 1.62e-02 0.00521
## 143 -5.50e-02 -5.50e-02 1.007 3.03e-03 0.00521
## 144  5.17e-02  5.17e-02 1.008 2.68e-03 0.00521
## 145 -5.72e-02 -5.72e-02 1.007 3.28e-03 0.00521
## 146  9.15e-02  9.15e-02 1.002 8.34e-03 0.00521
## 147  5.46e-02  5.46e-02 1.008 2.99e-03 0.00521
## 148 -9.45e-02 -9.45e-02 1.002 8.89e-03 0.00521
## 149  1.01e-01  1.01e-01 1.000 1.01e-02 0.00521
## 150  3.72e-02  3.72e-02 1.009 1.39e-03 0.00521
## 151 -5.43e-02 -5.43e-02 1.008 2.95e-03 0.00521
## 152 -3.74e-02 -3.74e-02 1.009 1.40e-03 0.00521
## 153  5.18e-02  5.18e-02 1.008 2.69e-03 0.00521
## 154 -3.60e-02 -3.60e-02 1.009 1.30e-03 0.00521
## 155  1.46e-01  1.46e-01 0.989 2.10e-02 0.00521
## 156  2.12e-02  2.12e-02 1.010 4.52e-04 0.00521
## 157 -6.05e-02 -6.05e-02 1.007 3.67e-03 0.00521
## 158 -4.78e-01 -4.78e-01 0.822 1.87e-01 0.00521   *
## 159 -7.81e-02 -7.81e-02 1.004 6.10e-03 0.00521
## 160  8.00e-03  8.00e-03 1.010 6.43e-05 0.00521
## 161  2.04e-03  2.04e-03 1.011 4.16e-06 0.00521
## 162 -1.22e-02 -1.22e-02 1.010 1.50e-04 0.00521
## 163  9.45e-02  9.45e-02 1.002 8.91e-03 0.00521
## 164 -2.44e-01 -2.44e-01 0.953 5.66e-02 0.00521   *
## 165  1.13e-01  1.13e-01 0.998 1.27e-02 0.00521
## 166  1.91e-01  1.91e-01 0.975 3.55e-02 0.00521   *
## 167  1.82e-02  1.82e-02 1.010 3.32e-04 0.00521
## 168 -9.65e-03 -9.65e-03 1.010 9.36e-05 0.00521
## 169  2.09e-01  2.09e-01 0.968 4.19e-02 0.00521   *
## 170 -1.29e-01 -1.29e-01 0.994 1.65e-02 0.00521
## 171 -7.84e-02 -7.84e-02 1.004 6.15e-03 0.00521
## 172  7.37e-02  7.37e-02 1.005 5.42e-03 0.00521
## 173 -1.06e-01 -1.06e-01 0.999 1.12e-02 0.00521
## 174 -6.52e-02 -6.52e-02 1.006 4.26e-03 0.00521
## 175 -5.43e-02 -5.43e-02 1.008 2.96e-03 0.00521
## 176  1.07e-01  1.07e-01 0.999 1.13e-02 0.00521
## 177  1.33e-01  1.33e-01 0.993 1.75e-02 0.00521
## 178 -5.38e-02 -5.38e-02 1.008 2.90e-03 0.00521
## 179  2.45e-02  2.45e-02 1.010 6.02e-04 0.00521
## 180 -2.10e-02 -2.10e-02 1.010 4.45e-04 0.00521
## 181  1.97e-01  1.97e-01 0.972 3.77e-02 0.00521   *
## 182  2.35e-02  2.35e-02 1.010 5.54e-04 0.00521
```

```
## 183  1.17e-01  1.17e-01 0.997 1.35e-02 0.00521
## 184 -9.81e-02 -9.81e-02 1.001 9.59e-03 0.00521
## 185 -1.41e-01 -1.41e-01 0.991 1.96e-02 0.00521
## 186  6.80e-02  6.80e-02 1.006 4.63e-03 0.00521
## 187 -2.49e-02 -2.49e-02 1.010 6.25e-04 0.00521
## 188  5.08e-02  5.08e-02 1.008 2.59e-03 0.00521
## 189  3.28e-02  3.28e-02 1.009 1.08e-03 0.00521
## 190  1.16e-01  1.16e-01 0.997 1.34e-02 0.00521
## 191 -4.02e-02 -4.02e-02 1.009 1.62e-03 0.00521
## 192 -1.31e-01 -1.31e-01 0.993 1.69e-02 0.00521
```

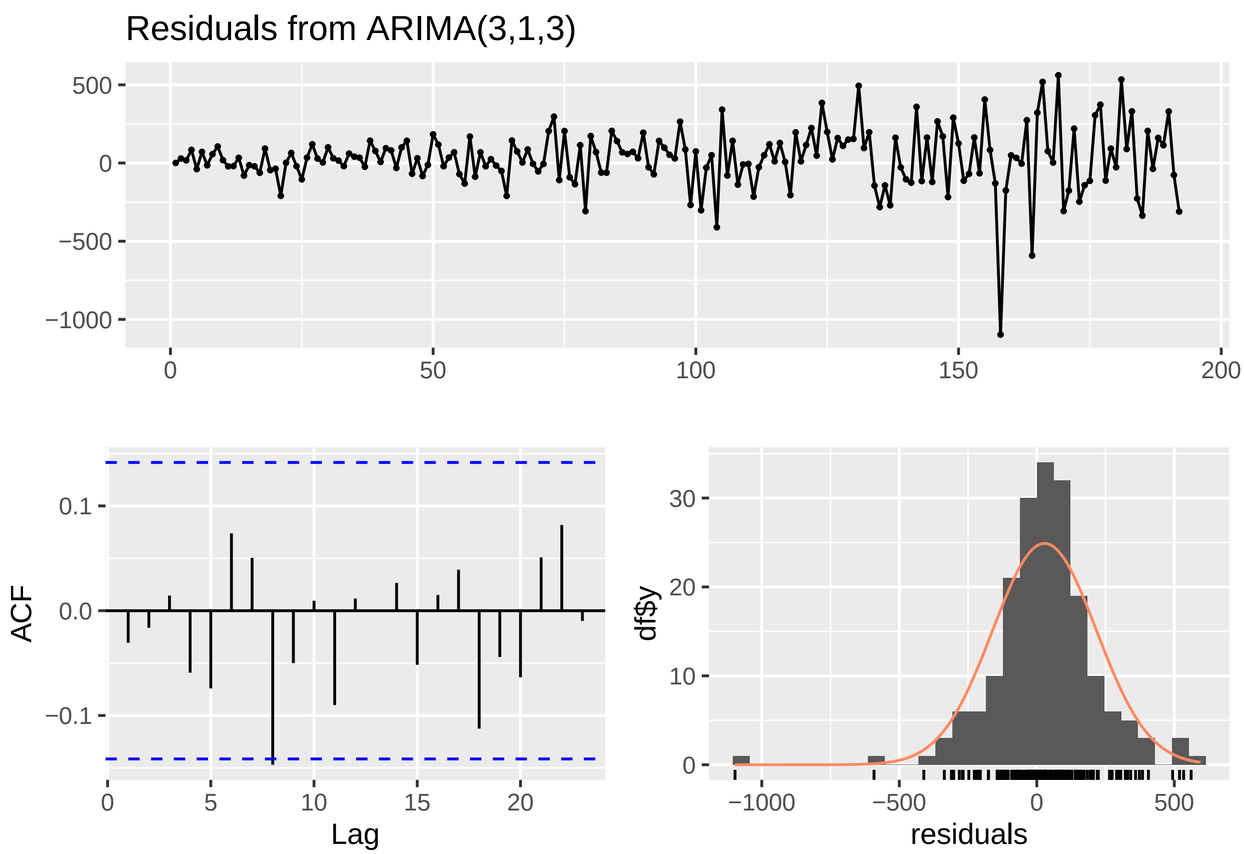
```
acf(residuals, lag.max = 20)
```




```
# Uji Autokorelasi Residuals (ACF1)
Box.test(residuals, lag = 1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 0.18275, df = 1, p-value = 0.669
```

```
checkresiduals(mfcast313)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,3)
## Q* = 8.5637, df = 4, p-value = 0.07298
##
## Model df: 6.   Total lags used: 10
```

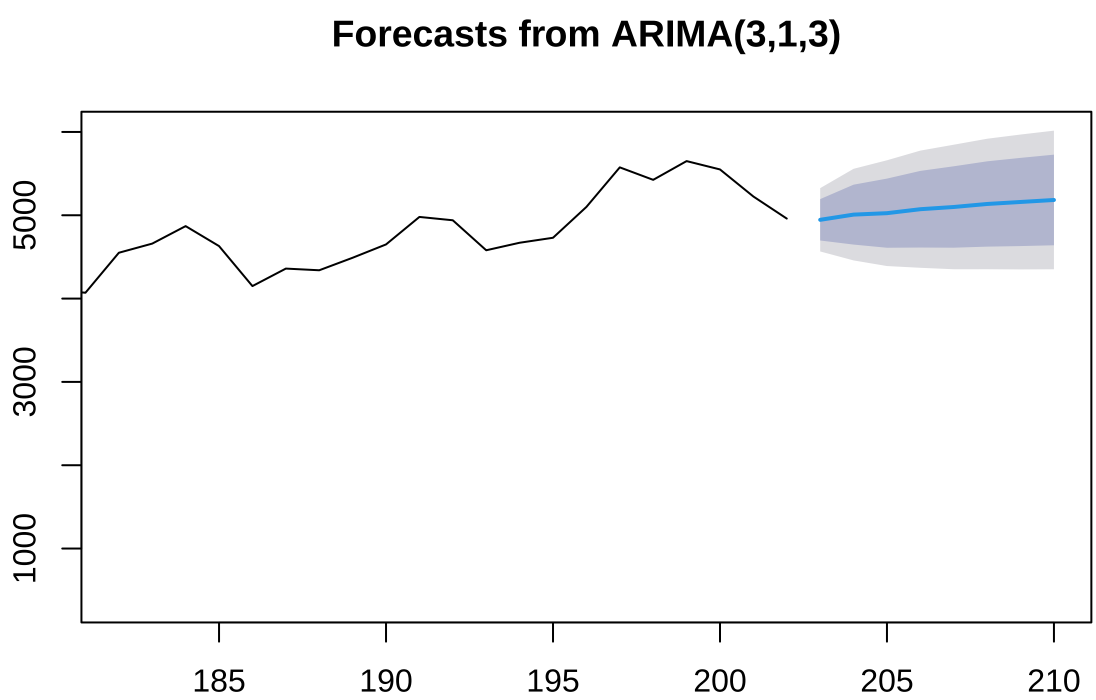
Peramalan

```
dataset$tutup = dataset_close$tutup
row.names(dataset) = as.Date(dataset_close$tanggal)
mfit_eval <- arima(dataset$tutup, order=c(3,1,3))
summary(mfit_eval)
```

```
##
## Call:
## arima(x = dataset$tutup, order = c(3, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     ma3
##       1.2137  0.2884  -0.5778  -1.1762  -0.5531  0.7899
## s.e.  0.1678  0.3162   0.1621   0.1382   0.2652  0.1365
##
## sigma^2 estimated as 37265:  log likelihood = -1345.17,  aic = 2704.34
##
```

```
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 28.29819 192.5634 135.9738 0.8998563 6.537238 0.9578077
##                     ACF1
## Training set -0.02121924
```

```r
mfcast_eval <- forecast(mfit_eval, h=8)
plot(mfcast_eval, xlim = c(length(dataset_close$tutup) - 20, length(dataset_close$tutup) + 8))
```




```r
summary(mfcast_eval)
```

```
##
## Forecast method: ARIMA(3,1,3)
##
## Model Information:
##
## Call:
## arima(x = dataset$tutup, order = c(3, 1, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     ma3
##       1.2137  0.2884  -0.5778  -1.1762  -0.5531  0.7899
## s.e.  0.1678  0.3162   0.1621   0.1382   0.2652  0.1365
##
## sigma^2 estimated as 37265:  log likelihood = -1345.17,  aic = 2704.34
##
## Error measures:
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 28.29819 192.5634 135.9738 0.8998563 6.537238 0.9578077
##                     ACF1
## Training set -0.02121924
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 203       4945.521 4697.065 5193.978 4565.540 5325.503
## 204       5007.690 4648.162 5367.218 4457.840 5557.540
## 205       5025.152 4610.060 5440.244 4390.323 5659.981
## 206       5072.638 4612.660 5532.615 4369.162 5776.113
## 207       5099.387 4610.635 5588.139 4351.906 5846.868
## 208       5135.457 4623.375 5647.539 4352.295 5918.619
## 209       5159.513 4630.296 5688.730 4350.145 5968.880
## 210       5183.655 4639.717 5727.593 4351.773 6015.537
```