Paulinus Alan Sanjaya Jamlu

193114033

**PENYELESAIAN**

Untuk melakukan pemodelan dengan metode Box-Jenkin, akan dilakukan dengan 4 tahapan

besar, yaitu

1. Identifikasi Model,

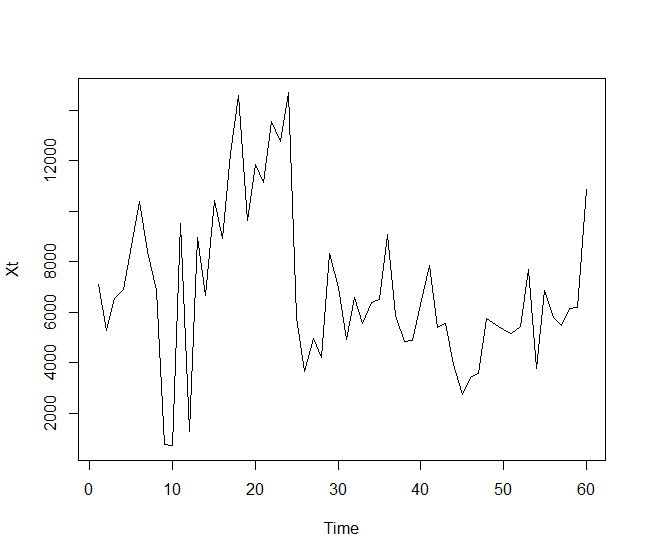
2. Pendugaan Model atau Estimasi Model,

3. Pemeriksaan Diagnostik Model, dan

4. Penggunaan Model Untuk Peramalan (forecasting)

1. **Identifikasi model**

Untuk tahap awal mengidentifikasi data yang akan di proses untuk mengetahui apakah mengandung tren atau musiman, dan akan ditunjukkan grafik plot, grafik ACF dan PACF dari data asli untuk mengetahui apakah data tersebut merupakan data stasioner atau tidak stasioner. Dan jika data sudah stasioner maka proses bisa dilanjutkan ke tahap berikutnya.



Dari plot di atas, terlihat bahwa Jumlah Wisatawan Dalam Negeri Hotel Bintang 1 di DIY tahun 2018 hingga 2022 menunjukkan bahwa data tidak stasioner.

Akan di analisis apakah data set Jumlah Wisatawan Dalam Negeri Hotel Bintang 1 di DIY tahun 2018 hingga 2022 stasioner menggunakan uji ADF (*augmented dicky fuller*)

1. ∶ Data Tidak Stasioner

∶ Data Stasioner

1. Tetapkan uji signifikansi 𝛼 = 0.05
2. Statistik uji

Menggunakan uji Augmented Dicky-Fuller

Dengan daerah penolaknya yaitu 𝐻0 ditolak jika p-value < 𝛼

1. Perhitungan

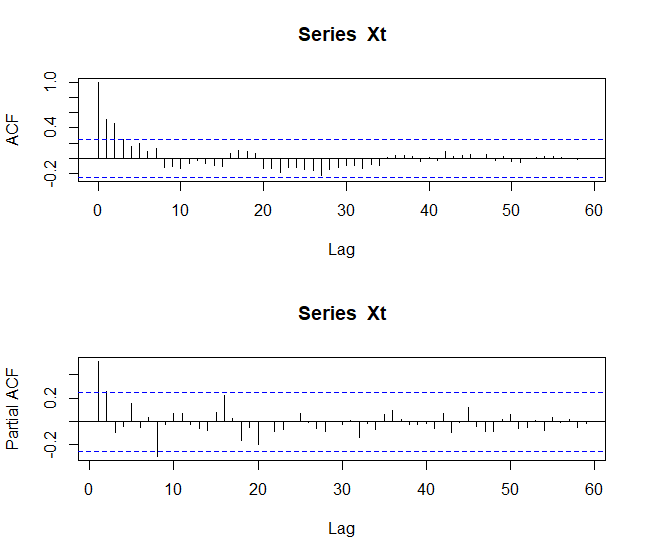
**p-value = 0.2515 > 𝛼 = 0.05**

1. Kesimpulan

Karena 𝑝 − 𝑣𝑎𝑙𝑢𝑒 = **0.2515** > 𝛼 = 0.05, sehingga 𝐻0 diterima sehingga

menghasilkan **Data tidak Stasioner**.

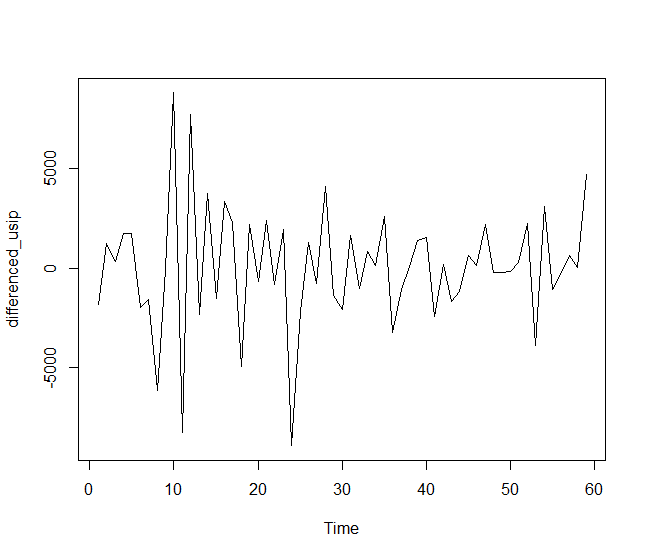
Akan dipastikan hasilnya menggunakan plot grafik acf dan pacf.



Dari grafik ACF, penurunan nilai terjadi dengan cepat, namun pada grafik PACF, mengindikasikan data set tidak stasioner karena merupakan grafik sinus teredam.

Karena data set di atas tidak stasioner, maka akan di-*differencing* untuk menstasionerkannya.

Setelah data di*-differencing*, berikut adalah grafiknya



Terlihat bahwa data sudah stasioner secara rata-rata dan variansinya.

Lalu akan diperiksa menggunakan uji adf.

Metode Uji Akar Unit (Uji Augmented Dickey Fuller).

Hipotesis yang di uji adalah:

1. 𝐻0 ∶ Data Tidak Stasioner

𝐻1 ∶ Data Stasioner

1. Tetapkan uji signifikansi 𝛼 = 0.05
2. Statistik uji

Menggunakan uji Augmented Dicky-Fuller

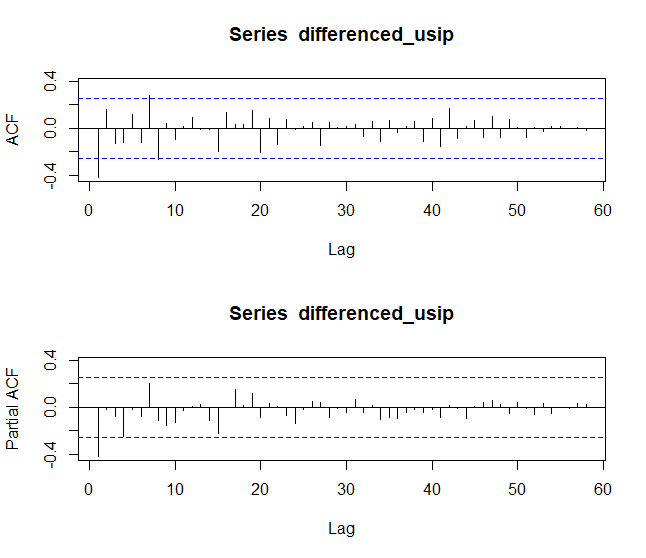
Dengan daerah penolakannya yaitu 𝐻0 ditolak jika p-value < 𝛼

1. Perhitungan

**p-value = 0.01< ∝ = 0.05**

1. Kesimpulan

Karena nilai <, maka ditolak sehingga hipotesis alternatif diterima (), **data stasioner**.

Akan dipastikan bahwa grafik ACF dan PACF juga menunjukkan hasil yang stasioner.

Mengacu pada grafik ACF dan PACF, tidak terlihat ada pola musiman pada data di atas, meskipun pada grafik terjadi pola naik turun, namun pola tersebut tidak berulang pada jangka waktu tertentu yang membuat data tersebut bukan termasuk data musiman.

1. **Estimasi Model**

Setelah di-*differencing*, akan dicari nilai p dan q

|  |  |  |  |
| --- | --- | --- | --- |
| model | | q | |
| p |  | 0 | 1 |
| 0 | (0,1,0) | (0,1,1) |
| 1 | (1,1,0) | (1,1,1) |

Setelah menentukan nilai p, d, q langkah berikutnya adalah mengestimasi parameter AR dan MA yang dimasukkan dalam model dengan menggunakan program R.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MODEL | KELAYAKAN | WHITE NOISE | DISTRIBUSI NORMAL | AIC |
| ARIMA (0,1,0) | TIDAK LAYAK | TIDAK | TIDAK | 1115.53 |
| ARIMA (1,1,0) | TIDAK LAYAK | YA | TIDAK | 1105.94 |
| ARIMA (0,1,1) | LAYAK | YA | YA | 1105.75 |
| ARIMA (1,1,1) | LAYAK | YA | YA | 1107.12 |

**Uji signifikansi parameter model**

Menentukan apakah model ARIMA memiliki parameter model yang signifikan atau tidak, hal ini dapat dilihat dari p-value. Apabila 𝑝 − 𝑣𝑎𝑙𝑢𝑒 < 𝛼 = 0.05, maka parameter dari model tersebut signifikan. Berdasarkan kriteria residual white noise, residual model berdistribusi normal, dan AIC terkecil, maka yang memenuhi kriteria tersebut adalah ARIMA (0,1,0). Jadi kita akan mengecek apakah ARIMA (0,1,0) memenuhi signifikansi parameter model atau tidak.

1. Hipotesis

𝐻0 ∶ Model tidak signifikan

𝐻1 ∶ model signifikan

1. Tetapkan uji signifikansi 𝛼 = 0.05
2. Statistik uji

Menggunakan uji Augmented Dicky-Fuller

Dengan daerah penolakannya yaitu 𝐻0 ditolak jika p-value < 𝛼

1. Perhitungan

**P-value = 0.0007 < 0.05.**

1. Kesimpulan

Karena nilai <, maka ditolak sehingga hipotesis alternatif diterima (), **model signifikan**. Sehingga model ARIMA (0,1,1) merupakan model terbaik untuk data tersebut.

1. **Persamaan matematika**

Berikut adalah persamaan matematika untuk mode ARIMA (0,1,1)

ma1 = -0.4850 =

Substitusi nilai ma1

1. **Peramalan**

Menggunakan program R untuk meramal 7 bulan kedepan

Model ARIMA (0,1,1)

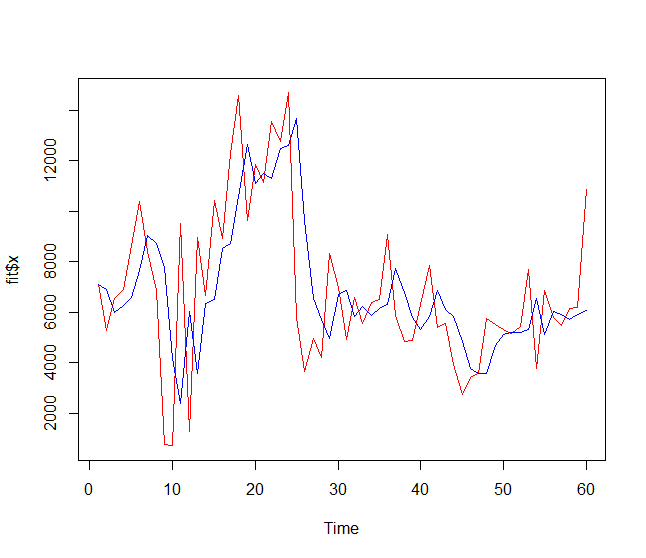
|  |  |
| --- | --- |
| Bulan | Jumlah Wisatawan |
| Januari | 8528.797 |
| Februari | 8528.797 |
| Maret | 8528.797 |
| April | 8528.797 |
| Mei | 8528.797 |
| Juni | 8528.797 |
| Juli | 8528.797 |

Karena peramalan dengan model ARIMA (0,1,1) terlihat tidak sesuai karena merupakan nilai konstan, kita akan menggunakan model terbaik ke-2 yaitu ARIMA (1,1,1)

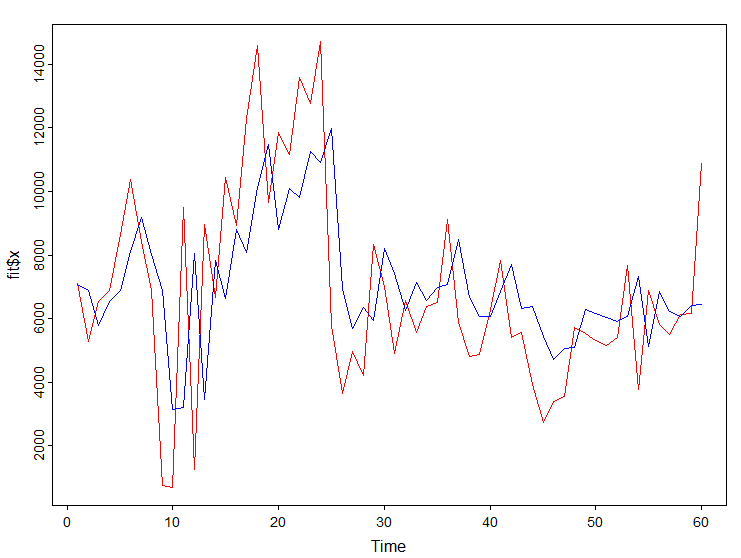
|  |  |
| --- | --- |
| Bulan | Jumlah Wisatawan |
| Januari | 9054.743 |
| Februari | 8075.582 |
| Maret | 7545.372 |
| April | 7258.266 |
| Mei | 7102.8 |
| Juni | 7018.616 |
| Juli | 6973.031 |

1. **Perbandingan**

**ARIMA (0,1,1)**



**ARIMA (1,1,1)**

****

1. **Kesimpulan**

Model ARIMA (0,1,1) merupakan model terbaik untuk data deret waktu kunjungan wisata di Jogja dari data yang sudah tersedia dari tahun 2018 hingga 2022. Namun, Model Arima ini tidak cukup baik untuk memprediksi jumlah wisatawan untuk 7 bulan kedepan. Hal ini bisa disebabkan oleh anomali pada 2019 hingga 2022 karena pandemi virus corona sehingga data wisatawan tidak dapat diprediksi dengan baik.

**Lampiran**

library(forecast)

Warning: package 'forecast' was built under R version 4.2.3

Registered S3 method overwritten by 'quantmod':  
 method from  
 as.zoo.data.frame zoo

library(tseries)

Warning: package 'tseries' was built under R version 4.2.3

1  
usip = read.csv(file = "D:/Semester VIII/Analisis Runtun Waktu/uts 2/DATA USIP2.csv", header = TRUE, sep = ";")  
attach(usip)  
usip

TAHUN BULAN JUMLAH  
 1 2018 jan 7092  
 2 2018 feb 5278  
 3 2018 mar 6525  
 4 2018 apr 6888  
 5 2018 mei 8646  
 6 2018 jun 10388  
 7 2018 jul 8417  
 8 2018 agu 6877  
 9 2018 sep 743  
 10 2018 okt 698  
 11 2018 nov 9505  
 12 2018 des 1272  
 13 2019 jan 8970  
 14 2019 feb 6676  
 15 2019 mar 10427  
 16 2019 apr 8933  
 17 2019 mei 12299  
 18 2019 jun 14578  
 19 2019 jul 9651  
 20 2019 agu 11844  
 21 2019 sep 11162  
 22 2019 okt 13570  
 23 2019 nov 12757  
 24 2019 des 14699  
 25 2020 jan 5785  
 26 2020 feb 3636  
 27 2020 mar 4956  
 28 2020 apr 4208  
 29 2020 mei 8336  
 30 2020 jun 6996  
 31 2020 jul 4907  
 32 2020 agu 6570  
 33 2020 sep 5562  
 34 2020 okt 6383  
 35 2020 nov 6501  
 36 2020 des 9101  
 37 2021 jan 5871  
 38 2021 feb 4822  
 39 2021 mar 4889  
 40 2021 apr 6287  
 41 2021 mei 7837  
 42 2021 jun 5397  
 43 2021 jul 5571  
 44 2021 agu 3924  
 45 2021 sep 2737  
 46 2021 okt 3406  
 47 2021 nov 3552  
 48 2021 des 5735  
 49 2022 jan 5525  
 50 2022 feb 5311  
 51 2022 mar 5144  
 52 2022 apr 5417  
 53 2022 mei 7681  
 54 2022 jun 3787  
 55 2022 jul 6879  
 56 2022 agu 5802  
 57 2022 sep 5492  
 58 2022 okt 6146  
 59 2022 nov 6176  
 60 2022 des 10863

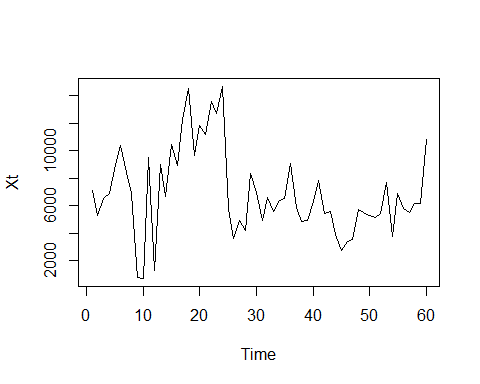
Xt = (usip$JUMLAH)  
Xt

[1] 7092 5278 6525 6888 8646 10388 8417 6877 743 698 9505 1272  
 [13] 8970 6676 10427 8933 12299 14578 9651 11844 11162 13570 12757 14699  
 [25] 5785 3636 4956 4208 8336 6996 4907 6570 5562 6383 6501 9101  
 [37] 5871 4822 4889 6287 7837 5397 5571 3924 2737 3406 3552 5735  
 [49] 5525 5311 5144 5417 7681 3787 6879 5802 5492 6146 6176 10863

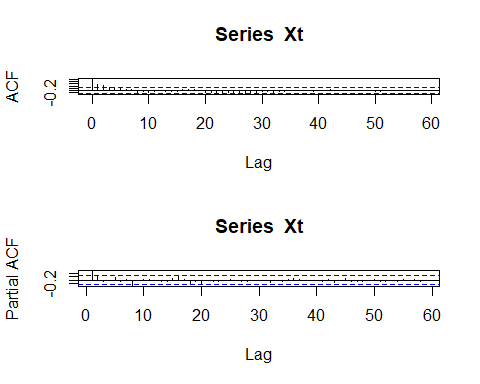
# Melakukan tes ADF pada data  
adf.test(Xt)

Augmented Dickey-Fuller Test  
   
 data: Xt  
 Dickey-Fuller = -2.7998, Lag order = 3, p-value = 0.2515  
 alternative hypothesis: stationary

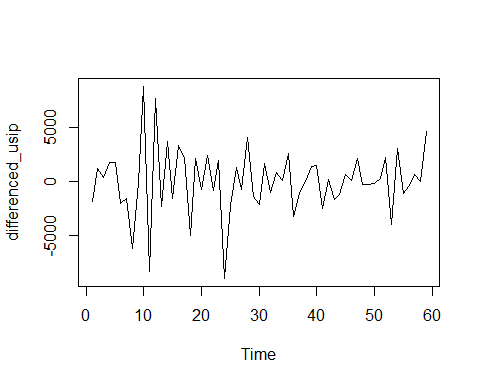
# Menampilkan plot data  
par(mfrow=c(1,1))  
plot.ts(Xt)



# Menampilkan plot ACF dan PACF  
par(mfrow=c(2,1))  
acf(Xt, lag.max = 120)  
pacf(Xt, lag.max = 120)



# Melakukan diferensiasi pada data untuk membuatnya stasioner  
differenced\_usip <- diff(Xt)  
  
# Menampilkan plot data yang sudah didiferensiasi  
par(mfrow=c(1,1))  
plot.ts(differenced\_usip)

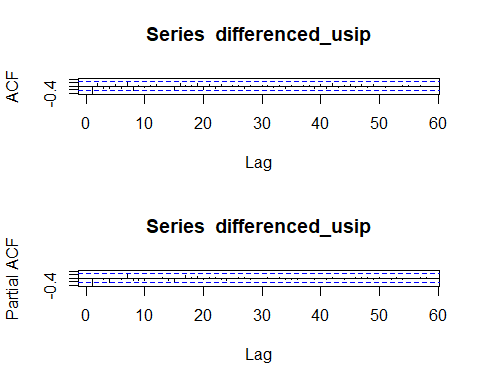


# Melakukan tes ADF pada data yang sudah didiferensiasi  
adf.test(differenced\_usip)

Warning in adf.test(differenced\_usip): p-value smaller than printed p-value

Augmented Dickey-Fuller Test  
   
 data: differenced\_usip  
 Dickey-Fuller = -5.2144, Lag order = 3, p-value = 0.01  
 alternative hypothesis: stationary

# Menampilkan plot ACF dan PACF dari data yang sudah didiferensiasi  
par(mfrow=c(2,1))  
Acf(differenced\_usip, lag.max = 120)  
Pacf(differenced\_usip, lag.max = 120)



fit = auto.arima(Xt)  
summary(fit)

Series: Xt   
 ARIMA(0,1,1)   
   
 Coefficients:  
 ma1  
 -0.4855  
 s.e. 0.1349  
   
 sigma^2 = 7635568: log likelihood = -550.87  
 AIC=1105.75 AICc=1105.96 BIC=1109.9  
   
 Training set error measures:  
 ME RMSE MAE MPE MAPE MASE  
 Training set 54.28668 2716.809 1949.862 -33.80457 55.76716 0.9057632  
 ACF1  
 Training set -0.001985753

estimasi  
#estimasi2  
estimasi2=arima(Xt,order=c(0,1,0))  
estimasi2

Call:  
 arima(x = Xt, order = c(0, 1, 0))  
   
   
 sigma^2 estimated as 9206750: log likelihood = -556.76, aic = 1115.53

residual1=resid(estimasi2)  
shapiro.test(residual1)

Shapiro-Wilk normality test  
   
 data: residual1  
 W = 0.94845, p-value = 0.01317

Box.test(residual1,lag=6,type="Ljung-Box")

Box-Ljung test  
   
 data: residual1  
 X-squared = 16.507, df = 6, p-value = 0.01128

#estimasi2  
estimasi2=arima(Xt,order=c(1,1,0))  
estimasi2

Call:  
 arima(x = Xt, order = c(1, 1, 0))  
   
 Coefficients:  
 ar1  
 -0.4283  
 s.e. 0.1192  
   
 sigma^2 estimated as 7538668: log likelihood = -550.97, aic = 1105.94

residual2=resid(estimasi2)  
shapiro.test(residual2)

Shapiro-Wilk normality test  
   
 data: residual2  
 W = 0.95066, p-value = 0.01673

Box.test(residual2,lag=6,type="Ljung-Box")

Box-Ljung test  
   
 data: residual2  
 X-squared = 4.3917, df = 6, p-value = 0.6238

#estimasi3  
estimasi3=arima(Xt,order=c(0,1,1))  
estimasi3

Call:  
 arima(x = Xt, order = c(0, 1, 1))  
   
 Coefficients:  
 ma1  
 -0.4855  
 s.e. 0.1349  
   
 sigma^2 estimated as 7506151: log likelihood = -550.87, aic = 1105.75

residual3=resid(estimasi3)  
shapiro.test(residual3)

Shapiro-Wilk normality test  
   
 data: residual3  
 W = 0.96369, p-value = 0.07141

Box.test(residual3,lag=6,type="Ljung-Box")

Box-Ljung test  
   
 data: residual3  
 X-squared = 4.2163, df = 6, p-value = 0.6474

#estimasi4  
estimasi4=arima(Xt,order=c(1,1,1))  
estimasi4

Call:  
 arima(x = Xt, order = c(1, 1, 1))  
   
 Coefficients:  
 ar1 ma1  
 0.5415 -0.9842  
 s.e. 0.1574 0.1784  
   
 sigma^2 estimated as 7195012: log likelihood = -550.56, aic = 1107.12

residual4=resid(estimasi4)  
shapiro.test(residual4)

Shapiro-Wilk normality test  
   
 data: residual4  
 W = 0.96979, p-value = 0.1425

Box.test(residual4,lag=6,type="Ljung-Box")

Box-Ljung test  
   
 data: residual4  
 X-squared = 6.8429, df = 6, p-value = 0.3356

3. Persamaan  
# ARMA (0,1,1)  
library(astsa)

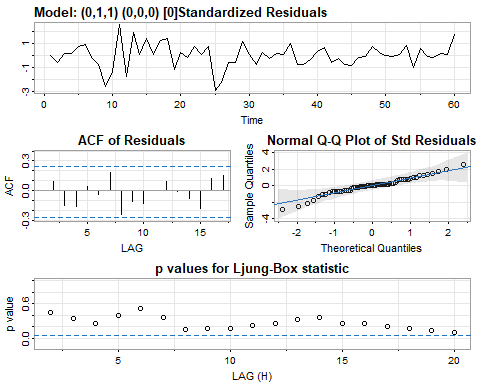
Warning: package 'astsa' was built under R version 4.3.0

Attaching package: 'astsa'

The following object is masked from 'package:forecast':  
   
 gas

Xt3<-sarima(Xt,0,1,1,0,0,0,0)

initial value 8.017502   
 iter 2 value 7.918634  
 iter 3 value 7.917425  
 iter 4 value 7.916216  
 iter 5 value 7.915639  
 iter 6 value 7.915633  
 iter 7 value 7.915633  
 iter 7 value 7.915633  
 iter 7 value 7.915633  
 final value 7.915633   
 converged  
 initial value 7.917696   
 iter 2 value 7.917672  
 iter 3 value 7.917639  
 iter 4 value 7.917628  
 iter 4 value 7.917628  
 iter 4 value 7.917628  
 final value 7.917628   
 converged



Xt3$ttable

Estimate SE t.value p.value  
 ma1 -0.4850 0.1347 -3.6010 0.0007  
 constant 33.0827 187.0056 0.1769 0.8602

auto.arima(Xt)

Series: Xt   
 ARIMA(0,1,1)   
   
 Coefficients:  
 ma1  
 -0.4855  
 s.e. 0.1349  
   
 sigma^2 = 7635568: log likelihood = -550.87  
 AIC=1105.75 AICc=1105.96 BIC=1109.9

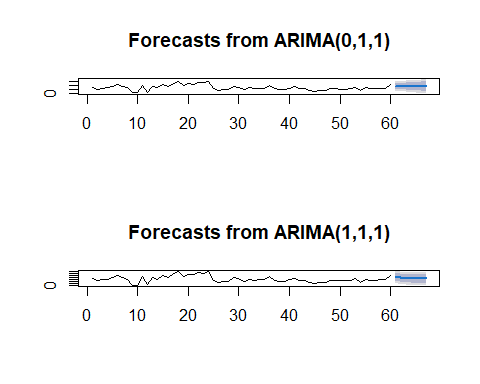
4. Peramalan  
library(forecast)  
# ARIMA (0,1,1)  
fit <- Arima(Xt, order = c(0, 1, 1))  
forecasted\_values <- forecast(fit, h = 7)  
forecasted\_values

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
 61 8528.797 4987.545 12070.05 3112.9201 13944.67  
 62 8528.797 4546.312 12511.28 2438.1112 14619.48  
 63 8528.797 4149.309 12908.28 1830.9477 15226.65  
 64 8528.797 3785.418 13272.18 1274.4248 15783.17  
 65 8528.797 3447.520 13610.07 757.6550 16299.94  
 66 8528.797 3130.733 13926.86 273.1699 16784.42  
 67 8528.797 2831.532 14226.06 -184.4177 17242.01

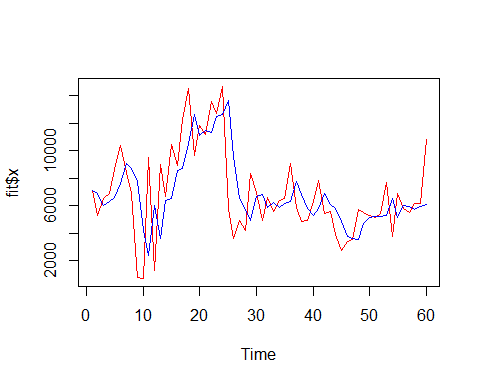
# Plot  
plot(forecasted\_values)  
  
# ARIMA (1,1,1)  
fit <- Arima(Xt, order = c(1, 1, 1))  
forecasted\_values <- forecast(fit, h = 7)  
forecasted\_values

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
 61 9054.743 5548.653 12560.83 3692.6407 14416.85  
 62 8075.582 4053.742 12097.42 1924.7082 14226.46  
 63 7545.372 3365.790 11724.95 1153.2524 13937.49  
 64 7258.266 3023.054 11493.48 781.0684 13735.46  
 65 7102.800 2845.214 11360.39 591.3842 13614.22  
 66 7018.616 2750.594 11286.64 491.2398 13545.99  
 67 6973.031 2699.298 11246.76 436.9200 13509.14

# Plot  
plot(forecasted\_values)



5. Perbandingan  
# ARIMA (0,1,1)  
library(forecast)  
par(mfrow=c(1,1))  
fit<-Arima(Xt,order=c(0,1,1))  
plot.ts(fit$x,col="red")  
lines(fitted(fit),col="blue")



#ARIMA (1,1,1)  
library(forecast)  
par(mfrow=c(1,1))  
fit<-Arima(Xt,order=c(1,1,1))  
plot.ts(fit$x,col="red")  
lines(fitted(fit),col="blue")

