

COVID-19 New Cases

Time Series Analysis of The Effect of Increasing Number of COVID-19 Cases in Indonesia on The Movement of IHSG

5/11/2021

Covid-19 is a world wide pandemic started in December 2019 and firstly reported from Wuhan, China. In Indonesia, Covid-19 started to spread in March 2020. Of course, this pandemic affect most all life aspects, including economic sector. This research focus on the impact of Covid-19 pandemic towards Indonesia economic condition. The economic parameter that will be considered in this research is Indonesia Stock Exchange (IDX) Composite.

The Composite Stock Price Index (IHSG) is an index of all stocks listed on the Indonesia Stock Exchange, IDX. The stock price index data we use is taken from IHSG for July 2020 until April 2021 which contains 179 observations data. The IHSG data then will be compared to Indonesia's COVID-19 new cases. The analysis method used is time series analysis (Box-Jenkins method and heteroskedastic effect modeling) and visual analysis. In this section, the COVID19 new cases data will be analyzed.

Data Input and Descriptive Statistics

```
# Indonesia's Covid-19 new cases
```

```
library(readxl)
```

```
library(tseries)
```

```
library(lmtest)
```

```
library(forecast)
```

```
library(TSA)
```

```
library(FinTS)
```

```
# Input Data
```

```
data <- read_xlsx("C:/Users/LEN0V0/Downloads/DataIHSGCovid.xlsx",  
sheet="Covid")
```

```
data
```

```
## # A tibble: 271 x 2
```

```
##   Tanggal      KasusBaru
```

```
##   <dtm>         <dbl>
```

```
## 1 2020-07-01 00:00:00    1385
```

```
## 2 2020-07-02 00:00:00    1624
```

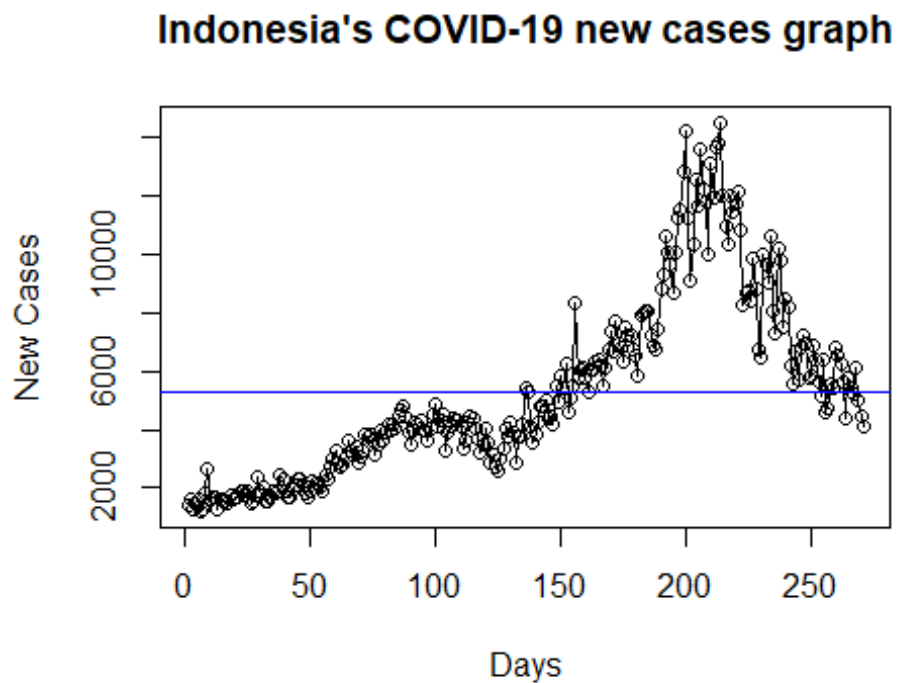
```
## 3 2020-07-03 00:00:00    1301
```

```
## 4 2020-07-04 00:00:00    1447
```

```
## 5 2020-07-05 00:00:00      1607
## 6 2020-07-06 00:00:00      1209
## 7 2020-07-07 00:00:00      1268
## 8 2020-07-08 00:00:00      1853
## 9 2020-07-09 00:00:00      2657
## 10 2020-07-10 00:00:00     1611
## # ... with 261 more rows

indeks <- data$KasusBaru

# Time Series data
indeksts<- ts(indeks)
# Scatter Plot Time Series Data
plot(indeksts, main = "Indonesia's COVID-19 new cases graph", ylab = "New
Cases", xlab = "Days", type = 'o') + abline(h = mean(indeks), col = 'blue')
```



```
# Descriptive statistics data
summary(indeks)

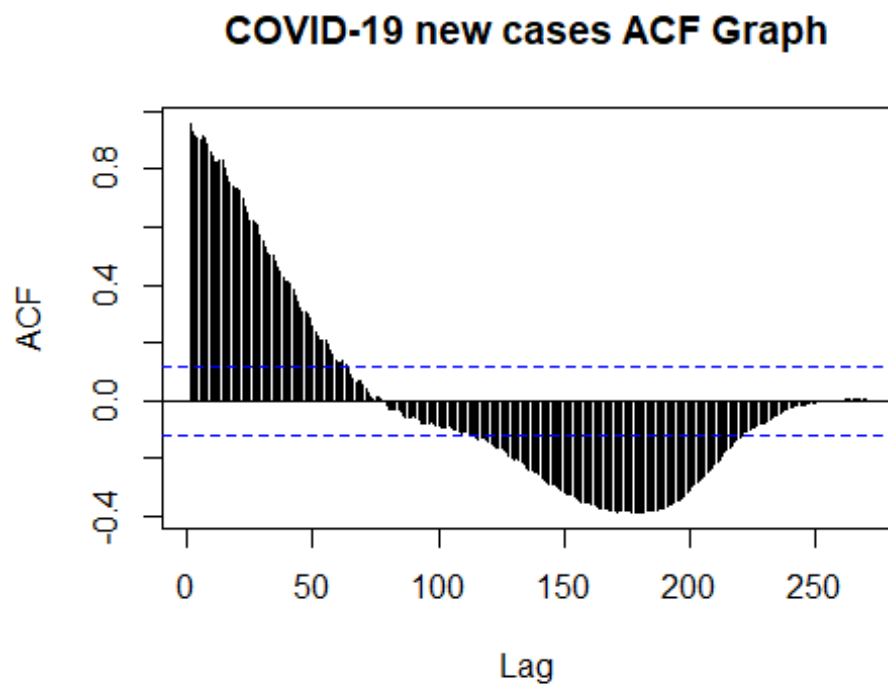
##      Min. 1st Qu.  Median    Mean 3rd Qu.     Max.
##    1209     3024     4432     5313     6836    14518
```

Data Modeling

Stationarity Test

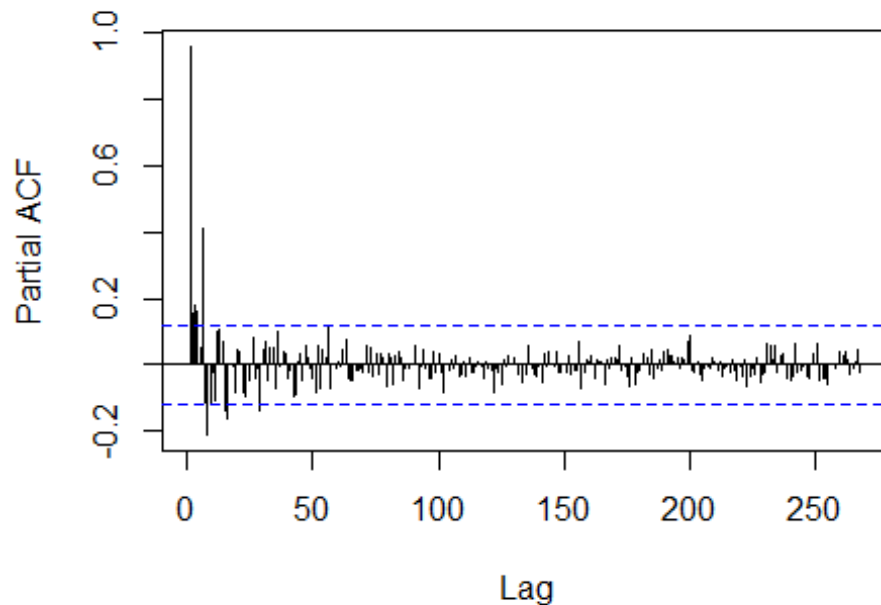
To conduct a time series analysis, the data should be stationary. Therefore, a stationarity test is applied.

```
# DATA MODELING  
# Stationarity Test  
# Plot ACF  
acf(indeksts, main = 'COVID-19 new cases ACF Graph', lag.max = 270)
```



```
pacf(indeksts, main = 'COVID-19 new cases PACF Graph', lag.max = 270)
```

COVID-19 new cases PACF Graph



```
# ADF(Augmented Dickey-Fuller) Test
adf.test(indeksts)

## Warning in adf.test(indeksts): p-value greater than printed p-value

##
## Augmented Dickey-Fuller Test
##
## data: indeksts
## Dickey-Fuller = 0.069728, Lag order = 6, p-value = 0.99
## alternative hypothesis: stationary
```

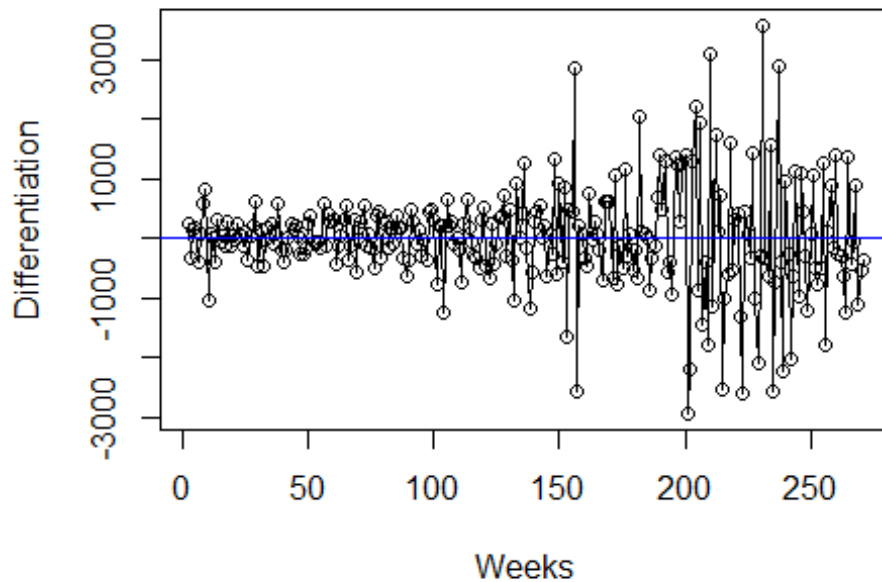
The ACF Graph shows a tail-off graph. From the ADF Test, it is found that P-Value = 0.99 for COVID-19 data, so H_0 is not rejected, meaning that the data is not stationary so differentiation is needed.

Differentiation

As the data is not stationary, differentiation is conducted.

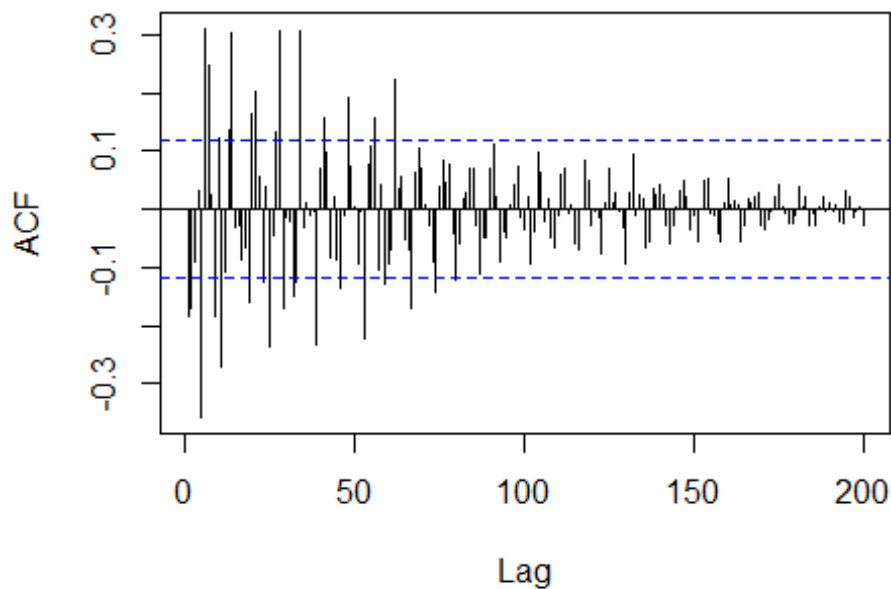
```
# Differentiation non stationary model
indekstdiff = diff(indeksts)
plot(indekstdiff, main = "1 Time Differentiation COVID-19 New Cases Graph",
     ylab = "Differentiation", xlab = "Weeks", type = 'o') + abline(h =
mean(indekstdiff), col = 'blue')
```

1 Time Differentiation COVID-19 New Cases Graph



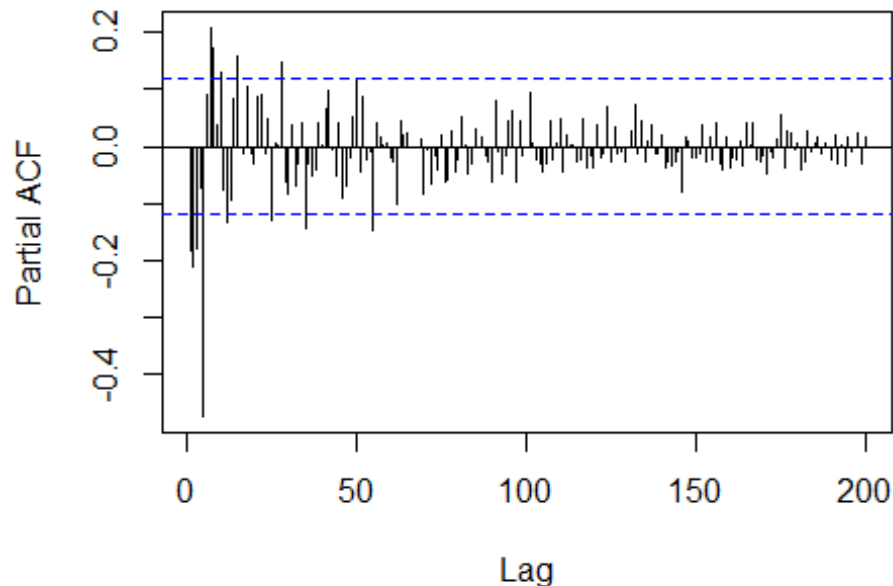
```
# Plot ACF 1x Differensiasi  
acf(indeksstdiff, main = '1 Time Differentiation COVID-19 New Cases ACF  
Graph', lag.max = 200)
```

1 Time Differentiation COVID-19 New Cases ACF Graph



```
pacf(indekstdiff, main = '1 Time Differentiation COVID-19 New Cases PACF Graph', lag.max = 200)
```

1 Time Differentiation COVID-19 New Cases PACF Graph



```
# ADF(Augmented Dickey-Fuller) Test
adf.test(indekstdiff)

## Warning in adf.test(indekstdiff): p-value smaller than printed p-value
##
## Augmented Dickey-Fuller Test
##
## data: indekstdiff
## Dickey-Fuller = -6.9809, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

It is shown that The ACF and PACF plot is not a tail-off graph. ADF Test on differentiation data gives P-Value = 0.01 for COVID-19 data. As H_0 is rejected, the data is now stationary.

Model Estimation

Based on the ACF and PACF plot after a differentiation on the data. it can be seen that the ACF cut-off is in the lag 2 and 5, while the PACF cut-off is in the lag 3 and 5. Therefore, the possible models are ARIMA (3,1,0), ARIMA (0,1,2), ARIMA (5,1,0), ARIMA (0,1,5), ARIMA (3,1,2), ARIMA (3,1,5), ARIMA (5,1,2), ARIMA (5,1,5).

```
# Manual Model Estimation
# ACF Cut Off Lag 2 and 5
```

```

# PACF Cut off Lag 3 and 5
model_arima1 = arima(indeksts, order = c(3,1,0))
summary(model_arima1)

##
## Call:
## arima(x = indeksts, order = c(3, 1, 0))
##
## Coefficients:
##          ar1          ar2          ar3
##      -0.2595  -0.2502  -0.1807
## s.e.   0.0599   0.0599   0.0599
##
## sigma^2 estimated as 680173:  log likelihood = -2196.29,  aic = 4398.58
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima2 = arima(indeksts, order = c(0,1,2))
summary(model_arima2)

##
## Call:
## arima(x = indeksts, order = c(0, 1, 2))
##
## Coefficients:
##          ma1          ma2
##      -0.3339  -0.2276
## s.e.   0.0548   0.0481
##
## sigma^2 estimated as 663273:  log likelihood = -2192.94,  aic = 4389.87
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima3 = arima(indeksts, order = c(5,1,0))
summary(model_arima3)

##
## Call:

```

```

## arima(x = indeksts, order = c(5, 1, 0))
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5
##      -0.3083  -0.3640  -0.3288  -0.2028  -0.4711
## s.e.   0.0534   0.0553   0.0562   0.0552   0.0532
##
## sigma^2 estimated as 522957:  log likelihood = -2161.44,  aic = 4332.89
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima4 = arima(indeksts, order = c(0, 1, 5))
summary(model_arima4)

##
## Call:
## arima(x = indeksts, order = c(0, 1, 5))
##
## Coefficients:
##          ma1          ma2          ma3          ma4          ma5
##      -0.2888  -0.3959  -0.1363   0.3306  -0.0003
## s.e.   0.0601   0.0612   0.0594   0.0690   0.0485
##
## sigma^2 estimated as 621349:  log likelihood = -2184.4,  aic = 4378.8
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima5 = arima(indeksts, order = c(3, 1, 2))
summary(model_arima5)

##
## Call:
## arima(x = indeksts, order = c(3, 1, 2))
##
## Coefficients:
##          ar1          ar2          ar3          ma1          ma2
##      -0.5593   0.1532  -0.2456   0.2965  -0.5716
## s.e.   0.0966   0.1142   0.0670   0.0871   0.0835

```



```
##
## sigma^2 estimated as 620345: log likelihood = -2184.1, aic = 4378.21
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##           ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima6 = arima(indeksts,order = c(3,1,5))
summary(model_arima6)

##
## Call:
## arima(x = indeksts, order = c(3, 1, 5))
##
## Coefficients:
##           ar1          ar2          ar3          ma1          ma2          ma3          ma4          ma5
##          -0.7436  -0.9522  -0.275   0.5083   0.6585  -0.1505  -0.2588  -0.3795
## s.e.    0.1810   0.0938   0.200   0.1635   0.0731   0.1627   0.0893   0.0505
##
## sigma^2 estimated as 555517: log likelihood = -2169.52, aic = 4355.04
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##           ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima7 = arima(indeksts,order = c(5,1,2))
summary(model_arima7)

##
## Call:
## arima(x = indeksts, order = c(5, 1, 2))
##
## Coefficients:
##           ar1          ar2          ar3          ar4          ar5          ma1          ma2
##          -0.3244  -0.6097  -0.3981  -0.2702  -0.5235   0.0252   0.3225
## s.e.    0.0970   0.0939   0.0652   0.0544   0.0518   0.1139   0.1031
##
## sigma^2 estimated as 503839: log likelihood = -2156.53, aic = 4327.07
##
## Training set error measures:
```

```
## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

model_arima8 = arima(indeksts, order = c(5,1,5))
summary(model_arima8)

##
## Call:
## arima(x = indeksts, order = c(5, 1, 5))
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5          ma1          ma2          ma3
##      -0.2335   -0.5507   -0.5149   -0.2529   -0.8021   -0.1014    0.3290    0.2455
## s.e.   0.0599    0.0608    0.0649    0.0646    0.0546    0.0893    0.1024    0.1073
##          ma4          ma5
##      0.0241    0.4590
## s.e.  0.0984    0.1009
##
## sigma^2 estimated as 453646:  log likelihood = -2142.86,  aic = 4305.73
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

# Data Coefficient Significancy Test
coeftest(model_arima8)

##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.233509   0.059878  -3.8998 9.629e-05 ***
## ar2 -0.550677   0.060806  -9.0563 < 2.2e-16 ***
## ar3 -0.514897   0.064930  -7.9300 2.191e-15 ***
## ar4 -0.252899   0.064593  -3.9152 9.031e-05 ***
## ar5 -0.802058   0.054569 -14.6982 < 2.2e-16 ***
## ma1 -0.101400   0.089319  -1.1353  0.256268
## ma2  0.328988   0.102381   3.2134  0.001312 **
## ma3  0.245486   0.107311   2.2876  0.022159 *
## ma4  0.024148   0.098388   0.2454  0.806115
## ma5  0.458965   0.100896   4.5489 5.393e-06 ***
```

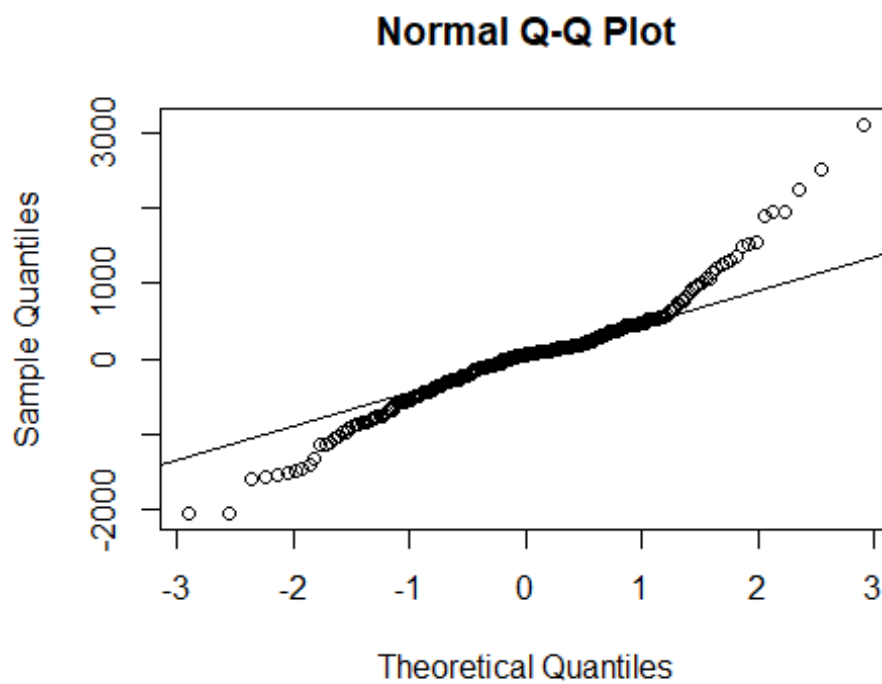
```
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model chosen is the best model with the smallest AIC value, namely **ARIMA(5,1,5)**.

Diagnostic Test

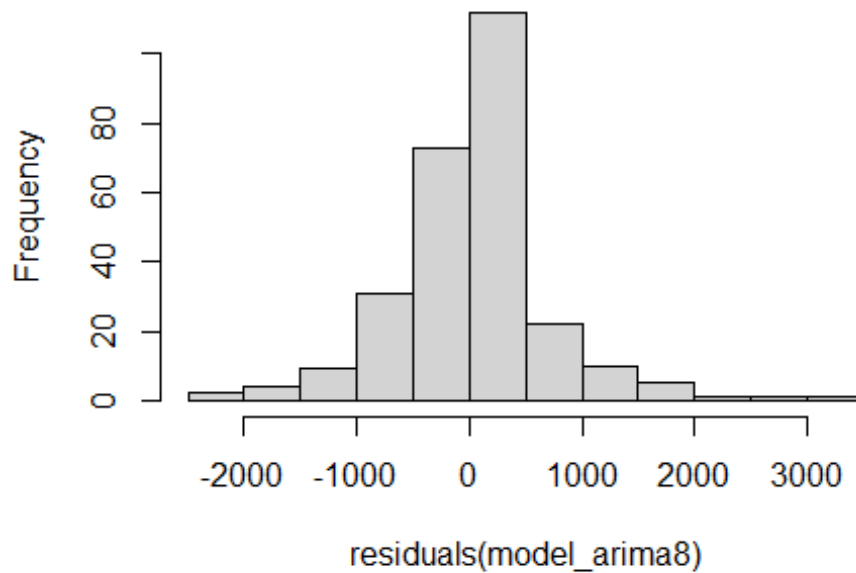
Further identification of the model is done in diagnostic test.

```
# Diagnostic Test  
qqnorm(residuals(model_arma8))  
qqline(residuals(model_arma8))
```



```
hist(residuals(model_arma8))
```

Histogram of residuals(model_arma8)



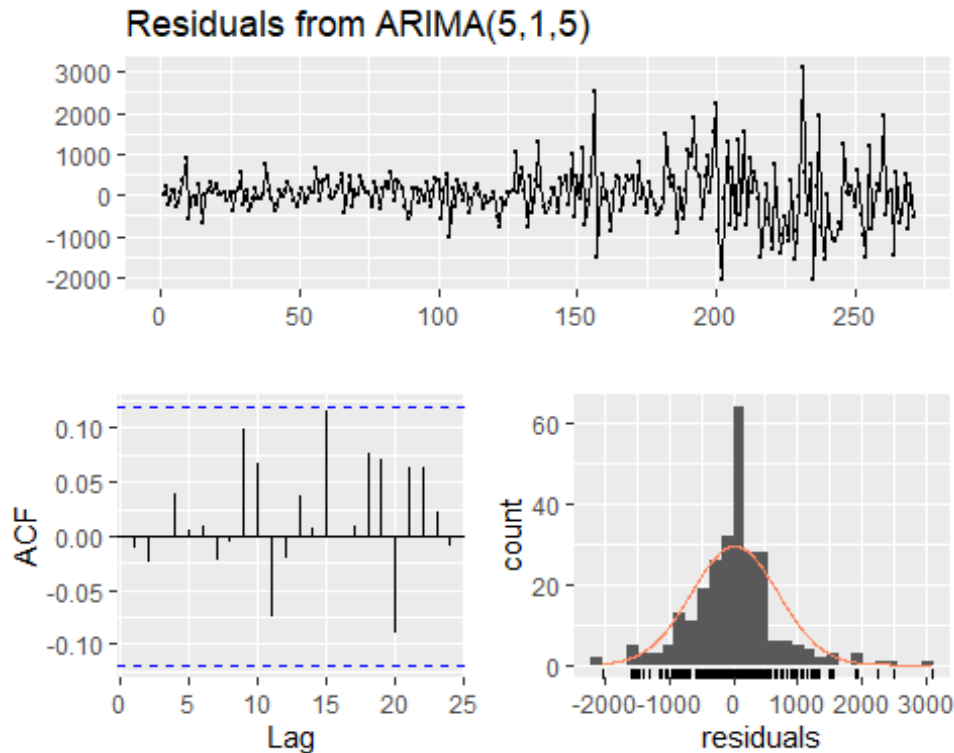
```
ks.test(residuals(model_arma8), "pnorm")
```

```
##  
## One-sample Kolmogorov-Smirnov test  
##  
## data: residuals(model_arma8)  
## D = 0.55351, p-value < 2.2e-16  
## alternative hypothesis: two-sided
```

```
Box.test(residuals(model_arma8), lag = 270, type="Ljung-Box")
```

```
##  
## Box-Ljung test  
##  
## data: residuals(model_arma8)  
## X-squared = 164.67, df = 270, p-value = 1
```

```
checkresiduals(model_arma8)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,1,5)
## Q* = 6.9389, df = 3, p-value = 0.07387
##
## Model df: 10.    Total lags used: 13

# Uji efek heteroskedastik
resid<-residuals(model_arima8)
ArchTest(resid)

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  resid
## Chi-squared = 43.175, df = 12, p-value = 2.109e-05
```

For the COVID-19 data, the histogram shows the model residuals are not close enough to the normal distribution and from the ACF plot it can be seen that nothing has crossed the significance limit, means that the errors are independent of each other's lags.

The results of the L-jung Box test are also supportive because it was found that P-Value = $0.07387 > \alpha = 0.05$, so H_0 was not rejected, which means that the model is quite suitable for the data.

But because the residuals plot looks strange and the variance seems getting bigger, then ARCH LM-test was performed to test the presence of heteroscedastic effect. Obtained PValue = $2.109e-05 < \alpha = 0.05$ then H_0 is rejected, which means that the residuals contain **heteroscedastic effect**.

Heteroscedastic Model Identification

As the ARCH-LM test shown that the residuals of the model contain heteroscedastic effect, Heteroscedastic model identification is conducted.

```
# Squared residual sequence definition
nt <- residuals(model_arma8)
nt2 <- nt^2

# Stationary test for the squared residual sequence
adf.test(nt2)

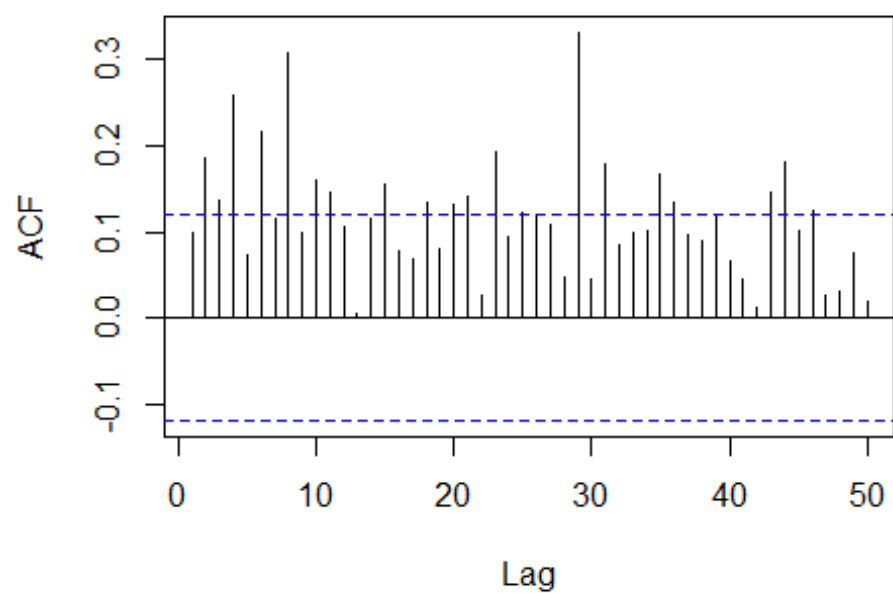
## Warning in adf.test(nt2): p-value smaller than printed p-value

##
## Augmented Dickey-Fuller Test
##
## data: nt2
## Dickey-Fuller = -5.0469, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The residual sequence is defined as nt and the square as nt2, then a stationary test is performed for nt2. Based on the ADF Test, it was found that P-Value = $0.01 < \alpha = 0.05$, then H_0 is rejected, which means that the residual quadratic sequence data is stationary.

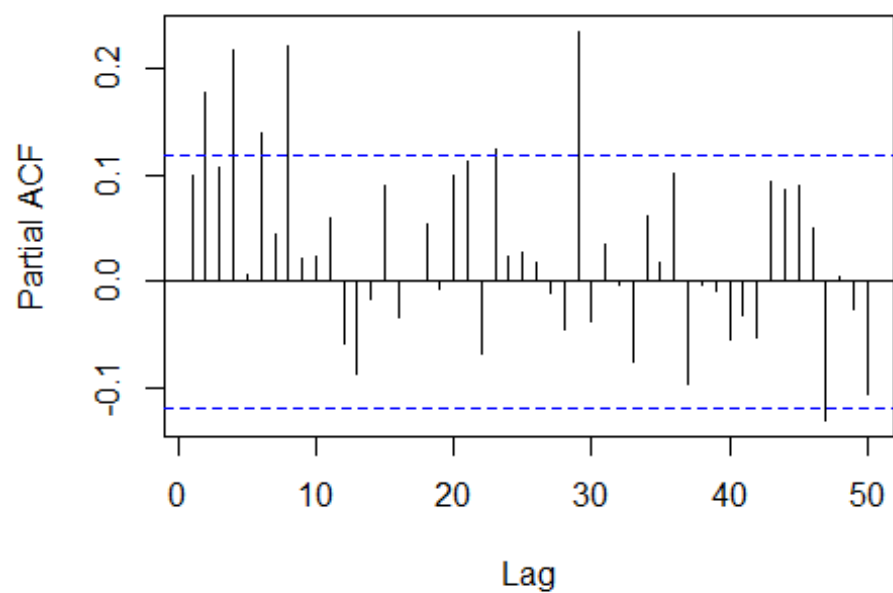
```
# ACF dan PACF squared residual
acf(nt2, lag.max = 50)
```

Series nt2



```
pacf(nt2, lag.max = 50)
```

Series nt2



```

# ARMA Modeling
# ACF cut-off lag 2,4,6
# PACF cut-off lag 2,4
# ARMA(max(p,q),p)
modelnt1 = arima(nt2,order = c(2,0,2))
summary(modelnt1)

##
## Call:
## arima(x = nt2, order = c(2, 0, 2))
##
## Coefficients:
##          ar1      ar2      ma1      ma2  intercept
##          0.0762  0.8911 -0.0756 -0.7705  428952.9
## s.e.    0.0611  0.0603  0.0800  0.0772  232132.4
##
## sigma^2 estimated as 8.823e+11:  log likelihood = -4111.9,  aic = 8233.8
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

modelnt2 = arima(nt2,order = c(2,0,4))
summary(modelnt2)

##
## Call:
## arima(x = nt2, order = c(2, 0, 4))
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3      ma4  intercept
##          0.0714  0.8477 -0.0487 -0.7946 -0.0034  0.1259  432632.1
## s.e.    0.0748  0.0777  0.0944  0.0933  0.0715  0.0827  185486.9
##
## sigma^2 estimated as 8.743e+11:  log likelihood = -4110.66,  aic = 8235.31
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

modelnt3 = arima(nt2,order = c(4,0,4))
summary(modelnt3)

```



```
##
## Call:
## arima(x = nt2, order = c(4, 0, 4))
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ma1          ma2          ma3          ma4
##      -0.3351  -0.0668   0.4435   0.8748   0.3602   0.2066  -0.4215  -0.7555
## s.e.   0.0642   0.0445   0.0438   0.0617   0.0846   0.0723   0.0726   0.0832
##      intercept
##      428477.6
## s.e.   227335.8
##
## sigma^2 estimated as 8.479e+11:  log likelihood = -4108.35,  aic = 8234.69
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

modelnt4 = arima(nt2,order = c(2,0,6))
summary(modelnt4)

##
## Call:
## arima(x = nt2, order = c(2, 0, 6))
##
## Coefficients:
##          ar1          ar2          ma1          ma2          ma3          ma4          ma5          ma6
##      0.1610   0.8069  -0.1439  -0.7545   0.0442   0.0902  -0.1189   0.0339
## s.e.   0.1288   0.1286   0.1412   0.1352   0.0765   0.0722   0.0789   0.0676
##      intercept
##      429835.7
## s.e.   230949.8
##
## sigma^2 estimated as 8.665e+11:  log likelihood = -4109.52,  aic = 8237.04
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN

modelnt5 = arima(nt2,order = c(4,0,6))
summary(modelnt5)
```

```
##
## Call:
## arima(x = nt2, order = c(4, 0, 6))
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ma1          ma2          ma3          ma4
##          0.0803 -0.1253  0.0792  0.8405 -0.0621  0.2099 -0.0461 -0.6878
## s.e.      0.0752  0.0776  0.0750  0.0788  0.0952  0.0968  0.1147  0.1241
##          ma5          ma6  intercept
##          -0.0420  0.1020  430074.2
## s.e.      0.0758  0.0875  196298.6
##
## sigma^2 estimated as 8.462e+11:  log likelihood = -4107.97,  aic = 8237.93
##
## Training set error measures:

## Warning in trainingaccuracy(object, test, d, D): test elements must be
within
## sample

##              ME RMSE MAE MPE MAPE
## Training set NaN   NaN NaN NaN  NaN

# Best Model :
#GARCH (2,2)
```

For the model identification, the ACF cut-off plot is found in lag 2, lag 4, and lag 6, while the PACF cut-off is in lag 2 and lag 4. Referring to the graph, ACF and PACF also have cut-offs on other lags, but they are not significant or can be considered too far. Time series variance was found following several GARCH (p, q) models with the residual squares following the ARMA model (max (p, q), p).

It is found that the model with the smallest AIC is ARMA (2,2). By estimating the parameters, the model **GARCH (2,2)** is found, but ω and the parameter for lag 1 are not significant.

```
# Signifikansi dari Koefisien Data Residual
coeftest(modelnt1)
```

```
##
## z test of coefficients:
##
##              Estimate Std. Error z value Pr(>|z|)
## ar1          7.6195e-02  6.1114e-02  1.2468  0.21248
## ar2          8.9114e-01  6.0309e-02 14.7763 < 2e-16 ***
## ma1         -7.5615e-02  7.9958e-02 -0.9457  0.34431
## ma2         -7.7050e-01  7.7200e-02 -9.9806 < 2e-16 ***
## intercept    4.2895e+05  2.3213e+05  1.8479  0.06462 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Menguji apakah masih ada efek heteroskedastik pada residual model nt2
ArchTest(residuals(modelnt1))
```

```
##
```

```
## ARCH LM-test; Null hypothesis: no ARCH effects
```

```
##
```

```
## data: residuals(modelnt1)
```

```
## Chi-squared = 1.9781, df = 12, p-value = 0.9994
```

```
# Tidak ada efek heteroskedastik
```

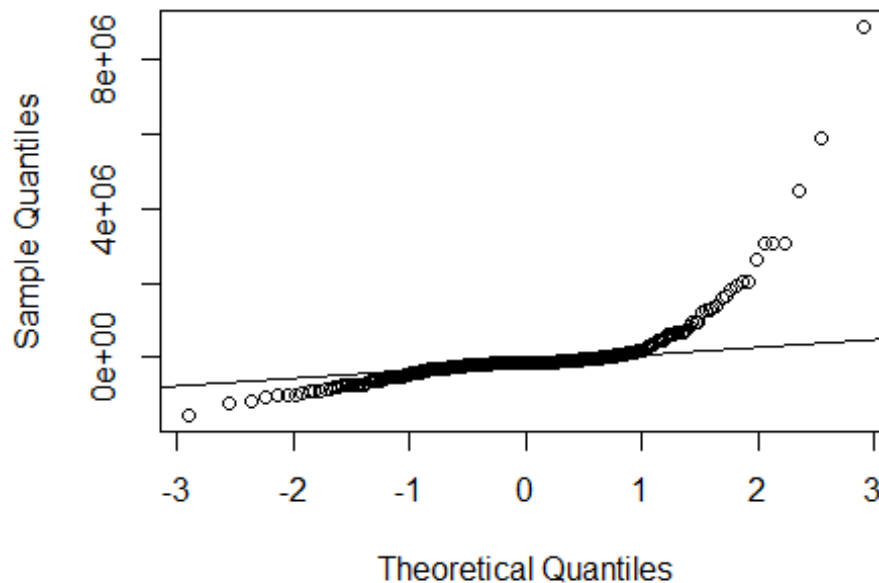
Furthermore, an ARCH-test was carried out to test whether there was still a heteroscedastic effect on the residuals of the nt2 model. Because the P-Value = 0.9994 > α = 0.05 then H0 is not rejected, which means that the GARCH model residual (2,2) does not contain heteroscedastic effects.

```
# Diagnostic Test of Squared residuals sequence's Model
```

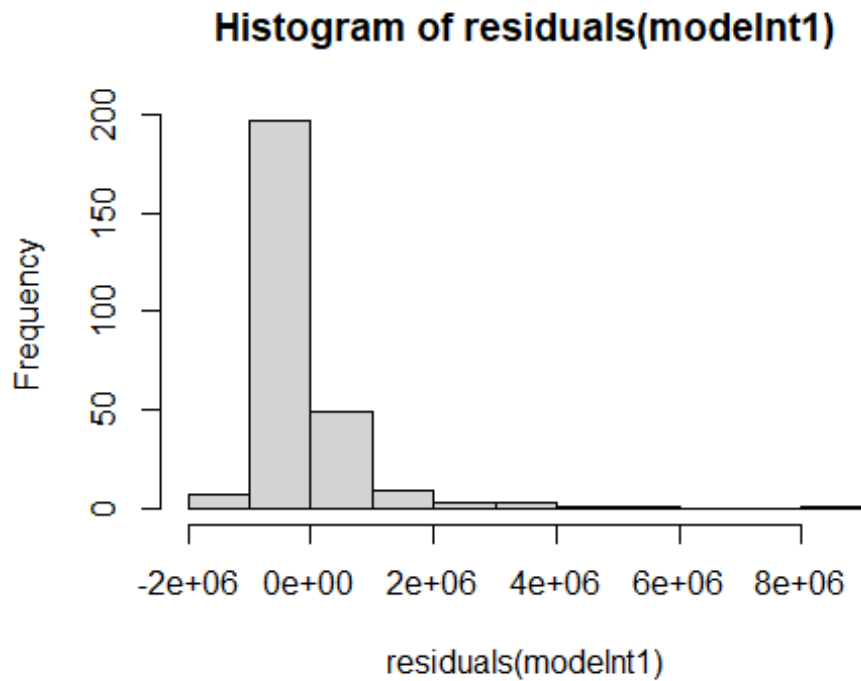
```
qqnorm(residuals(modelnt1))
```

```
qqline(residuals(modelnt1))
```

Normal Q-Q Plot



```
hist(residuals(modelnt1))
```



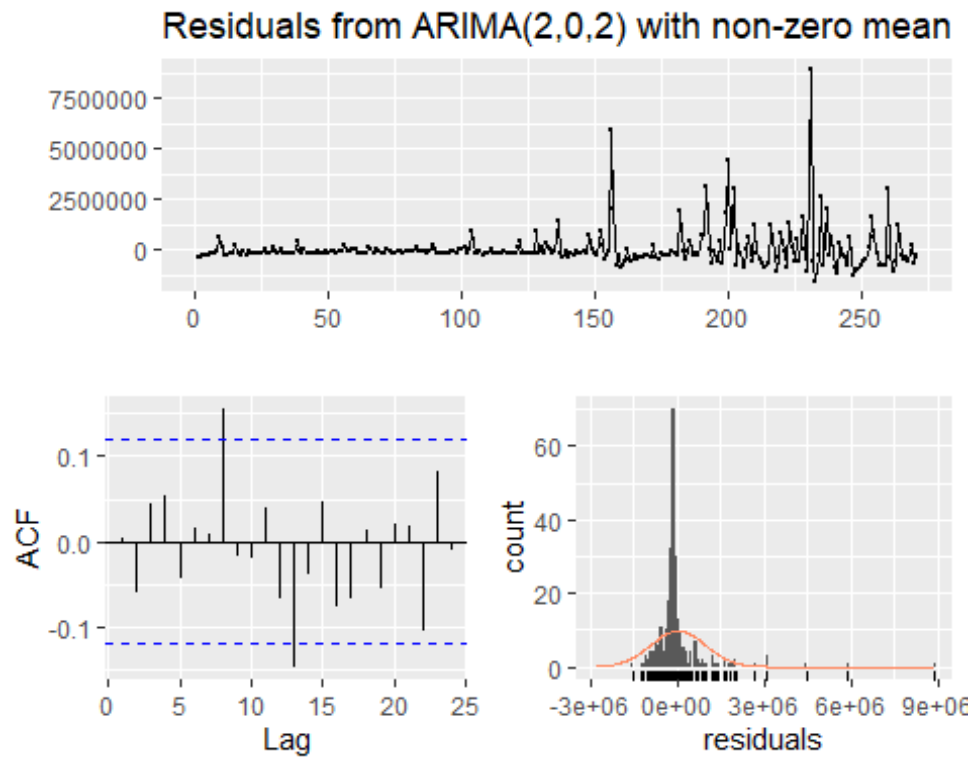
```
ks.test(residuals(modelInt1), "pnorm")

##
## One-sample Kolmogorov-Smirnov test
##
## data: residuals(modelInt1)
## D = 0.75277, p-value < 2.2e-16
## alternative hypothesis: two-sided

Box.test(residuals(modelInt1), lag = 100, type="Ljung-Box")

##
## Box-Ljung test
##
## data: residuals(modelInt1)
## X-squared = 125.05, df = 100, p-value = 0.04572

checkresiduals(modelInt1)
```



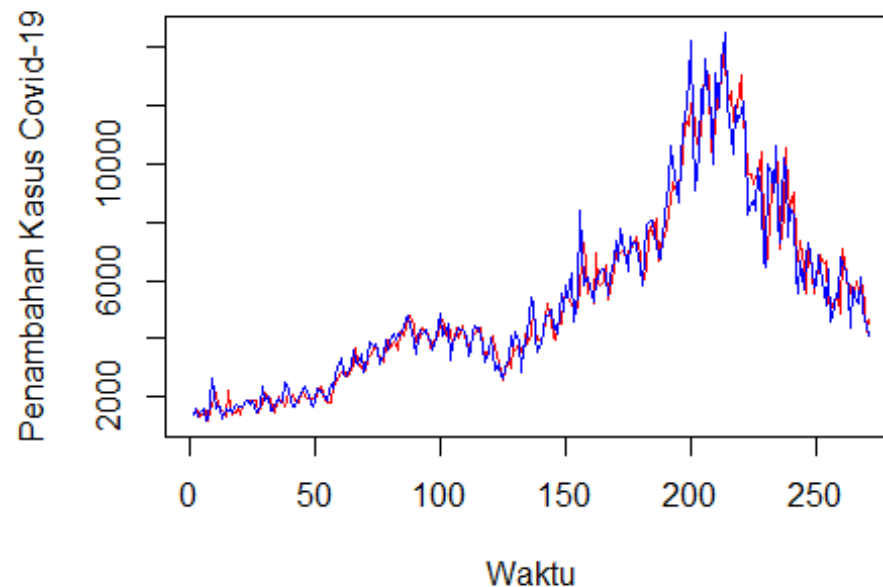
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,2) with non-zero mean
## Q* = 9.9115, df = 5, p-value = 0.07778
##
## Model df: 5.   Total lags used: 10
```

Forecasting

After the models has been predicted. Forecasting is conducted for the data.

```
#Data and Model comparison
data_pred <- indekssts - residuals(model_arma8)
ts.plot(data_pred,indekststs, xlab = 'Waktu', ylab = 'Penambahan Kasus Covid-
19',
        col = c('red', 'blue'), main = 'Perbandingan Data Kasus COVID-19 Asli
dengan Model')
```

Perbandingan Data Kasus COVID-19 Asli dengan Model

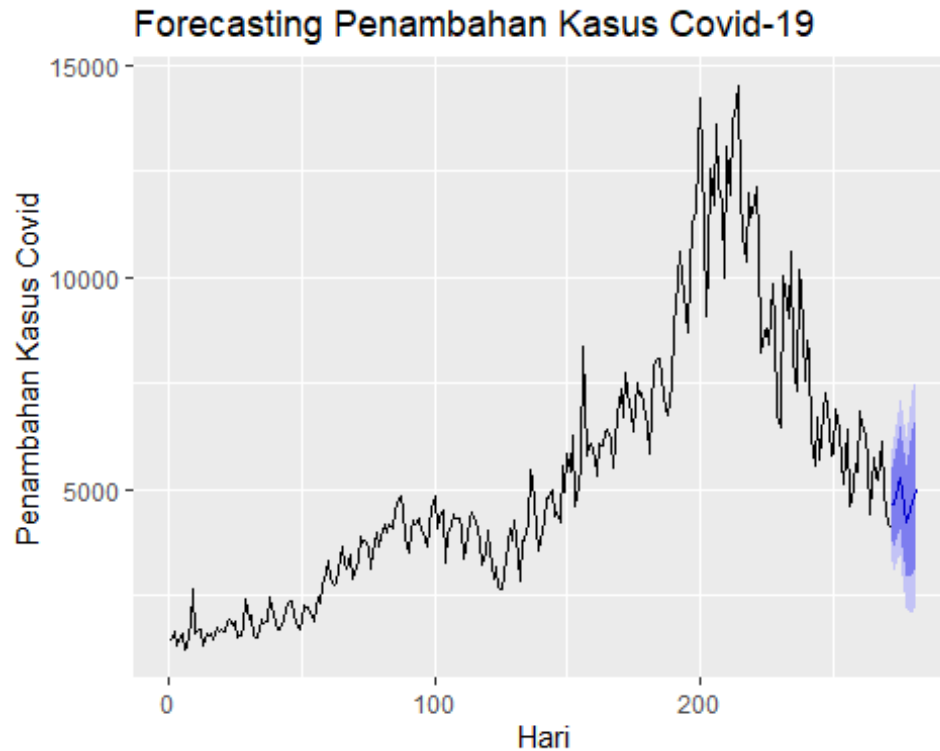


Forecasting

```
model_arima8 = stats::arima(indeks, order = c(5,1,5))  
(prediksi = forecast(model_arima8, h = 10))
```

##	Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 272	4627.451	3764.284	5490.618	3307.351	5947.551	
## 273	4657.379	3620.735	5694.022	3071.969	6242.788	
## 274	5064.106	3933.912	6194.301	3335.623	6792.590	
## 275	5283.242	4082.404	6484.080	3446.719	7119.765	
## 276	4909.780	3632.317	6187.243	2956.069	6863.491	
## 277	4222.641	2911.789	5533.493	2217.865	6227.416	
## 278	4349.054	2908.806	5789.303	2146.384	6551.724	
## 279	4508.583	2907.588	6109.578	2060.073	6957.094	
## 280	4674.213	2961.073	6387.353	2054.191	7294.235	
## 281	4955.913	3189.618	6722.209	2254.597	7657.229	

```
autoplot(prediksi, main="Forecasting Penambahan Kasus Covid-  
19", ylab="Penambahan Kasus Covid", xlab="Hari")
```



Conclusion

(see the IHSG data analysis)

1. The best model for the IHSG data is $ARIMA(2,1,0)$ and for the COVID-19 data is $ARIMA(5,1,5)$ with heteroscedastic effect $brti$ GARCH $(2,2)$.
2. By comparing both the data plot, we can see that the increasing number of Covid-19 cases in Indonesia had an impact on IHSG movement. At the 40th day until the 60th day, when the number of cases increased, the IHSG went down (Around August-September 2020). But after that, IHSG increased greatly though the number of cases also increased. Based on this, we can conclude that for certain moment, Covid-19 affects on Indonesia economy. But, after September 2020, Indonesia's economy is able to recover quickly and grow firmly regardless of the increasing number of Covid-19 cases.

From this conclusion, we encourage potential investors to take the opportunity and keep calm though the number of Covid-19 cases is still high. Because the models show us that Indonesia's economy could stand firm in Covid-19 pandemic period.