Tugas Besar AK 2281 Analisis Deret Waktu - Analisis Kualitas Udara Berbasis Indeks Standar Pencemaran Udara (ISPU) di DKI Jakarta

Tabina Wan Kedana[1, a)], Antonius[1, b)], Pamella Cathryn[1, c)], Jeremy[1, d)]

[1]*Institut Teknologi Bandung, Jl. Ganesa No.10, Lb. Siliwangi, Kecamatan Coblong, Kota Bandung, Jawa Barat, Indonesia*

Email: [a)]10820002@mahasiswa.itb.ac.id, [b)]10820007@mahasiswa.itb.ac.id, [c)]10820033@mahasiswa.itb.ac.id, [d)]10820034@mahasiswa.itb.ac.id

Abstract

The issue of air pollution is closely related to the 7th Sustainable Development Goal (SDG), namely Clean and Affordable Energy, and the 13th SDG, namely Climate Change Management. According to a report from sciencedirect.com, air pollutant emissions will cause 40% more deaths than today, carbon emissions will increase by 0.4% annually, and almost 2 billion people will have difficulty getting clean air. This study aims to (1) Model the amount of PM10 particulates in DKI Jakarta using the SARIMA model, (2) Predict the amount of PM10 particulates in DKI Jakarta for two years after October 2021, and (3) Determine the air quality at several points in DKI Jakarta (Bundaran HI, Kelapa Gading, Jagakarsa, and Lubang Buaya). The research method used is literature study and SARIMA time series data modeling. The forecasting results show that DKI Jakarta has air quality that is classified as moderate.

Keywords*: Air Pollution, PM10, SARIMA.*

Abstrak

Masalah Pencemaran Udara berkaitan erat dengan *Sustainable Development Goal* (SDG) ke-7, yaitu Energi Bersih dan Terjangkau, dan SDG ke-13, yaitu Penanganan Perubahan Iklim. Dilansir dari sciencedirect.com, Emisi polutan udara akan menyebabkan 40% lebih banyak kematian dini daripada saat ini, emisi karbon akan meningkat sebesar 0,4% tiap tahunnya, dan hampir 2 miliar orang akan kesusahan mendapatkan udara bersih. Penelitian ini bertujuan untuk (1) Memodelkan jumlah partikulat PM10 di DKI Jakarta menggunakan model SARIMA, (2) Melakukan prediksi jumlah partikulat PM10 di DKI Jakarta selama dua tahun setelah Oktober 2021, dan (3) Mengetahui kategori kualitas udara di beberapa titik DKI Jakarta (Bundaran HI, Kelapa Gading, Jagakarsa, dan Lubang Buaya). Metode penelitian yang digunakan adalah studi literatur dan pemodelan data deret waktu SARIMA. Hasil prakiraan menunjukkan DKI Jakarta memiliki kualitas udara yang masuk dalam klasifikasi sedang.

Kata-kata kunci: Pencemaran Udara, PM10, SARIMA.

PENDAHULUAN

Perubahan iklim adalah salah satu masalah yang belum dapat diselesaikan oleh manusia. Masalah tersebut diperparah dengan kenaikan emisi polutan udara akibat dari revolusi industri dan kemajuan teknologi. Penelitian ini dilakukan dengan alasan perubahan iklim akibat kemajuan teknologi dan faktor-faktor, seperti Jakarta merupakan salah satu kota paling berpolusi di dunia lalu dianalisis dari tingkat PM10 di Jakarta. *Sustainable Development Goals* (SDGs) yang terkait dari penelitian yang dilakukan adalah SDG ke-7, yaitu Energi Bersih dan Terjangkau, dan SDG ke-13, yaitu Penanganan Perubahan Iklim. data yang digunakan adalah data Indeks Standar Pencemaran Udara (ISPU) di DKI Jakarta yang diambil dari Jakarta Open Data. Data tersebut dipilih karena data tersebut memiliki data PM10 yang lengkap dan ekstensif.

Salah satu kajian literatur laporan ini adalah penelitian yang dilakukan oleh D.A Suryanto pada tahun 2012 yang berjudul Analisis Tingkat Polusi Udara Terhadap Pengaruh Pertumbuhan Kendaraan Studi Kasus DKI Jakarta yang dikutip dari *e-journal Gunadarma*. penelitian ini meneliti tingkat polusi udara di Jakarta dari sisi polutan yang dihasilkan oleh kendaraan bermotor. Penelitian ini pun menghasilkan model yang bergantung pada CO (karbon monoksida), HC (hidrokarbon), dan NO (natrium monoksida). Hasil dari penelitian tersebut menyimpulkan bahwa terdapat relasi hasil tingkat polusi udara terhadap pengaruh pertumbuhan kendaraan. Berbeda dengan penelitian yang telah dilakukan oleh D.A Suryanto tersebut kami menganalisis kualitas udara dari sisi PM10. PM10 tidak mengukur partikel-partikel seperti CO, HC, dan NO, tetapi PM10 terdiri dari partikel-partikel yang memiliki diameter kurang dari 10 mikron dapat membahayakan bagi manusia dan mengakibatkan gejala batuk-batuk atau bahkan asma. Perbedaan ini dapat menghasilkan analisis yang berbeda dan melihat kualitas udara di Jakarta dari sudut pandang yang baru. Kami menganalisis kualitas udara berbasis PM10 dengan tujuan memprediksi tingkat PM10 di Jakarta dan menghasilkan rekomendasi-rekomendasi yang baik dilakukan untuk mengurangi tingkat PM10 tersebut.

METODOLOGI

A. Teori Pendukung

Model SARIMA (Seasonal ARIMA) merupakan salah satu model untuk merepresentasikan data deret waktu yang memiliki perilaku musiman yang kuat. Perilaku musiman dapat diartikan sebagai sebuah pola perubahan yang tetap dan berulang selama periode waktu S, di mana S didefinisikan sebagai periode waktu sampai pola berulang kembali. Dalam model ARIMA musiman, model AR musiman memprediksi Y_t menggunakan nilai data pada lag dengan kelipatan S (rentang musiman), sedangkan model MA musiman memprediksi y_t menggunakan *error* dengan lag yang merupakan kelipatan S.

Perilaku musiman pun seringkali menyebabkan data menjadi tidak stasioner. Oleh karena itu, dilakukan diferensiasi musiman (*seasonal differencing*) yang didefinisikan sebagai selisih antara nilai variabel dengan nilai variabel pada lag kelipatan S.

Model SARIMA menggabungkan faktor nonmusiman dan musiman dalam model multiplikatif. Notasi yang digunakan untuk model ini adalah sebagai berikut.

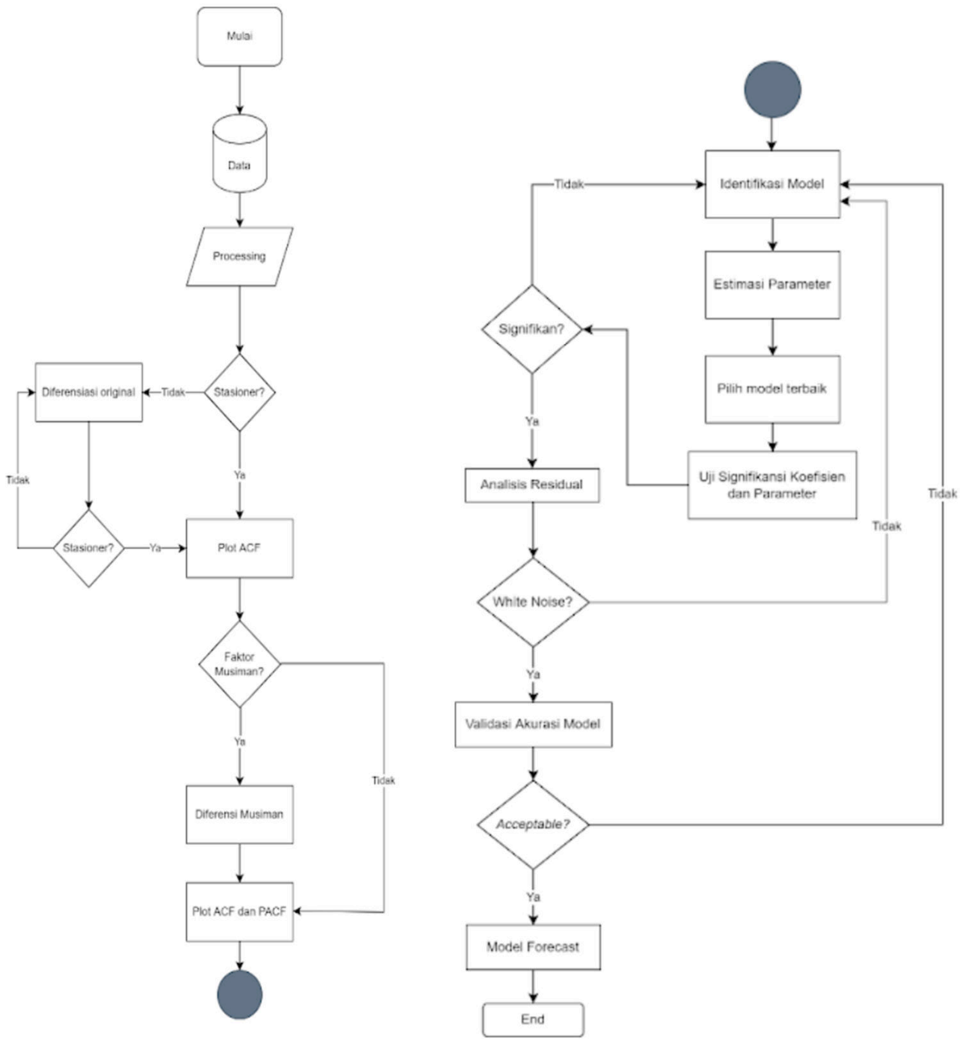
$$SARIMA(p, d, q) \times (P, D, Q)_S$$

Dengan keterangan dari persamaan tersebut sebagai berikut.

1. p = orde dari model AR
2. d = orde diferensiasi nonmusiman
3. q = orde dari model MA
4. P = orde dari model AR musiman
5. D = orde diferensiasi musiman
6. Q = orde dari model MA musiman
7. S = rentang musiman

B. Tahapan Penelitian

Berikut merupakan diagram alir dari langkah-langkah penelitian.



GAMBAR 1. *Flowchart* Penelitian

BAHAN DAN DATA

Lokasi Penelitian

Data diperoleh dari Data Indeks Standar Pencemaran Udara (ISPU) di DKI Jakarta tahun 2010 hingga 2021, didapat dari https://data.jakarta.go.id/. Lokasi penelitian dari data yang digunakan adalah DKI Jakarta, khususnya Bundaran HI, Kelapa Gading, Jagakarsa, dan Lubang Buaya. Ruang lingkup data.

Ruang Lingkup Data

Penelitian ini mengambil objek penelitian variabel PM10 pada Indeks Standar Pencemaran Udara (ISPU) di DKI Jakarta yang terdiri dari 4 stasiun. Adapun stasiun yang dimaksud adalah sebagai berikut.
1. DKI1 adalah daerah Bundaran HI (Jakarta Pusat).
2. DKI2 adalah daerah Kelapa Gading (Jakarta Utara).
3. DKI3 adalah daerah Jagakarsa (Jakarta Utara).
4. DKI4 adalah daerah Lubang Buaya (Jakarta Timur).

Data Primer

Berikut adalah cuplikan data mentah PM10 yang merupakan data keseluruhan per hari yang diperoleh dari https://data.jakarta.go.id/.

TABEL 1. Cuplikan Data Mentah

tanggal	stasiun	PM10	SO2	CO	O3	NO2	max	critical	categori
2016-01-01	DKI1 (Bunderan HI)	59	19	27	31	1	59	PM10	SEDANG
2016-01-02	DKI1 (Bunderan HI)	52	19	---	33	2	52	PM10	SEDANG
2016-01-03	DKI1 (Bunderan HI)	44	19	---	36	2	44	PM10	BAIK
2016-01-04	DKI1 (Bunderan HI)	58	21	---	46	5	58	PM10	SEDANG
2016-01-05	DKI1 (Bunderan HI)	70	19	---	41	4	70	PM10	SEDANG
2016-01-06	DKI1 (Bunderan HI)	69	19	---	19	4	69	PM10	SEDANG

Berikut adalah cuplikan data bersih, data bersih yang digunakan merupakan rata-rata per bulan dari data mentah.

TABEL 2. Cuplikan Data Bersih

bulan/tahun	PM10	SO2	CO	O3	NO2	max
01/2010	44	4.871	34.129	20.903	13.903	44.645
02/2010	53.643	5.964	40.571	38.607	17.786	57.929
03/2010	53.032	7.742	37.806	41.387	18.548	57.903
04/2010	52	6.167	34.233	38.5	17.833	56.267
05/2010	52.032	14.839	36.677	52.613	19.194	65.452
06/2010	57.333	16.867	40.467	38.533	19.633	62.2

Dengan keterangan sebagai berikut.

1. Tanggal : Tanggal pengukuran kualitas udara.
2. Stasiun : Lokasi pengukuran di stasiun (DKI1-DKI4).
3. PM10 : Partikulat salah satu parameter yang diukur.
4. SO2 : Sulfida (dalam bentuk SO2) salah satu parameter yang diukur.
5. CO : Karbon Monoksida salah satu parameter yang diukur.
6. O3 : Ozon salah satu parameter yang diukur.
7. NO2 : Nitrogen dioksida salah satu parameter yang diukur.
8. max : Nilai ukur paling tinggi seluruh parameter dalam waktu yang sama

Sumber Data

Penelitian ini menggunakan data yang diperoleh dari suatu *website* yang menyediakan beragam jenis data dan dapat diakses secara bebas. Data yang digunakan dalam penelitian ini diperoleh dari https://data.jakarta.go.id/. Adapun yang menjadi sumber data primer dalam penelitian ini adalah konsentrasi PM10 pada Indeks Standar Pencemaran Udara (ISPU) di DKI Jakarta tahun 2010 hingga 2021.

Variabel Penelitian

Variabel yang difokuskan untuk diteliti pada penelitian kali ini adalah PM10. Hasil pengolahan data dari konsentrasi PM10 ini dapat dipengaruhi oleh konsentrasi PM10 di masa sebelumnya dan galat.

Teknik Pengumpulan Data

Untuk memperoleh data yang dikehendaki sesuai dengan permasalahan dalam penelitian ini menggunakan metode angket dengan melakukan observasi terhadap konsentrasi PM10 pada Indeks Standar Pencemaran Udara (ISPU) di DKI Jakarta tahun 2010 hingga 2021.

METODE PENELITIAN

A. *Autocorrelation Function* (ACF)
Fungsi Autokorelasi atau yang lebih dikenal dengan ACF merupakan koefisien korelasi antara Y_t dan Y_{t-k} pada lag ke k. ACF biasanya dinotasikan dengan ρ_k dan didefinisikan sebagai berikut. (Wei, 2006)

$$\rho_k = \frac{Cov(Y_t, Y_{t-k})}{\sqrt{Var(Y_t)Var(Y_{t-k})}} = \frac{Cov(Y_t, Y_{t-k})}{Var(Y_t)} = \frac{\gamma_k}{\gamma_0}$$

B. *Partial Autocorrelation Function* (PACF)
Fungsi Autokorelasi Parsial atau PACF merupakan sebuah fungsi dari ACF yang digunakan untuk menentukan orde p dari model AR. Pada analisis deret waktu, PACF dinotasikan dengan ϕ_{kk} dengan persamaan sebagai berikut. (Wei, 2006)

$$\hat{\phi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^{k} \hat{\phi}_{kj}\, \hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k} \hat{\phi}_{kj}\, \hat{\rho}_j}$$

Dengan keterangan sebagai berikut.
$\hat{\rho}_k$: *sample* ACF lag-k
$\hat{\phi}_{kk}$: *sample* ACF lag-k

C. Uji Kestasioneran: Uji *Augmented Dickey-Fuller* (ADF)
Uji *Augmented Dickey-Fuller* (ADF) memasukkan adanya autokorelasi di dalam variabel gangguan dengan memasukkan variabel independen berupa kelambanan diferensi. Hipotesis uji stasioner data deret waktu menggunakan ADF adalah:
H_0: data tidak stasioner
H_1: data stasioner

D. Uji Diagnostik: Uji Ljung-Box
Uji Ljung-Box digunakan untuk menguji independensi residual antar lag pada model. Hipotesis yang digunakan adalah:
H_0: residual saling bebas
H_1: residual tidak saling bebas

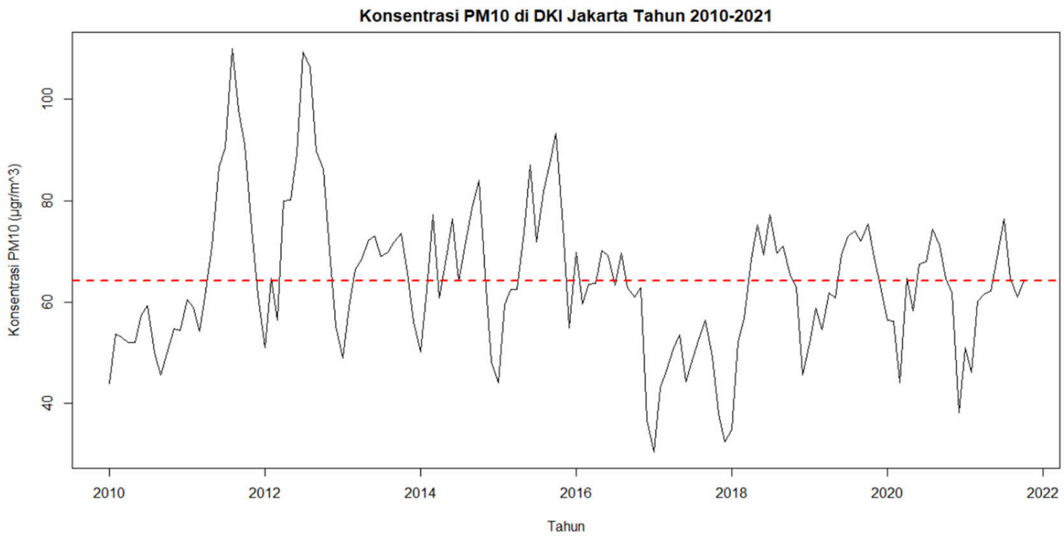
E. Uji Diagnostik: Uji Kolmogorov-Smirnov
Uji Kolmogorov-Smirnov menggunakan perbandingan distribusi data yang akan diuji normalitasnya dengan distribusi normal baku. Hipotesis yang digunakan dalam uji ini adalah:
H_0: residual berdistribusi normal
H_1: residual tidak berdistribusi normal

F. Uji Heteroskedastik
Uji heterokedastisitas bertujuan menguji apakah dalam model regresi terjadi ketidaksamaan varians dari residual satu pengamatan ke pengamatan yang lain. Hipotesis yang digunakan dalam uji heteroskedastik adalah:

H_0: tidak ada efek ARCH
H_1: terdapat efek ARCh

HASIL DAN PEMBAHASAN

Berdasarkan Gambar 2 dapat dilihat bahwa konsentrasi PM10 di DKI Jakarta dari Januari 2010 hingga Oktober 2021 mengalami fluktuasi. Dari statistika deskriptif terlihat bahwa rata-rata konsentrasi PM10 adalah $64.17\, \mu gr/m^3$ yang tergolong tingkat kualitas udara menengah (sedang). Di sisi lain, kenyataannya konsentrasi PM10 minimum bernilai $30.48\, \mu gr/m^3$ dan maksimum bernilai $110.06\, \mu gr/m^3$. Dapat dilihat bahwa jangkauan konsentrasi PM10 cukup besar.

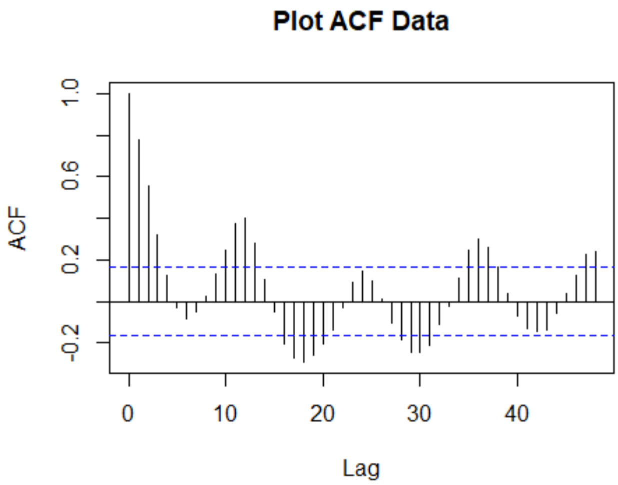


GAMBAR 2. *Line Chart* Konsentrasi PM10 di DKI Jakarta Tahun 2010-2021

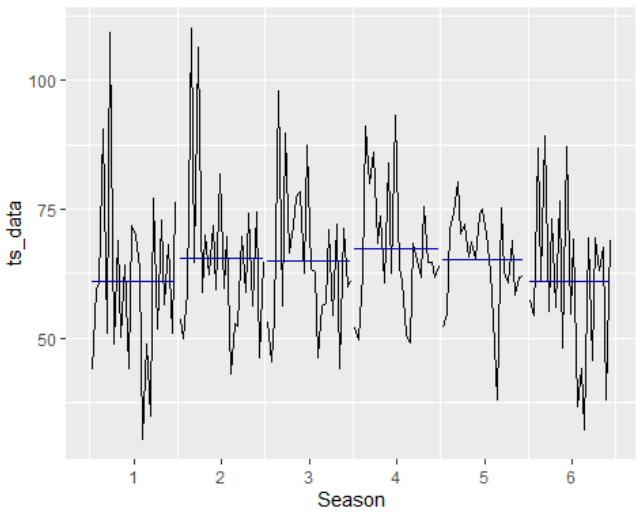
TABEL 3. Sari Numerik Data

Min	Q1	Median	Mean	Q3	Max	Std. Dev.
30.48	54.57	63.06	64.17	71.87	110.06	14.478

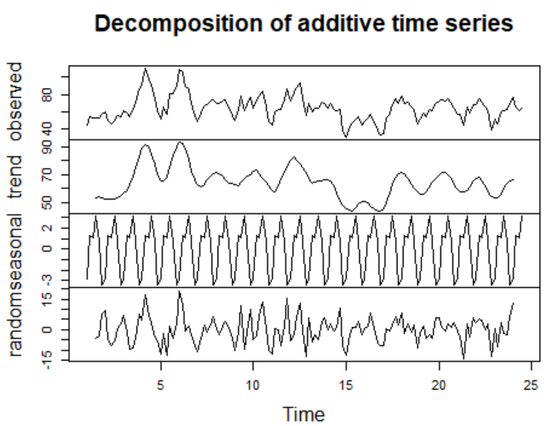
Kemudian, dari plot ACF data pada Gambar 3 terlihat bahwa nilai ACF signifikan pada lag kelipatan enam. Hal tersebut menunjukkan adanya pola musiman dengan periode enam bulan yang bersesuaian dengan musim kemarau dan musim hujan di Indonesia. Pola musiman ini terlihat juga pada subseries plot pada Gambar 4 dan plot dekomposisi pada Gambar 5.



GAMBAR 3. Grafik ACF Data

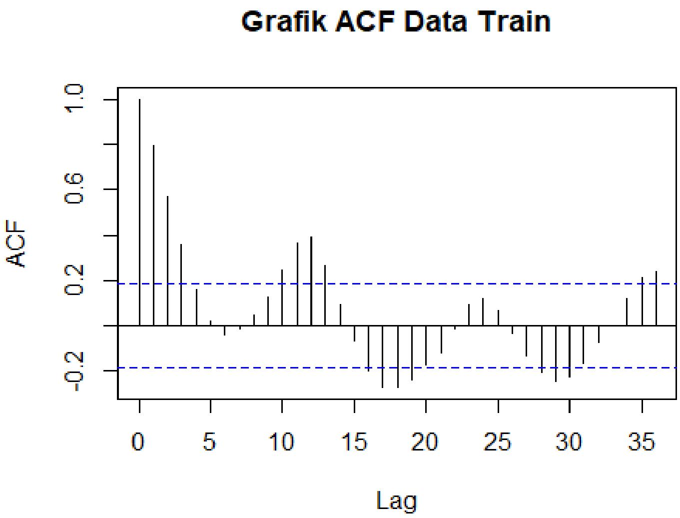


GAMBAR 4. Grafik Data Setiap Musim *(Subseries Plot)*



GAMBAR 5. Dekomposisi Data

Data yang kami gunakan memiliki 142 observasi yang berarti ada sekitar 12 tahun data konsentrasi PM10. Dengan demikian, terdapat sebanyak $0.8 \times 12\ tahun \approx 9.5\ tahun = 114$ observasi dan sisa datanya akan menjadi data validation. Terlihat bahwa ACF data *training* pada Gambar 6 masih memiliki pola musiman sehingga dilakukan diferensiasi musiman. Kemudian, uji ADF pada data diferensiasi musiman tersebut menghasilkan nilai $p - value$ sebesar 0.01 sehingga dapat disimpulkan bahwa data telah stasioner.



GAMBAR 6. Grafik ACF Data *Training*

Dari plot ACF dan PACF diferensiasi musiman kami menduga model yang cocok adalah sebagai berikut.

1. $SARIMA(0,0,2) \times (1,1,0)6$
2. $SARIMA(1,0,0) \times (1,1,0)6$
3. $SARIMA(1,0,2) \times (1,1,0)6$

Kemudian, diperoleh nilai AIC dari setiap model tersebut sebagai berikut.

TABEL 4. AIC Setiap Model

MODEL	AIC
$SARIMA(0,0,2) \times (1,1,0)6$	826.77
$SARIMA(1,0,2) \times (1,1,0)6$	808.01
$SARIMA(1,0,0) \times (1,1,0)6$	804.47
$ARIMA(2,0,0)$ (Cara Otomatis)	836.56

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(1,0,0) \times (1,1,0)_6$ sehingga akan dipilih model $SARIMA(1,0,0) \times (1,1,0)_6$ untuk dianalisis lebih lanjut. Berikutnya, akan dilakukan uji signifikansi dari model.

```
z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.793308   0.058509  13.559 < 2.2e-16 ***
sar1 -0.739606   0.059864 -12.355 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

GAMBAR 7. Hasil Uji Signifikansi Parameter

Dapat dilihat bahwa semua parameter signifikan sehingga dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memodelkan data. Berikutnya, akan dilihat performa model.

```
                    ME     RMSE     MAE        MPE     MAPE      MASE         ACF1
Training set 0.2283871 9.259311 7.06156 -0.8839814 11.46901 0.8899227 -0.01361112
```

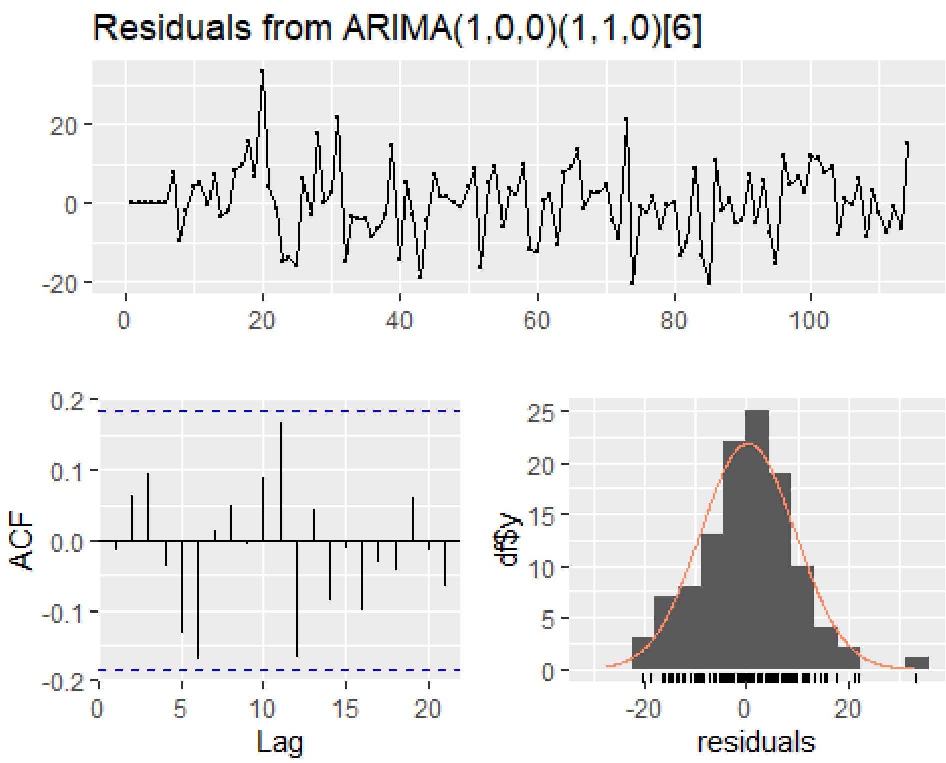
GAMBAR 8. Nilai Galat dari Model

Dapat dilihat bahwa RMSE dari model sebesar 9.259 yang sangat kecil dibandingkan dengan standar deviasi data. Diperoleh juga nilai MAPE sebesar 11.47%. Jadi, dapat disimpulkan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

Selanjutnya, dilakukan uji diagnostik pada model. Diperoleh bahwa uji Ljung-Box menghasilkan $p-value$ sebesar 0.3664 yang lebih besar dari alpha untuk $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data saling bebas. Hal ini juga diperkuat dari plot ACF yang tidak signifikan. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan plot residual terlihat memiliki rataan nol dan variansi yang konstan.

Kemudian, kami melakukan uji Kolmogorov-Smirnov dengan hipotesis
H_0: data berdistribusi normal
H_1: data tidak berdistribusi normal
Diperoleh $p-value = 0.9535 > \alpha = 0.05$ maka H_0 tidak ditolak dan dapat disimpulkan bahwa residual model berdistribusi normal.

Selanjutnya, kami juga melakukan uji ARCH-LM dan memperoleh $p-value = 0.3587 > \alpha = 0.05$ sehingga dapat disimpulkan bahwa data tidak perlu dimodelkan dengan model heteroskedastik. Hal ini didukung pula oleh variansi galat yang terlihat konstan pada plot galat.



GAMBAR 9. Plot Residual

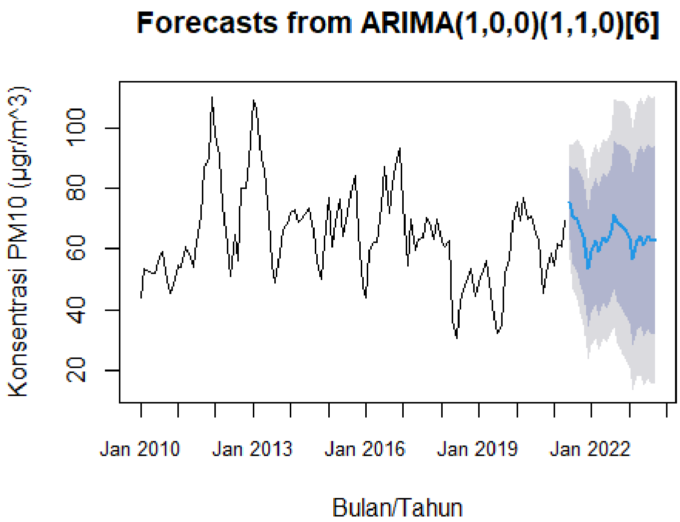
Dengan demikian, dipilih model $SARIMA(1,0,0) \times (1,1,0)6$ dengan persamaan

$$Y_t = 0.799Y_{t-1} + 0.261Y_{t-6} + 0.532Y_{t-7} - 0.586Y_{t-13} + e_t$$

yang menunjukkan bahwa konsentrasi PM10 bulan ini dipengaruhi nilai konsentrasi PM10 satu, enam, tujuh, dan tiga belas bulan sebelumnya.

Sebelum dilakukan prakiraan, kami menguji performa model untuk memodelkan data validasi dan didapat nilai RMSE sebesar 8.7389 yang lebih kecil dari RMSE yang diperoleh sebelumnya, yaitu 9.2593. Kemudian, diperoleh juga MAPE sebesar 11.578% yang tidak berbeda jauh dari MAPE yang diperoleh sebelumnya, yaitu 11.469%.

Berikutnya, dilakukan prakiraan untuk 4 musim ke depan (atau 2 tahun ke depan) dengan model $SARIMA(1,0,0) \times (1,1,0)6$ dan diperoleh grafik prakiraan sebagai berikut.

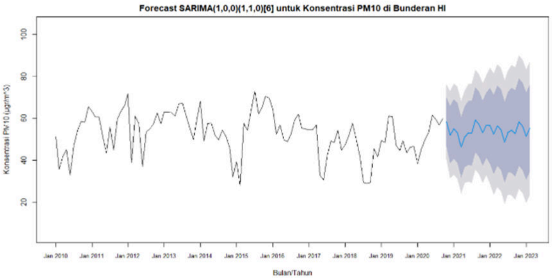


GAMBAR 10. Grafik *Forecasting* Konsentrasi PM10 di DKI Jakarta

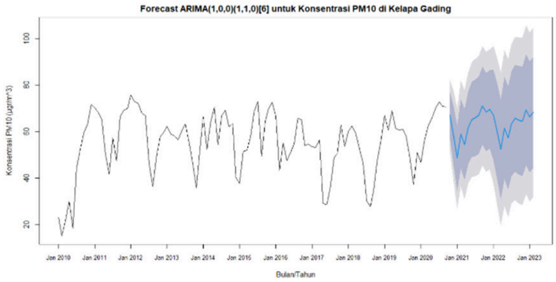
Lebih lanjut, dengan menggunakan cara serupa, kami memodelkan data untuk setiap stasiun tempat partikulat udara diukur dan memperoleh model yang sesuai untuk setiap stasiun sebagai berikut.

STASIUN DKI1: Bundaran HI (Jakarta Pusat) - $SARIMA(1,0,0) \times (1,1,0)_6$
$PM10_{DKI1,t} = 0.586PM10_{DKI1,t-1} - 0.363PM10_{DKI1,t-6} + 0.425PM10_{DKI1,t-7} - 0.374PM10_{DKI1,t-13} + e_t$
STASIUN DKI2: Kelapa Gading (Jakarta Utara) - $SARIMA(1,0,0) \times (1,1,0)_6$
$PM10_{DKI2,t} = 0.780PM10_{DKI2,t-1} - 0.191PM10_{DKI2,t-6} + 0.660PM10_{DKI2,t-7} - 0.631PM10_{DKI2,t-13} + e_t$
STASIUN DKI3: Jagakarsa (Jakarta Utara) - $SARIMA(1,0,0) \times (0,1,1)_6$
$PM10_{DKI3,t} = 0.727PM10_{DKI3,t-1} + PM10_{DKI3,t-6} - 0.727PM10_{DKI3,t-7} + e_t + 0.729e_{t-6}$
STASIUN DKI4: Lubang Buaya (Jakarta Timur) - $SARIMA(1,0,0) \times (1,1,0)_6$
$PM10_{DKI4,t} = 0.689PM10_{DKI4,t-1} - 0.206PM10_{DKI4,t-6} + 0.652PM10_{DKI4,t-7} - 0.547PM10_{DKI4,t-13} + e_t$

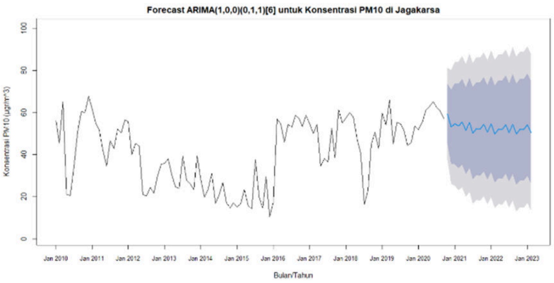
Perhatikan bahwa model DKI1, DKI2, dan DKI4 dipengaruhi oleh konsentrasi PM10 1 musim (6 bulan) sebelumnya. Sementara itu, model DKI3 dipengaruhi oleh galat 1 musim (6 bulan) sebelumnya. Kemudian, dengan melakukan prakiraan untuk 4 musim ke depan dengan hasil sebagai berikut.



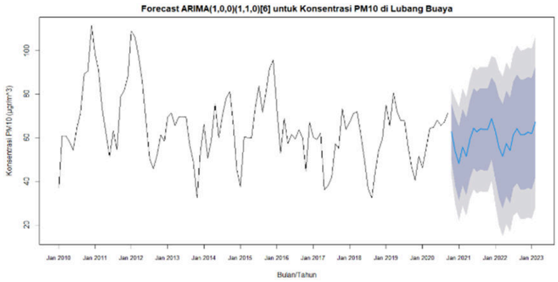
DKI 1



DKI 2

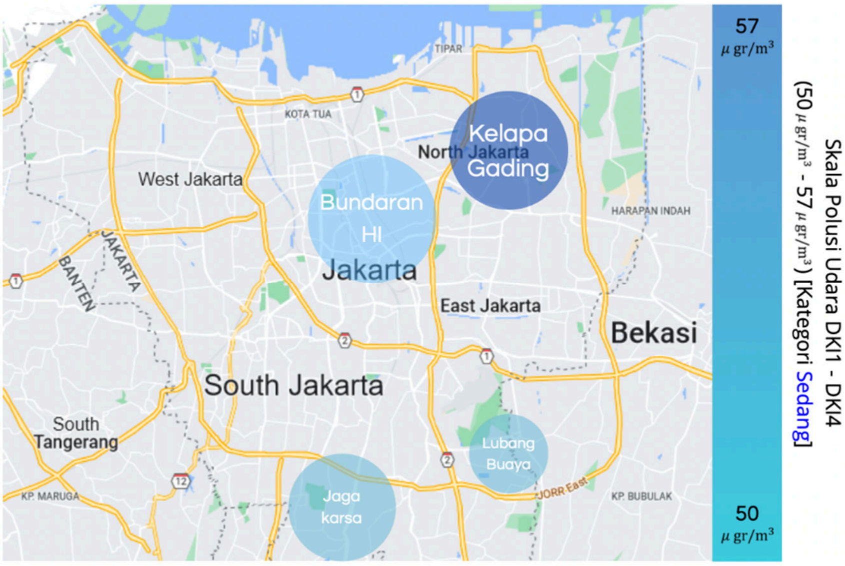


DKI 3



DKI 4

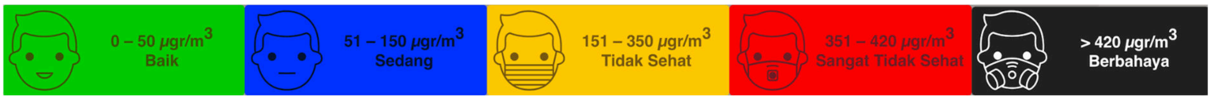
GAMBAR 11. Grafik *Forecasting* Konsentrasi PM10 di masing-masing Stasiun tempat pengukuran PM10



GAMBAR 12. Peta Polusi DKI Jakarta

KESIMPULAN DAN SARAN

Dari hasil penelitian, diperoleh model yang cocok untuk memodelkan konsentrasi partikulat PM10 di DKI Jakarta adalah model $SARIMA(1,0,0) \times (1,1,0)_6$ dengan hasil forecast pemodelan memiliki rata-rata sekitar 50-57$\mu gr/m^3$. Menurut klasifikasi pada bmkg.go.id, angka tersebut termasuk dalam klasifikasi sedang yang berarti tidak terlalu perlu dikhawatirkan namun kualitas udara tetap perlu ditingkatkan agar dalam jangka panjang tidak akan berkembang menjadi sebuah masalah yang serius.



GAMBAR 13. Peta Polusi di Setiap Stasiun DKI Jakarta yang Dipilih

UCAPAN TERIMA KASIH

Terima kasih kepada dosen pengampu mata kuliah AK2281 Analisis Deret Waktu, Dr. Utriweni Mukhaiyar, S.Si., M.Si., serta asisten dosen yang membantu penelitian, membantu dalam diskusi, dan memberi masukan yang membangun dan membuat penelitian berjalan lebih baik.

REFERENSI

Cryer, J. D., & Chan, K. S. (2008). Time series analysis: with applications in R (Vol. 2). New York: Springer.

Suryanto, D.A (2012). Analisis Tingkat Polusi Udara Terhadap Pengaruh Pertumbuhan Kendaraan Studi Kasus DKI Jakarta, Ejournal Gunadarma.

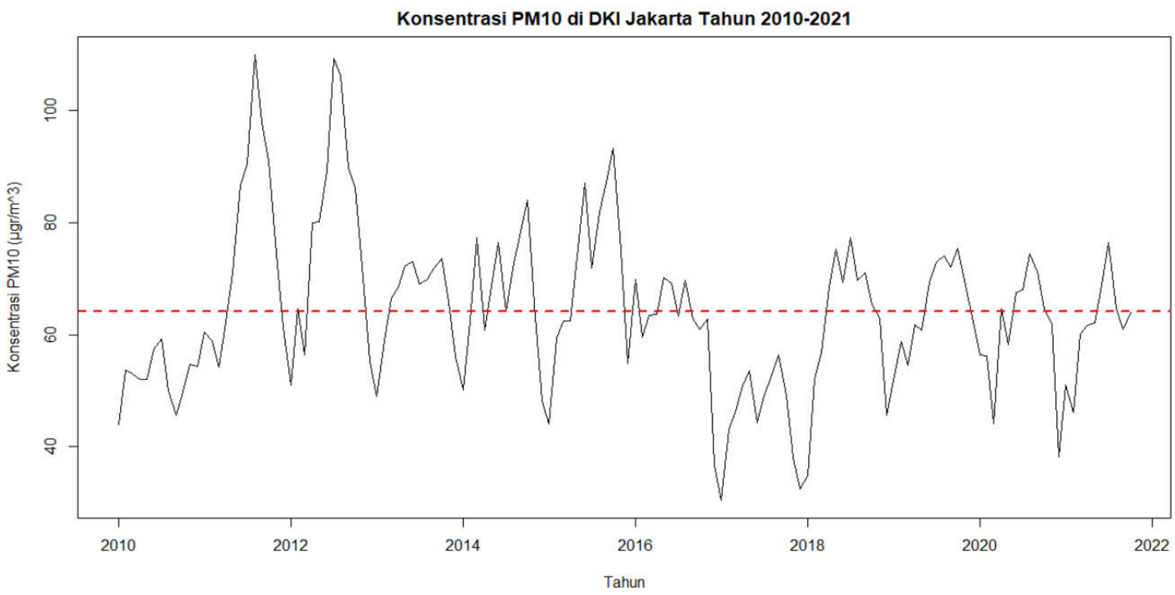
Wei, W. W. (2006). Time series analysis. In The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2.

LAMPIRAN

1) *Syntax* pemodelan PM10 di DKI Jakarta

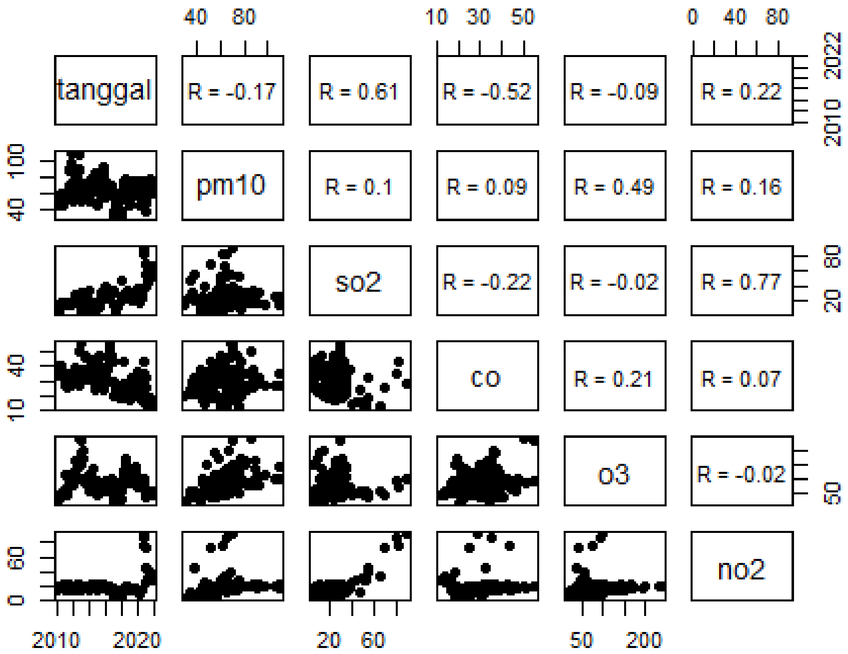
Library yang digunakan

```
# 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)
# Untuk melakukan uji heteroskedastik
library(FinTS)

# Memanggil Data
setwd("E:/Praktikum ADW")
library(readr)
data <- read_csv("ispu.csv", col_types = cols(tanggal = col_date(format = "
%m/%Y")))
# Membuat grafik data
plot(data$tanggal,data$pm10, type = 'l', main = 'Konsentrasi PM10 di DKI Ja
karta Tahun 2010-2021', xlab = "Tahun", ylab = "Konsentrasi PM10 (µgr/m^3)"
)
# Membuat garis rataan
abline(h=mean(data$pm10),lwd=2,
lty = 2, col ='red')
```

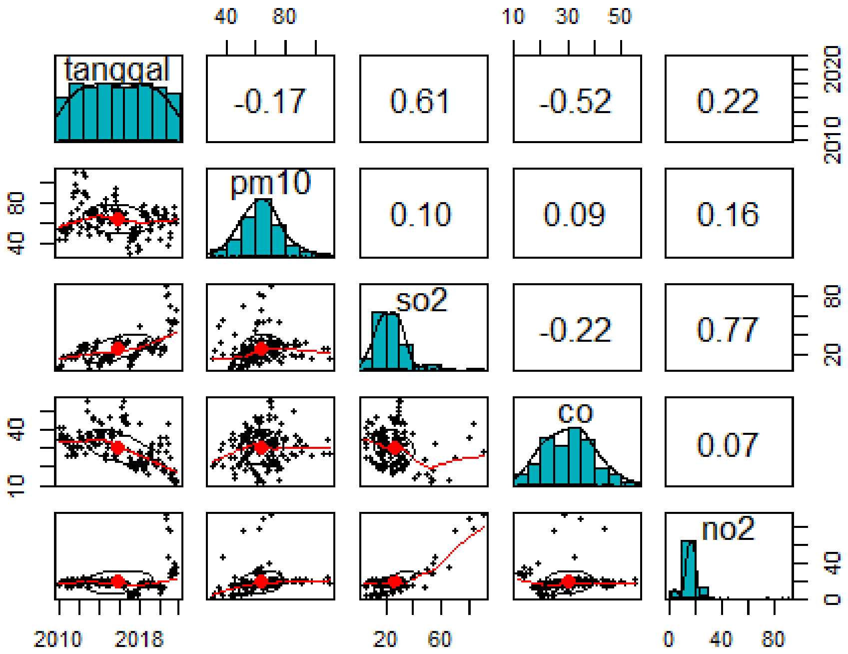



```
# Correlation panel
panel.cor <- function(x, y){
    usr <- par("usr"); on.exit(par(usr))
```

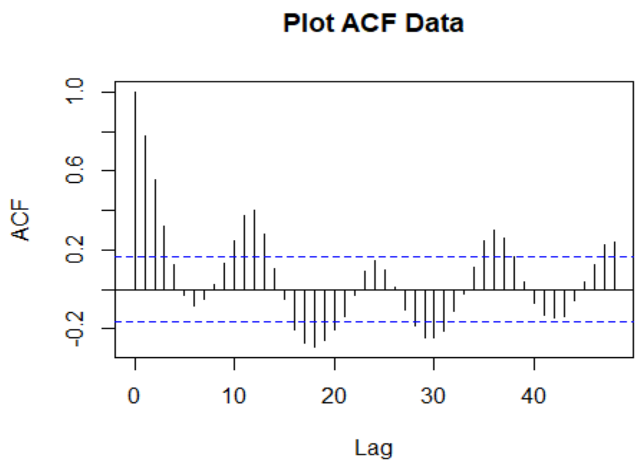
```
    par(usr = c(0, 1, 0, 1))
    r <- round(cor(x, y), digits=2)
    txt <- paste0("R = ", r)
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt)
}
# Customize upper panel
lower.panel<-function(x, y){
  points(x,y, pch = 19)
}
# Create the plots
pairs(data[,1:6],
      lower.panel = lower.panel,
      upper.panel = panel.cor)
```




```
library(psych)
pairs.panels(data[,-5],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



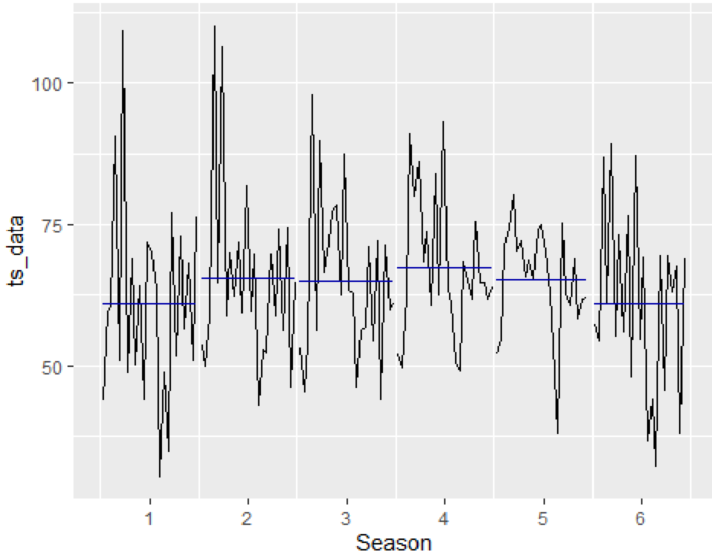

```
summary(data$pm10)

##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    30.48   54.57   63.06   64.17   71.87  110.06

sd(data$pm10)

## [1] 14.47786

acf(data$pm10, lag = 48, main="Plot ACF Data")
```



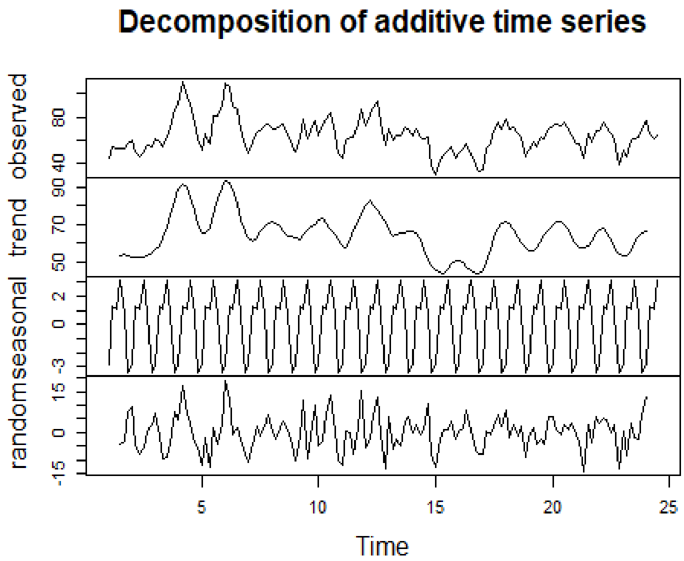
Pada grafik ACF dapat dilihat bahwa nilai ACF setiap lag kelipatan 6 lebih tinggi dari yang lain sehingga akan diasumsikan bahwa data memiliki pola musiman dengan $s = 6$

```
# Membuat data menjadi time series
ts_data <- ts(data$pm10, frequency = 6)
```

```
#Plot per subseries
ggsubseriesplot(ts_data)
```




```
plot(decompose(ts_data))
```

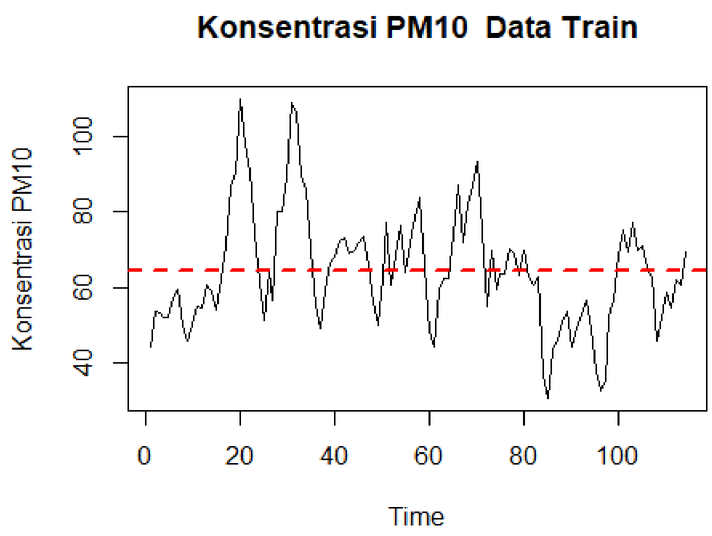


Pada data terdapat 142 observasi yang berarti terdapat sekitar 12 tahun. Dengan demikian, jumlah observasi yang akan menjadi data train adalah sebanyak 0.8 × 12 tahun ≈ 9.5 tahun = 114 observasi dan sisa datanya akan menjadi data validation

```
# Membuat data train dan validation dan mengubahnya menjadi data time serie
s
pm10 <- ts(data$pm10)
# data yang akan digunakan untuk membuat model
pm10_train <- ts(pm10[1:114])
# data yang akan digunakan untuk memvalidasi model
pm10_validation <-ts(pm10[114:142])
```

2. Identifikasi Model

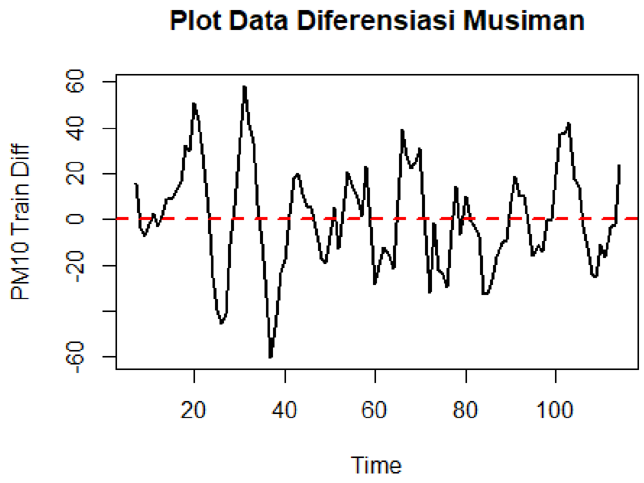
```r
# Membuat grafik data
plot(pm10_train , type = 'l', main = 'Konsentrasi PM10  Data Train', ylab =
"Konsentrasi PM10")
# Membuat garis rataan
abline(h=mean(pm10_train),lwd=2, lty = 2, col ='red')
```



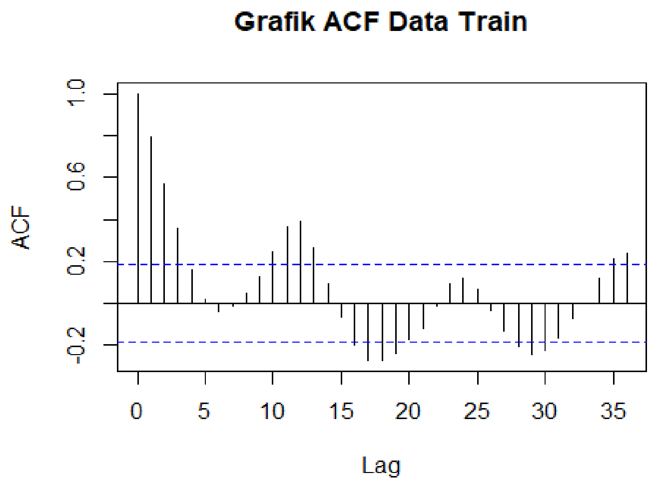

```r
adf.test(pm10_train)

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train
## Dickey-Fuller = -4.3767, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Diperoleh nilai p-value = 0.01 < $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data telah stasioner dalam rataan. sehingga akan dilakukan diferensiasi musiman

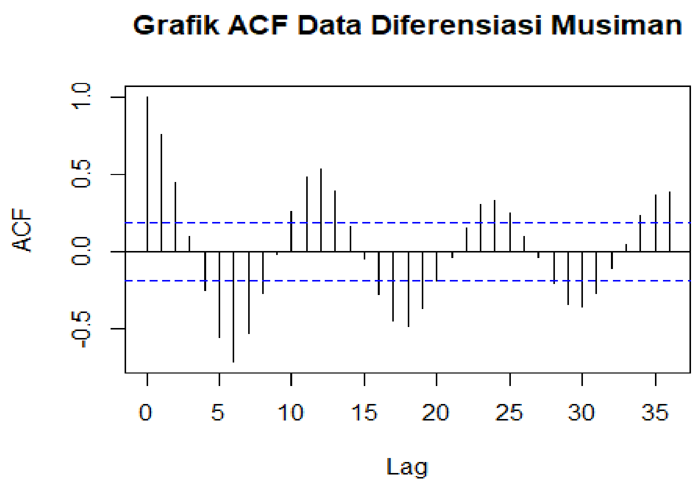
```r
pm10_train_diff <- diff(pm10_train, lag = 6)
plot(pm10_train_diff,lwd = 2,main = 'Plot Data Diferensiasi Musiman', ylab=
"PM10 Train Diff")
abline(h=mean(pm10_train_diff), lwd=2,lty = 2, col ='red')
```



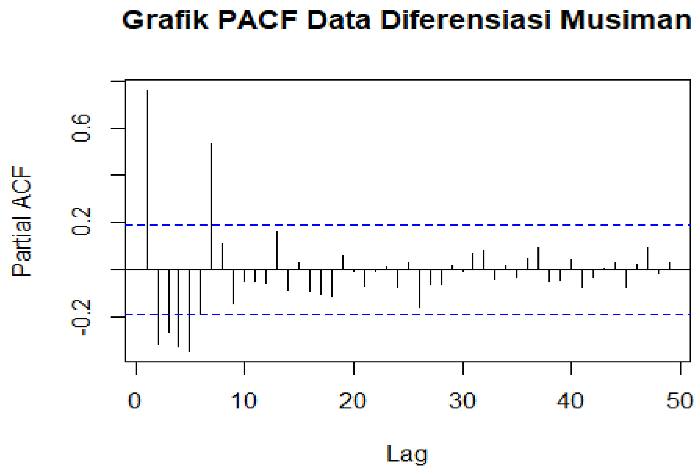

```
acf(pm10_train, main = "Grafik ACF Data Train", lag.max = 36)
```




```
adf.test(pm10_train_diff)

## Warning in adf.test(pm10_train_diff): p-value smaller than printed p-val
ue

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train_diff
## Dickey-Fuller = -8.0834, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

acf(pm10_train_diff, main = "Grafik ACF Data Diferensiasi Musiman", lag.max
= 36)
```




```
pacf(pm10_train_diff, main = "Grafik PACF Data Diferensiasi Musiman", lag.m
ax = 49)
```



Dari grafik-grafik tersebut dapat disimpulkan beberapa model yang akan dicoba, yaitu: 1. $SARIMA(0,0,2) \times (1,1,0)_6$ 2. $SARIMA(1,0,0) \times (1,1,0)_6$ 3. $SARIMA(1,0,2) \times (1,1,0)_6$

3. PENAKSIRAN PARAMETER

```
mod_1 <- arima(pm10_train, order = c(0,0,2), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_1

##
## Call:
## arima(x = pm10_train, order = c(0, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ma1     ma2     sar1
##       0.7516  0.4231  -0.6666
```

```
## s.e.  0.0856  0.0895   0.0690
##
## sigma^2 estimated as 110.2:  log likelihood = -409.39,  aic = 826.77
```

```r
mod_2 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_2
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1     sar1
##       0.7933  -0.7396
## s.e.  0.0585   0.0599
##
## sigma^2 estimated as 90.5:  log likelihood = -399.24,  aic = 804.47
```

```r
mod_3 <- arima(pm10_train, order = c(1,0,2), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_3
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1     ma2     sar1
##       0.7751  -0.0076  0.0750  -0.7384
## s.e.  0.0851   0.1284  0.1093   0.0629
##
## sigma^2 estimated as 90.09:  log likelihood = -399,  aic = 808.01
```

```r
mod_auto = auto.arima(pm10_train, max.p=4,max.q=4, max.P = 4, max.Q = 4,sea
sonal =TRUE, stationary = FALSE)
mod_auto
```

```
## Series: pm10_train
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2     mean
##       0.9414  -0.1783  64.0954
## s.e.  0.0924   0.0925   3.5329
##
## sigma^2 estimated as 85.4:  log likelihood=-414.28
## AIC=836.56   AICc=836.93   BIC=847.5
```

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(1,0,0) \times (1,1,0)_6$ sehingga akan dipilih model $SARIMA(1,0,0) \times (1,1,0)_6$ untuk dianalisis lebih lanjut. Berikutnya, akan dilakukan uji signifikansi dari model.

4. Signifikansi dari Koefisien Parameter dan Pembuatan Model

```
coeftest(mod_2)

##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.793308   0.058509  13.559 < 2.2e-16 ***
## sar1 -0.739606   0.059864 -12.355 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

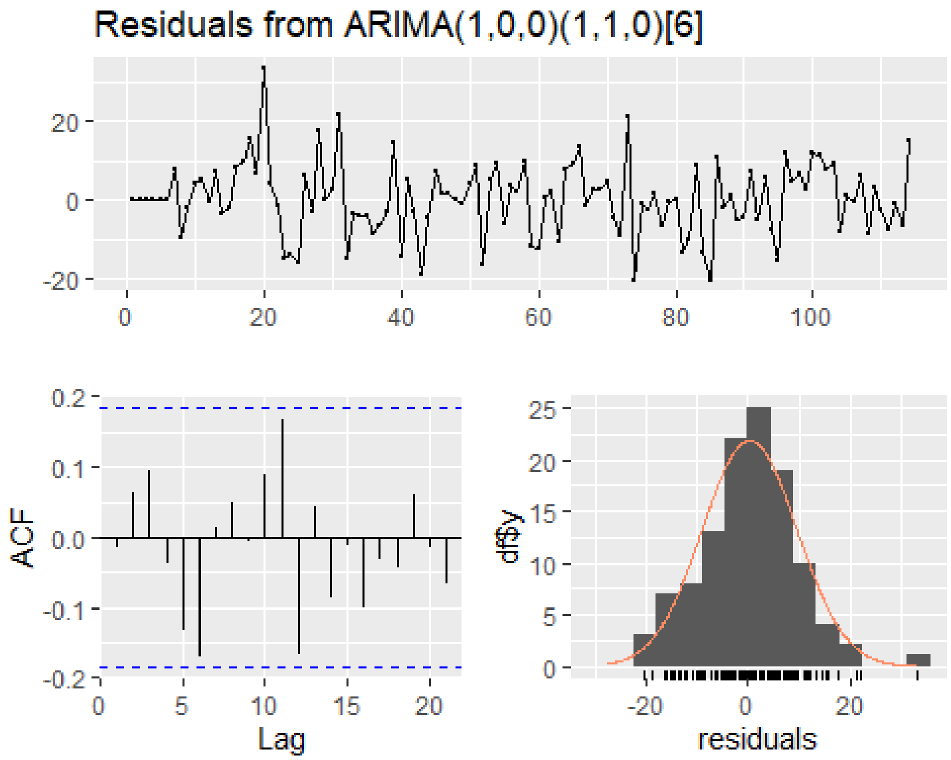
Dapat dilihat bahwa semua parameter signifikan sehingga dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memodelkan data. Berikutnya, akan dilihat performa model.

```
accuracy(mod_2)

##                     ME     RMSE     MAE        MPE     MAPE      MASE
## Training set 0.2283871 9.259311 7.06156 -0.8839814 11.46901 0.8899227
##                     ACF1
## Training set -0.01361112
```

Dapat dilihat bahwa RMSE dari model sebesar 9.259 yang sangat kecil dibandingkan dengan standar deviasi data. Diperoleh juga nilai MAPE sebesar 11.47%. Jadi, dapat disimpulkan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

5. Uji Diagnostik

```
checkresiduals(mod_2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[6]
## Q* = 8.7204, df = 8, p-value = 0.3664
##
## Model df: 2.   Total lags used: 10
```

Perhatikan bahwa p-value > $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data saling bebas. Hal ini juga diperkuat dari plot ACF yang tidak signifikan. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan plot residual terlihat memiliki rataan nol dan variansi yang konstan.

```
data_residuals <- residuals(mod_2)
ks.test(residuals(mod_2), "pnorm", mean=mean(data_residuals), sd=sd(data_re
siduals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  residuals(mod_2)
## D = 0.048233, p-value = 0.9535
## alternative hypothesis: two-sided
```

Uji Kolmogorov-Smirnov dengan hipotesis:
H_0: data berdistribusi normal
H_1: data tidak berdistribusi normal
Diperoleh p-value > $\alpha = 0.05$ maka H_0 tidak ditolak dan dapat disimpulkan bahwa residual model berdistribusi normal.

```
ArchTest(data_residuals)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  data_residuals
## Chi-squared = 13.144, df = 12, p-value = 0.3587
```

Karena p-value = 0.3587 > $\alpha = 0.05$, dapat disimpulkan bahwa data tidak perlu dimodelkan dengan model heteroskedastik. Hal ini didukung pula oleh variansi galat yang terlihat konstan pada plot galat.

5.1 Forecasting

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

```
validation <- forecast(pm10_train,model=mod_2,h=length(pm10_validation))
actual <- as.vector(pm10_validation)
rmse_validation <- sqrt(mean((as.vector(validation$mean)-actual)^2))
rmse_validation

## [1] 8.738945
```

Dapat dilihat bahwa RMSE dari model ≈ 8.7389 yang lebih kecil daripada RMSE yang diperoleh sebelumnya. Dengan demikian, dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memprediksi data.

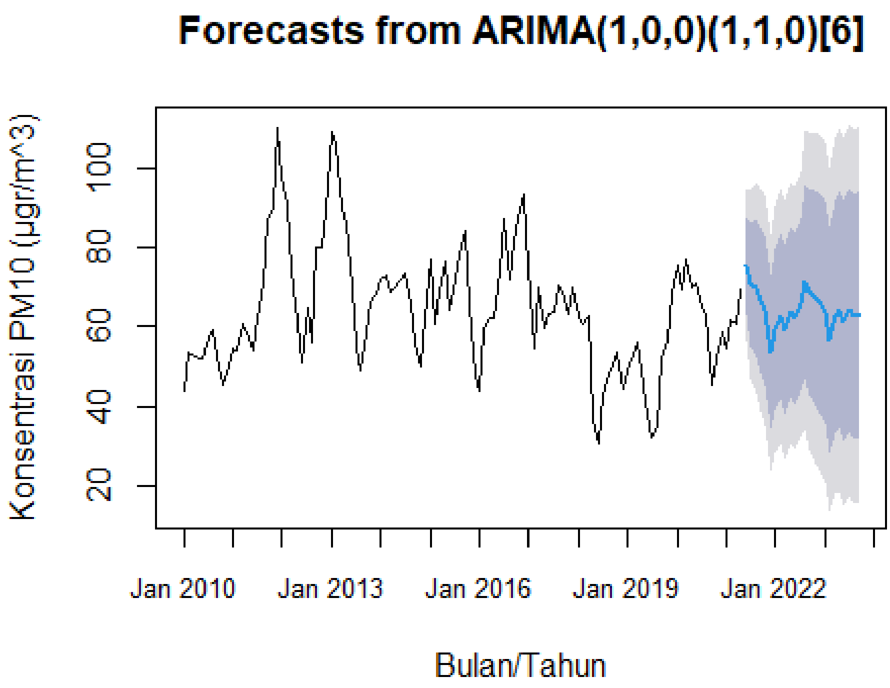
```
ape_validation <- abs((as.vector(validation$mean) -actual)/actual)*100
mape_validation <- mean(ape_validation)
mape_validation

## [1] 11.57768
```

Diperoleh juga MAPE sebesar ≈ 11.578 yang hampir sama dengan nilai MAPE yang diperoleh sebelumnya

```
fc <- forecast(pm10,model=mod_2, h=24)
summary(fc)

##
## Forecast method: ARIMA(1,0,0)(1,1,0)[6]
##
## Model Information:
## Series: object
## ARIMA(1,0,0)(1,1,0)[6]
##
## Coefficients:
##          ar1     sar1
##       0.7933  -0.7396
## s.e.  0.0000   0.0000
##
## sigma^2 estimated as 90.5:  log likelihood=-500.75
## AIC=1003.5   AICc=1003.53   BIC=1006.41
##
## Error measures:
##                       ME     RMSE       MAE        MPE       MAPE       MASE
```

```
## Training set 0.1285096 9.221845 7.122363 -1.070013 11.76014 0.9265439
##                       ACF1
## Training set -0.09898477
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## 143       61.89696 49.70552 74.08840 43.25177  80.54216
## 144       46.17466 30.61284 61.73649 22.37490  69.97442
## 145       57.65201 40.30156 75.00246 31.11679  84.18724
## 146       51.03059 32.64347 69.41770 22.90993  79.15125
## 147       60.34434 41.33378 79.35490 31.27020  89.41848
## 148       62.30707 42.91443 81.69972 32.64858  91.96557
## 149       62.10533 41.74161 82.46906 30.96171  93.24896
## 150       62.96843 42.01663 83.92022 30.92542  95.01143
## 151       71.54042 50.22684 92.85400 38.94412 104.13672
## 152       61.15449 39.61635 82.69263 28.21475  94.09424
## 153       60.81005 39.13177 82.48832 27.65598  93.96411
## 154       63.53962 41.77361 85.30562 30.25138  96.82785
## 155       61.95448 37.38738 86.52158 24.38235  99.52662
## 156       50.55025 24.37354 76.72696 10.51643  90.58407
## 157       61.27053 34.12973 88.41133 19.76226 102.77880
## 158       53.66842 25.93807 81.39878 11.25851  96.07834
## 159       60.46690 32.37186 88.56194 17.49924 103.43456
## 160       62.62905 34.30690 90.95120 19.31406 105.94403
## 161       62.06687 32.70331 91.43042 17.15919 106.97455
## 162       59.73545 29.73503 89.73587 13.85376 105.61713
## 163       68.86671 38.47232 99.26110 22.38251 115.35091
## 164       59.20557 28.56584 89.84529 12.34615 106.06498
## 165       60.72102 29.92789 91.51414 13.62700 107.81504
## 166       63.30276 32.41349 94.19204 16.06170 110.54383
```

```
plot(forecast(mod_2, h=24, level=c(80,95)), xaxt='n', ylab="Konsentrasi PM1
0 (µgr/m^3)", xlab="Bulan/Tahun")
axis(1, at=seq(1, 150,10), labels=format(seq(as.Date("2010/1/1"), by = "yea
r", length.out = 15),format("%b %Y")), cex.axis=0.8, xpd=TRUE)
```

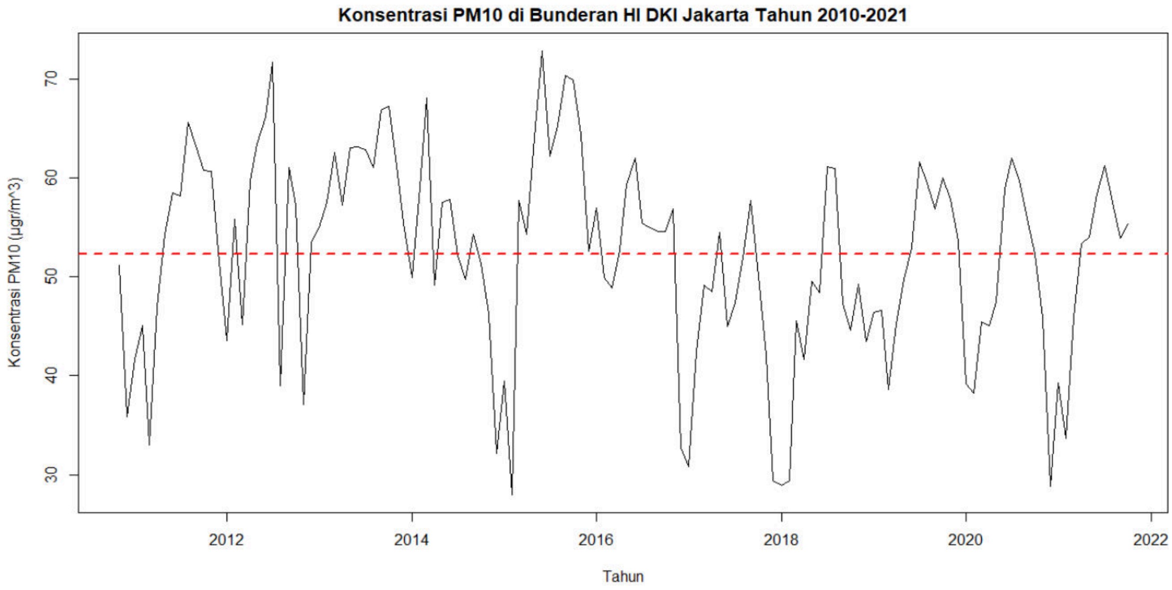


2) *Syntax* pemodelan PM10 di stasiun DKI 1

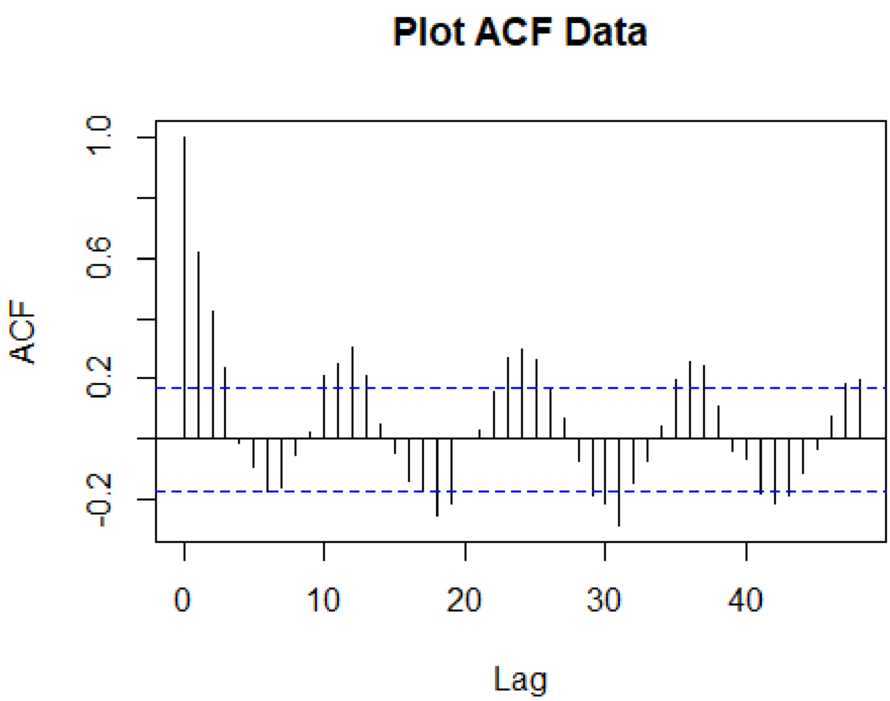
Library yang digunakan

```
# 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)
# Untuk melakukan uji heteroskedastik
library(FinTS)

# Memanggil Data
setwd("E:/Praktikum ADW")
library(readr)
data <- read_csv("ispu_stasiun.csv",
    col_types = cols(`bulan/tahun` = col_date(format = "%m/%Y"),
        `DKI 1` = col_number(), `DKI 2` = col_number(),
        `DKI 3` = col_number(), `DKI 4` = col_number()))
# Membuat grafik data
plot(data$`bulan/tahun`,data$`DKI 1`, type = 'l', main = 'Konsentrasi PM10
di Bunderan HI DKI Jakarta Tahun 2010-2021', xlab = "Tahun", ylab = "Konsen
trasi PM10 (µgr/m^3)")
# Membuat garis rataan
```

```
abline(h=mean(data$`DKI 1`),lwd=2,
lty = 2, col ='red')
```

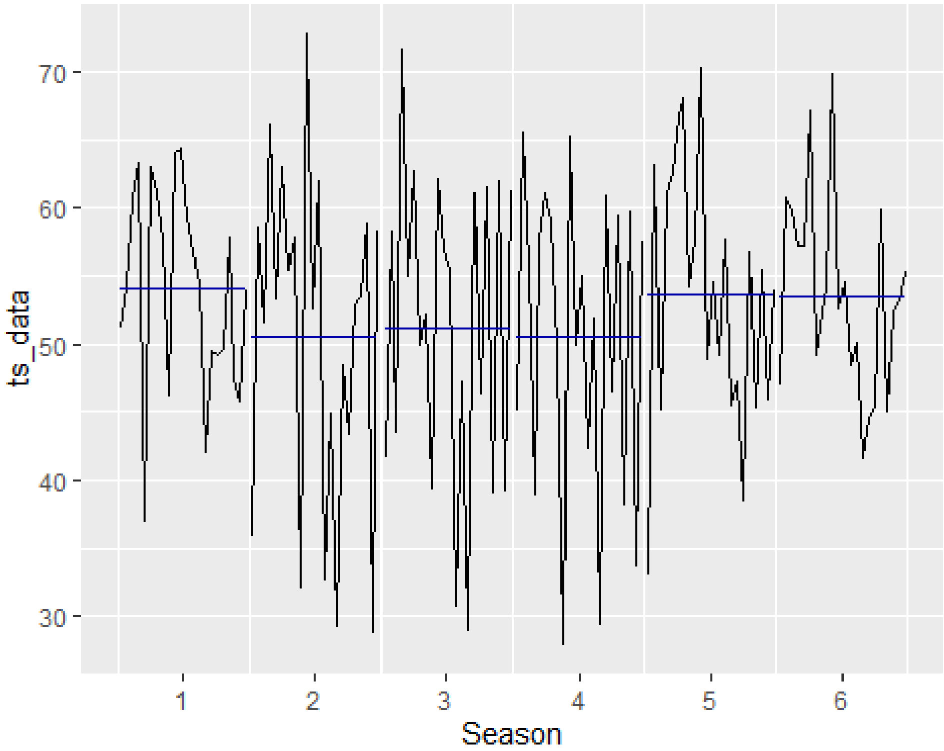



```
summary(data$`DKI 1`)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   28.00   45.89   53.98   52.27   59.50   72.85

sd(data$`DKI 1`)

## [1] 10.02104

acf(data$`DKI 1`, lag = 48, main="Plot ACF Data")
```

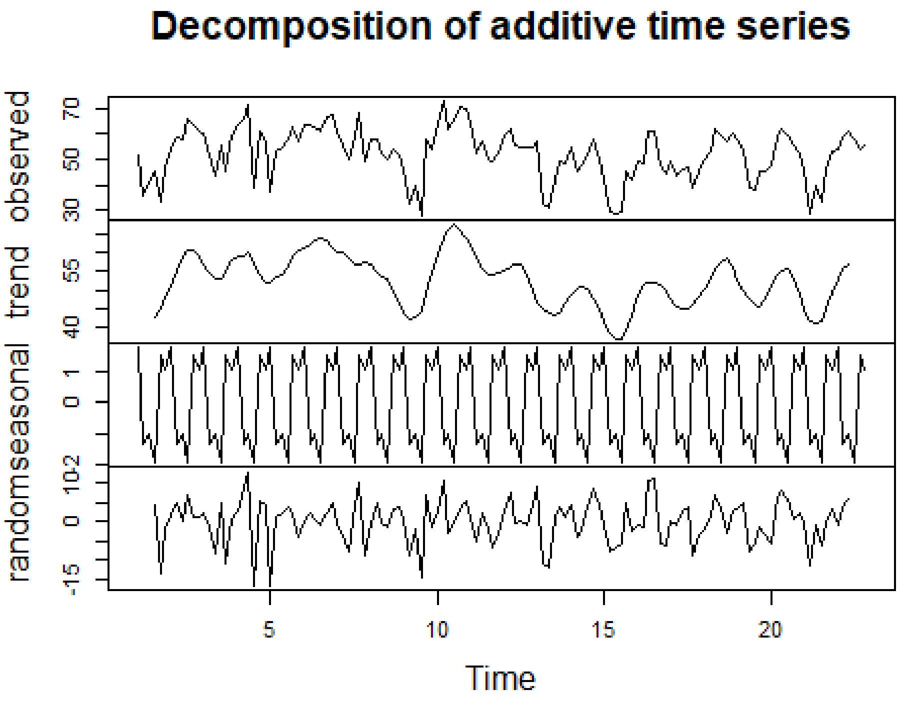


Pada grafik ACF dapat dilihat bahwa nilai ACF setiap lag kelipatan 6 lebih tinggi dari yang lain sehingga akan diasumsikan bahwa data memiliki pola musiman dengan $s = 6$

```
# Membuat data menjadi time series
ts_data <- ts(data$`DKI 1`, frequency = 6)
#Plot per subseries
ggsubseriesplot(ts_data)
```




```
plot(decompose(ts_data))
```

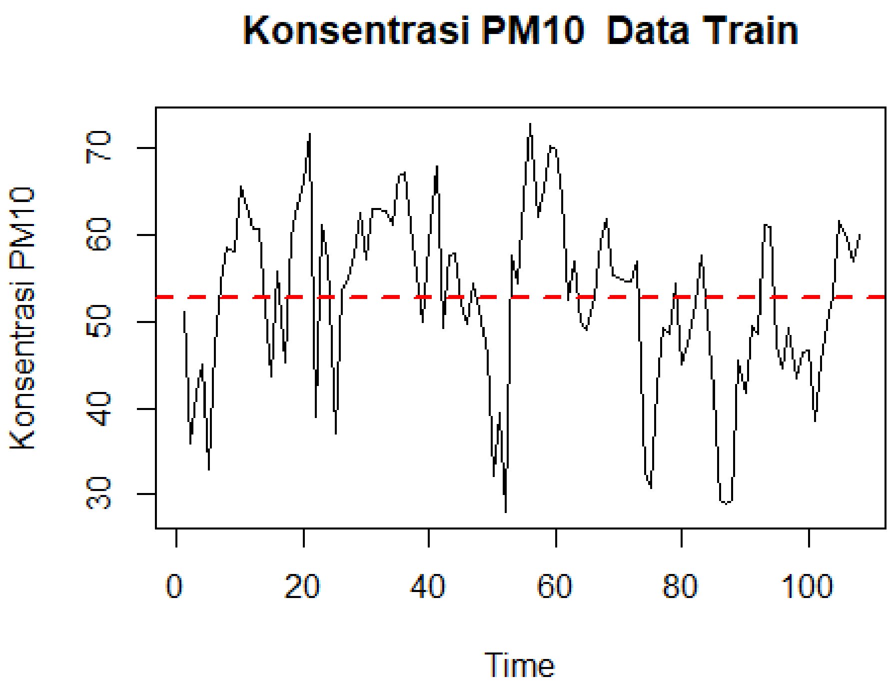


Pada data terdapat 132 observasi yang berarti terdapat 11 tahun. Dengan demikian, jumlah observasi yang akan menjadi data train adalah sebanyak 0.8 × 11 tahun ≈ 9 tahun = 108 observasi dan sisa datanya akan menjadi data validation

```
# Membuat data train dan validation dan mengubahnya menjadi data time serie
s
pm10 <- ts(data$`DKI 1`)
# data yang akan digunakan untuk membuat model
pm10_train <- ts(pm10[1:108])
# data yang akan digunakan untuk memvalidasi model
pm10_validation <-ts(pm10[108:132])
```

2. Identifikasi Model

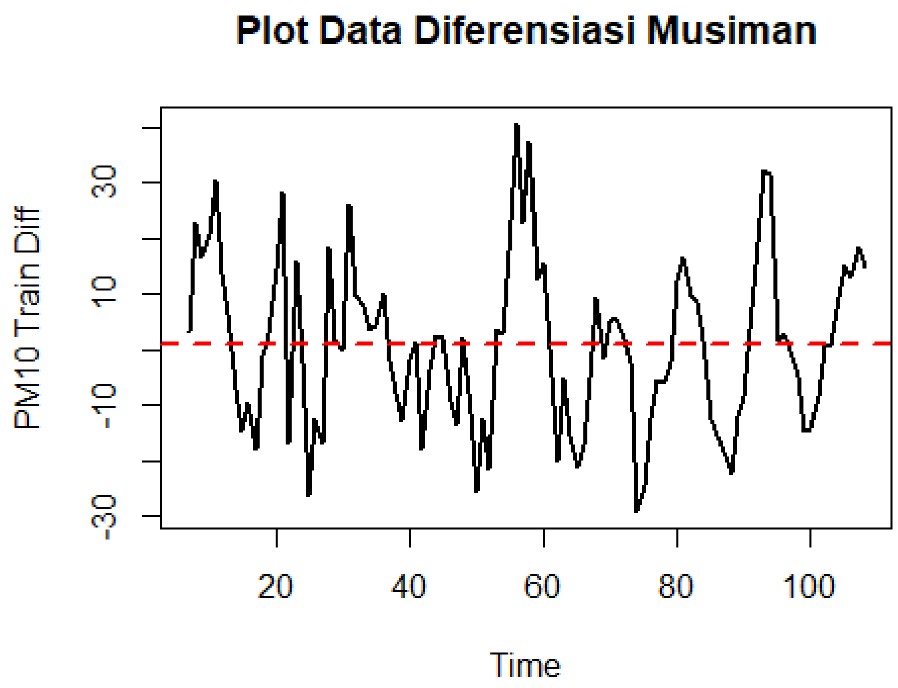
```
# Membuat grafik data
plot(pm10_train , type = 'l', main = 'Konsentrasi PM10  Data Train', ylab =
"Konsentrasi PM10")
# Membuat garis rataan
abline(h=mean(pm10_train),lwd=2, lty = 2, col ='red')
```



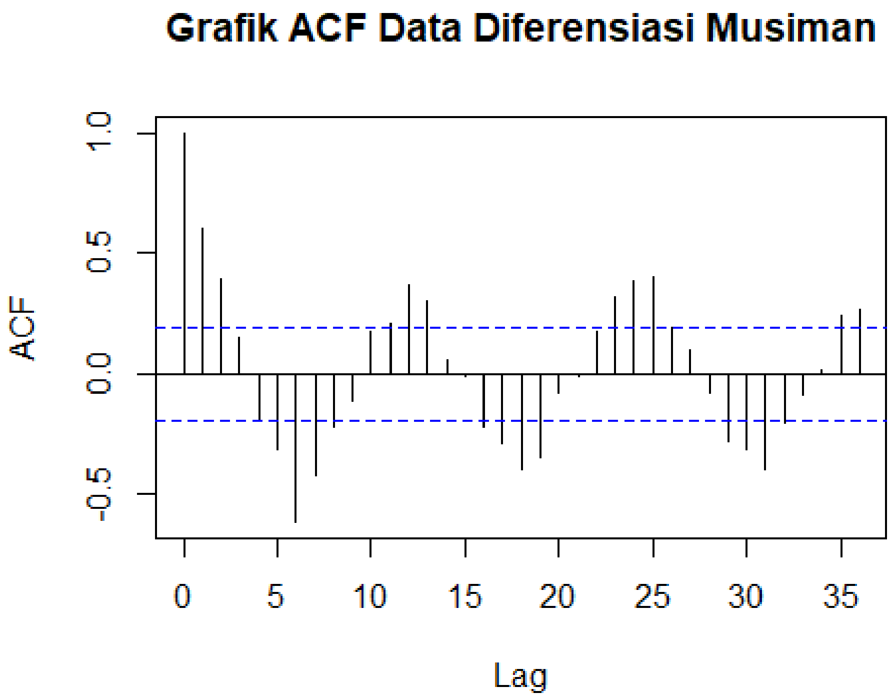

```
adf.test(pm10_train)

## Warning in adf.test(pm10_train): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train
## Dickey-Fuller = -4.6338, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Diperoleh nilai p-value = 0.01 < $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data telah stasioner dalam rataan. sehingga akan dilakukan diferensiasi musiman

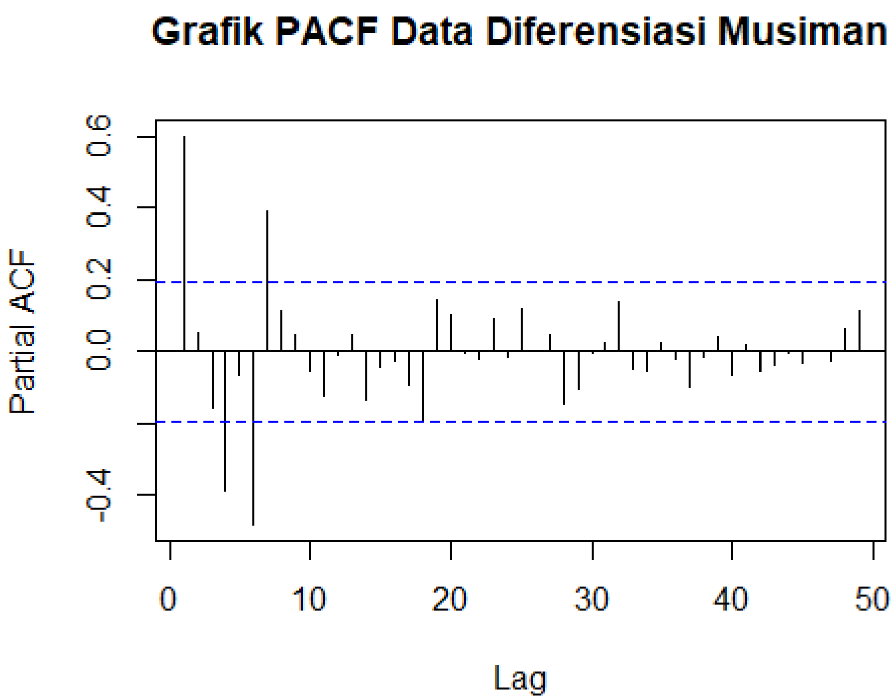
```
pm10_train_diff <- diff(pm10_train, lag = 6)
plot(pm10_train_diff,lwd = 2,main = 'Plot Data Diferensiasi Musiman', ylab=
"PM10 Train Diff")
abline(h=mean(pm10_train_diff), lwd=2,lty = 2, col ='red')
```




```
adf.test(pm10_train_diff)

## Warning in adf.test(pm10_train_diff): p-value smaller than printed p-val
ue

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train_diff
## Dickey-Fuller = -5.7281, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

acf(pm10_train_diff, main = "Grafik ACF Data Diferensiasi Musiman", lag.max
= 36)
```




```
pacf(pm10_train_diff, main = "Grafik PACF Data Diferensiasi Musiman", lag.max = 49)
```



Dari grafik-grafik tersebut dapat disimpulkan beberapa model yang akan dicoba, yaitu: 1. $SARIMA(0,0,2) \times (1,1,0)_6$ 2. $SARIMA(1,0,0) \times (1,1,0)_6$ 3. $SARIMA(1,0,2) \times (1,1,0)_6$

3. PENAKSIRAN PARAMETER

```
mod_1 <- arima(pm10_train, order = c(0,0,2), seasonal = list(order = c(1,1,
0), period = 6),
                method = 'ML')
mod_1

##
## Call:
## arima(x = pm10_train, order = c(0, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ma1     ma2      sar1
##       0.4412  0.2355   -0.6283
## s.e.  0.1104  0.1194    0.0778
##
## sigma^2 estimated as 90.5:  log likelihood = -376.15,  aic = 760.3

mod_2 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(1,1,
0), period = 6),
                method = 'ML')
mod_2

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1     sar1
##       0.5864  -0.6368
## s.e.  0.0811   0.0740
##
## sigma^2 estimated as 83.42:  log likelihood = -372.09,  aic = 750.18

mod_3 <- arima(pm10_train, order = c(1,0,2), seasonal = list(order = c(1,1,
0), period = 6),
                method = 'ML')
mod_3

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1     ma2     sar1
##       0.6975  -0.2282  0.1056  -0.6450
## s.e.  0.1281   0.1474  0.1413   0.0759
##
## sigma^2 estimated as 81.1:  log likelihood = -370.71,  aic = 751.42
```

```
mod_auto = auto.arima(pm10_train, max.p=4,max.q=4, max.P = 4, max.Q = 4,sea
sonal =TRUE, stationary = FALSE)
mod_auto
```

```
## Series: pm10_train
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1     mean
##       0.5998  52.8261
## s.e.  0.0761   1.9168
##
## sigma^2 estimated as 66.54:  log likelihood=-379.14
## AIC=764.28   AICc=764.51   BIC=772.33
```

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(1,0,0) \times (1,1,0)_6$ sehingga akan dipilih model $SARIMA(1,0,0) \times (1,1,0)_6$ untuk dianalisis lebih lanjut. Berikutnya, akan dilakukan uji signifikansi dari model.

4. Signifikansi dari Koefisien Parameter dan Pembuatan Model

```
coeftest(mod_2)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.586425   0.081094  7.2315 4.778e-13 ***
## sar1 -0.636769   0.073964 -8.6092 < 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. Berikutnya, akan dilihat performa model.

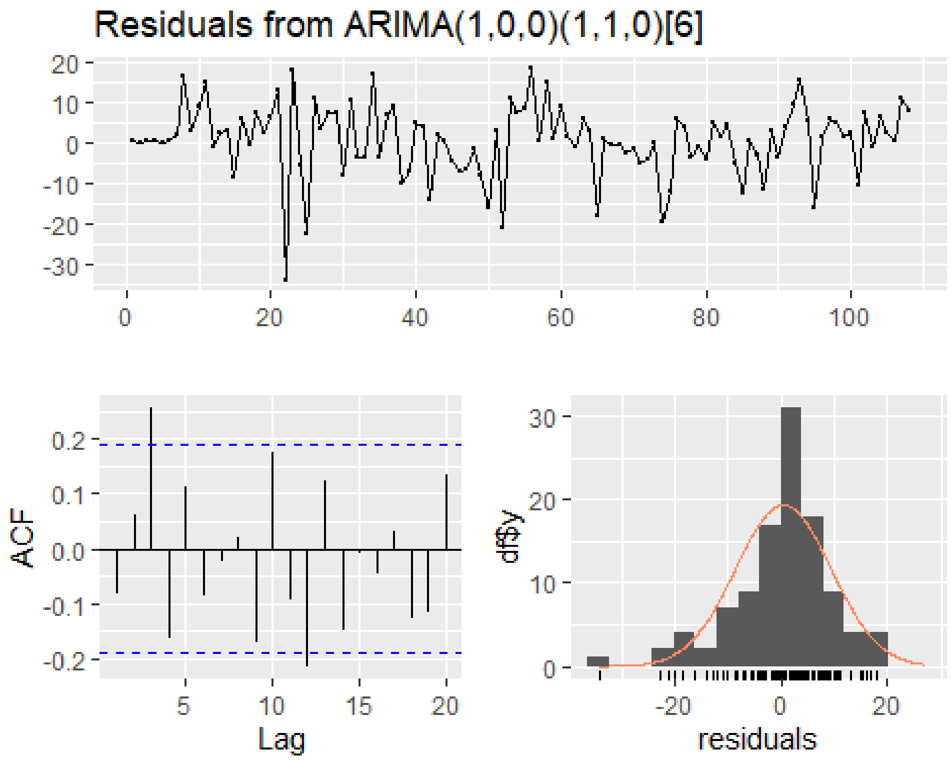
```
accuracy(mod_2)
```

```
##                     ME     RMSE       MAE       MPE     MAPE       MASE
ACF1
## Training set 0.3894066 8.87588 6.503669 -1.572567 13.3971 0.9690168 -0.0
8258947
```

Dapat dilihat bahwa RMSE dari model sebesar 8.876 yang sangat kecil dibandingkan dengan standar deviasi data (= 10.021). Jadi, dapat disimpulkan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

5. Uji Diagnostik

```
checkresiduals(mod_2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[6]
## Q* = 21.208, df = 8, p-value = 0.006616
##
## Model df: 2.   Total lags used: 10
```

Perhatikan bahwa p-value > $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data saling bebas. Hal ini juga diperkuat dari plot ACF yang tidak signifikan. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan plot residual terlihat memiliki rataan nol dan variansi yang konstan.

```
data_residuals <- residuals(mod_2)
ks.test(residuals(mod_2), "pnorm", mean=mean(data_residuals), sd=sd(data_re
siduals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  residuals(mod_2)
## D = 0.11247, p-value = 0.1301
## alternative hypothesis: two-sided
```

Uji Kolmogorov-Smirnov dengan hipotesis:
H_0: data berdistribusi normal
H_1: data tidak berdistribusi normal
Diperoleh p-value = 0.1301 > $\alpha = 0.05$ maka H_0 tidak ditolak dan dapat disimpulkan bahwa residual model berdistribusi normal.

```
ArchTest(data_residuals)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  data_residuals
## Chi-squared = 7.4467, df = 12, p-value = 0.8267
```

Karena p-value = $0.8267 > \alpha = 0.05$, dapat disimpulkan bahwa data tidak perlu dimodelkan dengan model heteroskedastik. Hal ini didukung pula oleh variansi galat yang terlihat konstan pada plot galat.

5.1 Forecasting

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

```
validation <- forecast(pm10_train,model=mod_2,h=length(pm10_validation))
actual <- as.vector(pm10_validation)
rmse_validation <- sqrt(mean((as.vector(validation$mean)-actual)^2))
rmse_validation

## [1] 8.714455
```

Dapat dilihat bahwa RMSE dari model ≈ 8.714 yang lebih kecil daripada RMSE yang diperoleh sebelumnya. Dengan demikian, dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memprediksi data.

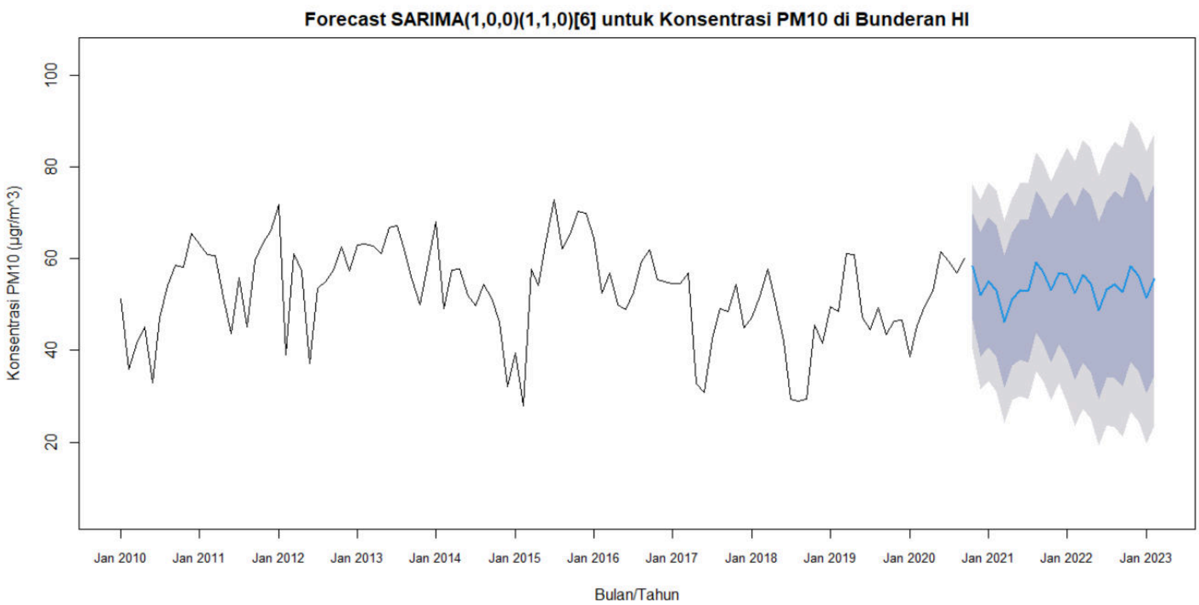
```
ape_validation <- abs((as.vector(validation$mean) -actual)/actual)*100
mape_validation <- mean(ape_validation)
mape_validation

## [1] 15.21531
```

Diperoleh juga MAPE sebesar ≈ 15.215

```
fc <- forecast(pm10,model=mod_2, h=24)
summary(fc)

##
## Forecast method: ARIMA(1,0,0)(1,1,0)[6]
##
## Model Information:
## Series: object
## ARIMA(1,0,0)(1,1,0)[6]
##
## Coefficients:
##          ar1     sar1
##       0.5864  -0.6368
## s.e.  0.0000   0.0000
##
## sigma^2 estimated as 83.42:  log likelihood=-454.99
## AIC=911.99   AICc=912.02   BIC=914.82
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.2179156 8.627932 6.351321 -1.924762 13.39272 0.9791167
```

```
##                        ACF1
## Training set -0.06640536
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 133       50.41776 38.71309 62.12242 32.51702 68.31850
## 134       40.50198 26.93318 54.07078 19.75029 61.25367
## 135       47.80290 33.64965 61.95614 26.15738 69.44842
## 136       42.69088 28.34215 57.03962 20.74639 64.63538
## 137       49.00719 34.59184 63.42254 26.96082 71.05356
## 138       54.15273 39.71455 68.59091 32.07143 76.23403
## 139       52.78366 37.59120 67.97611 29.54880 76.01851
## 140       51.89402 36.45069 67.33736 28.27549 75.51256
## 141       56.37248 40.84381 71.90115 32.62343 80.12154
## 142       52.16444 36.60653 67.72235 28.37067 75.95821
## 143       52.15142 36.58346 67.71937 28.34229 75.96055
## 144       54.91640 39.34500 70.48781 31.10199 78.73081
## 145       51.27975 33.19883 69.36066 23.62737 78.93212
## 146       44.64146 25.77451 63.50840 15.78696 73.49596
## 147       50.91654 31.78674 70.04633 21.66004 80.17303
## 148       46.13250 26.91314 65.35186 16.73903 75.52597
## 149       50.14958 30.89952 69.39964 20.70915 79.59001
## 150       54.43030 35.16969 73.69091 24.97374 83.88686
## 151       52.23750 31.95740 72.51759 21.22177 83.25323
## 152       49.25973 28.64068 69.87878 17.72562 80.79384
## 153       54.39075 33.65642 75.12508 22.68033 86.10117
## 154       49.97347 29.19965 70.74730 18.20265 81.74430
## 155       51.42430 30.63691 72.21169 19.63273 83.21587
## 156       54.73984 33.94779 75.53190 22.94114 86.53855
```

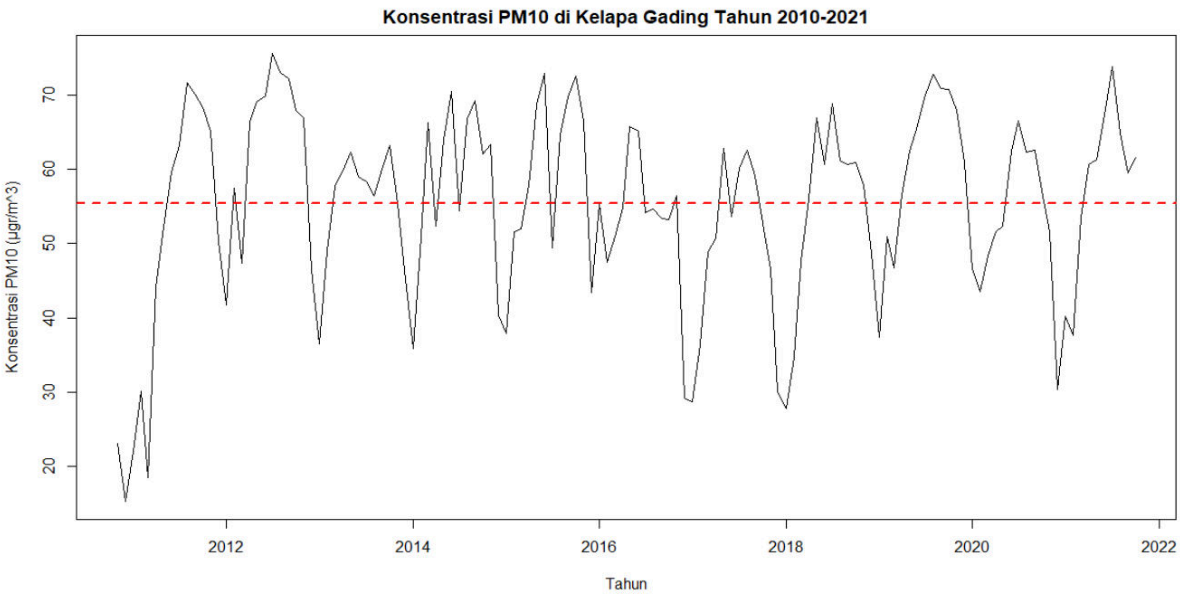
```
plot(forecast(mod_2, h=24, level=c(80,95)), xaxt='n',ylab="Konsentrasi PM10
(µgr/m^3)", xlab="Bulan/Tahun", main="Forecast SARIMA(1,0,0)(1,1,0)[6] untu
k Konsentrasi PM10 di Bunderan HI", asp=0.6)
axis(1, at=seq(1, 140,10), labels=format(seq(as.Date("2010/1/1"), by = "yea
r", length.out = 14),format("%b %Y")), cex.axis=0.8, xpd=TRUE)
```



3) *Syntax* pemodelan PM10 di stasiun DKI 2

Library yang digunakan

```
# 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)
# Untuk melakukan uji heteroskedastik
library(FinTS)

# Memanggil Data
setwd("E:/Praktikum ADW")
library(readr)
data <- read_csv("ispu_stasiun.csv",
    col_types = cols(`bulan/tahun` = col_date(format = "%m/%Y"),
        `DKI 1` = col_number(), `DKI 2` = col_number(),
        `DKI 3` = col_number(), `DKI 4` = col_number()))
# Membuat grafik data
plot(data$`bulan/tahun`,data$`DKI 2`, type = 'l', main = 'Konsentrasi PM10
di Kelapa Gading Tahun 2010-2021', xlab = "Tahun", ylab = "Konsentrasi PM10
(µgr/m^3)")
# Membuat garis rataan
abline(h=mean(data$`DKI 2`),lwd=2,
lty = 2, col ='red')
```



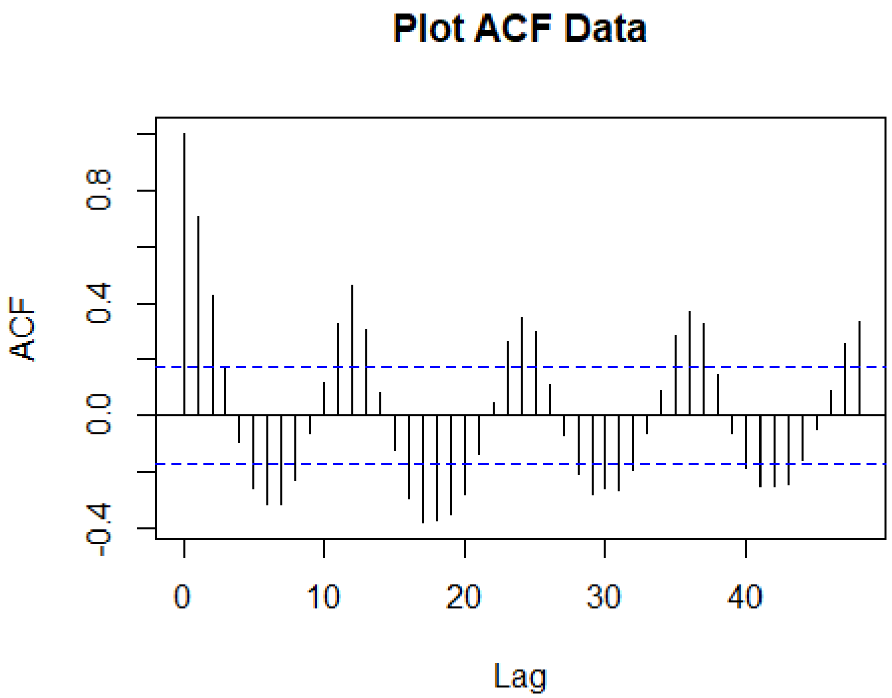

```
summary(data$`DKI 2`)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    15.26   48.85   57.79   55.40   65.14   75.71
```

```
sd(data$`DKI 2`)
```

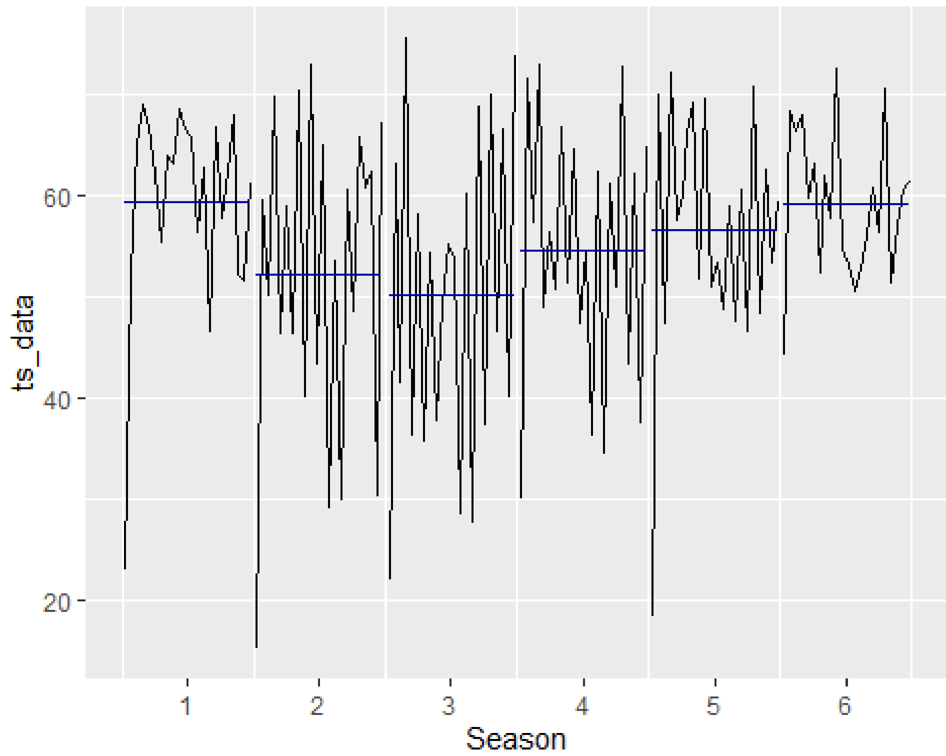
```
## [1] 12.89014
```

```
acf(data$`DKI 2`, lag = 48, main="Plot ACF Data")
```

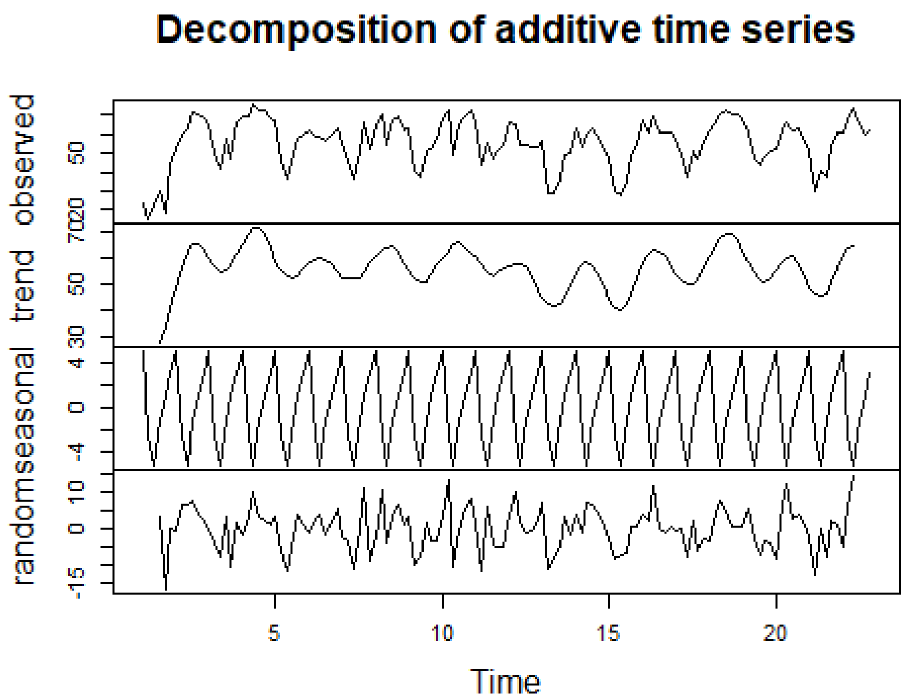


Pada grafik ACF dapat dilihat bahwa nilai ACF setiap lag kelipatan 6 lebih tinggi dari yang lain sehingga akan diasumsikan bahwa data memiliki pola musiman dengan $s = 6$

```
# Membuat data menjadi time series
ts_data <- ts(data$`DKI 2`, frequency = 6)
#Plot per subseries
ggsubseriesplot(ts_data)
```




```
plot(decompose(ts_data))
```

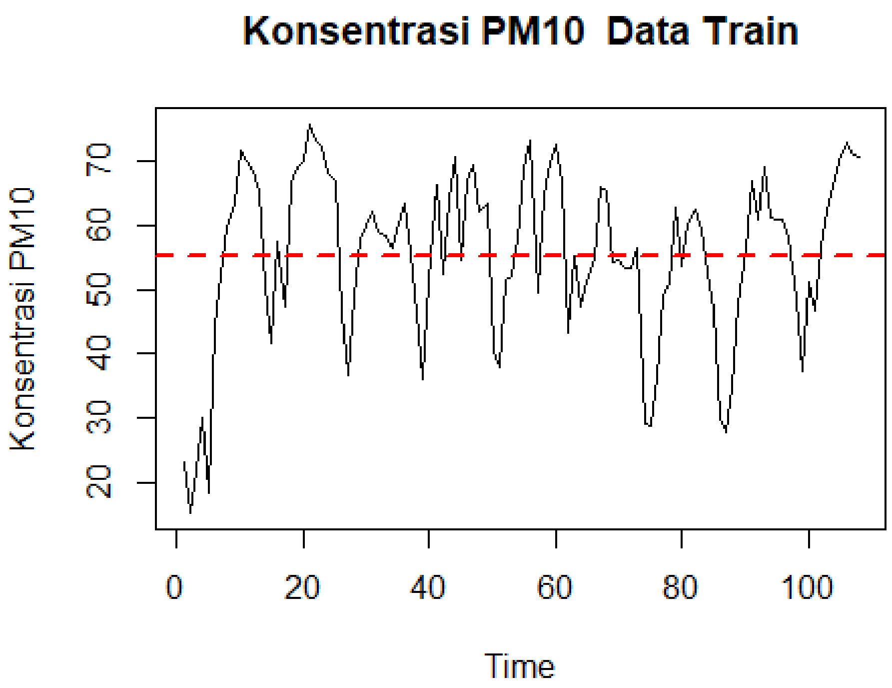


Pada data terdapat 132 observasi yang berarti terdapat 11 tahun. Dengan demikian, jumlah observasi yang akan menjadi data train adalah sebanyak 0.8 × 11 tahun ≈ 9 tahun = 114 observasi dan sisa datanya akan menjadi data validation

```
# Membuat data train dan validation dan mengubahnya menjadi data time serie
s
pm10 <- ts(data$`DKI 2`)
# data yang akan digunakan untuk membuat model
pm10_train <- ts(pm10[1:108])
# data yang akan digunakan untuk memvalidasi model
pm10_validation <-ts(pm10[108:132])
```

2. Identifikasi Model

```
# Membuat grafik data
plot(pm10_train , type = 'l', main = 'Konsentrasi PM10  Data Train', ylab =
"Konsentrasi PM10")
# Membuat garis rataan
abline(h=mean(pm10_train),lwd=2, lty = 2, col ='red')
```

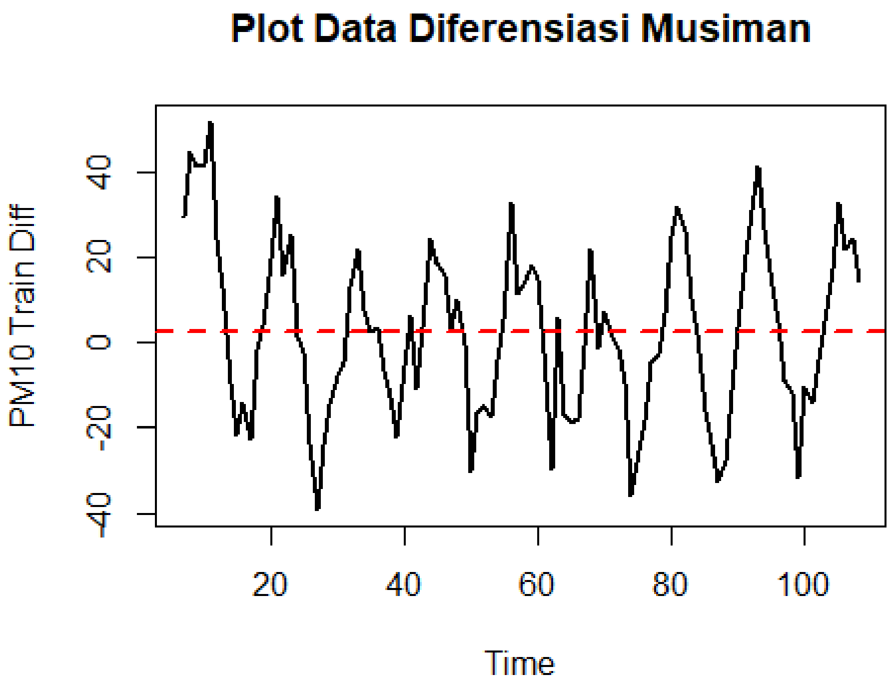



```
adf.test(pm10_train)

## Warning in adf.test(pm10_train): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train
## Dickey-Fuller = -5.9926, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

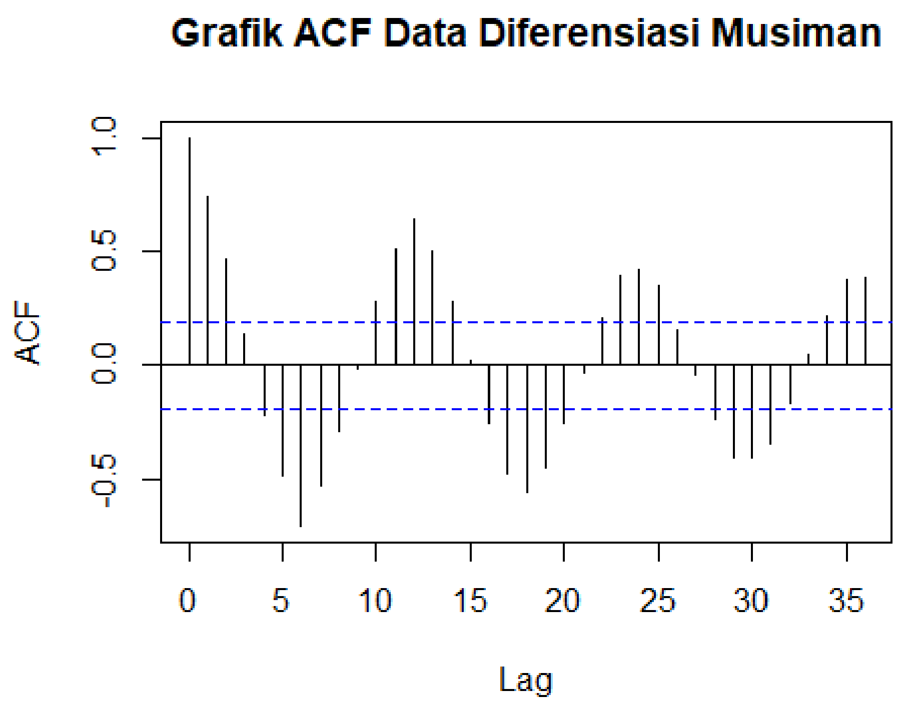
Diperoleh nilai p-value = 0.01 < $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data telah stasioner dalam rataan. sehingga akan dilakukan diferensiasi musiman

```
pm10_train_diff <- diff(pm10_train, lag = 6)
plot(pm10_train_diff,lwd = 2,main = 'Plot Data Diferensiasi Musiman', ylab=
```

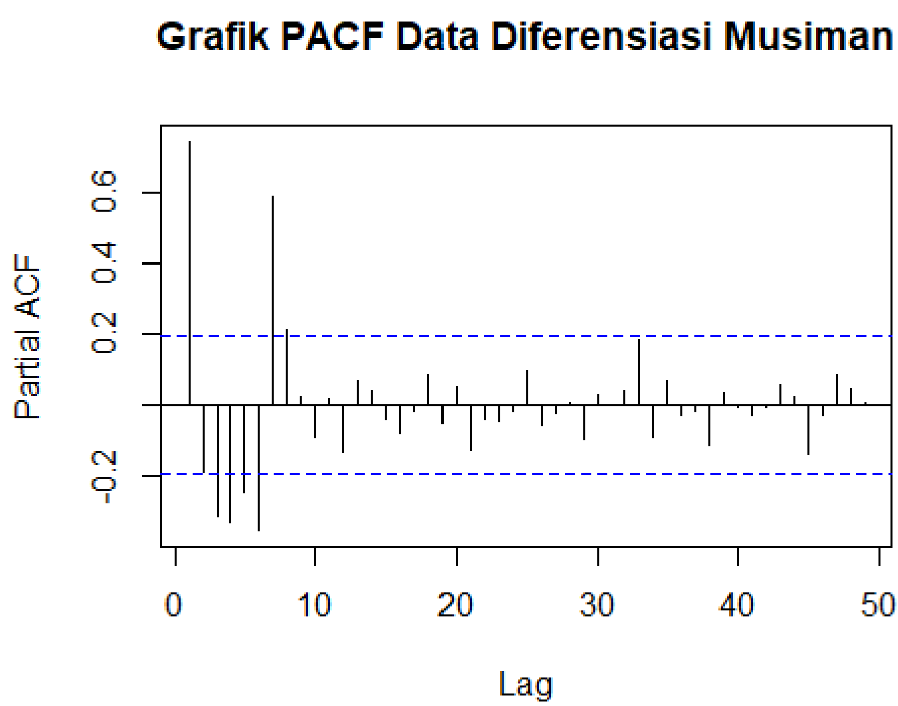
```
"PM10 Train Diff")
abline(h=mean(pm10_train_diff), lwd=2,lty = 2, col ='red')
```




```
adf.test(pm10_train_diff)

## Warning in adf.test(pm10_train_diff): p-value smaller than printed p-val
ue

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train_diff
## Dickey-Fuller = -8.489, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

acf(pm10_train_diff, main = "Grafik ACF Data Diferensiasi Musiman", lag.max
= 36)
```




```
pacf(pm10_train_diff, main = "Grafik PACF Data Diferensiasi Musiman", lag.max = 49)
```



Dari grafik-grafik tersebut dapat disimpulkan beberapa model yang akan dicoba, yaitu: 1. $SARIMA(0,0,2) \times (1,1,0)_6$ 2. $SARIMA(1,0,0) \times (1,1,0)_6$ 3. $SARIMA(1,0,1) \times (1,1,0)_6$ 4. $SARIMA(1,0,2) \times (1,1,0)_6$

3. PENAKSIRAN PARAMETER

```
mod_1 <- arima(pm10_train, order = c(0,0,2), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_1

##
## Call:
## arima(x = pm10_train, order = c(0, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ma1     ma2     sar1
##       0.5619  0.4358  -0.8033
## s.e.  0.0906  0.1089   0.0624
##
## sigma^2 estimated as 73.24:  log likelihood = -367.23,  aic = 742.46

mod_2 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_2

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1     sar1
##       0.7803  -0.8099
## s.e.  0.0713   0.0548
##
## sigma^2 estimated as 61.03:  log likelihood = -357.91,  aic = 721.82

mod_3 <- arima(pm10_train, order = c(1,0,1), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_3

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 1), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1     sar1
##       0.9273  -0.3296  -0.8643
## s.e.  0.0610   0.1190   0.0500
##
## sigma^2 estimated as 56.47:  log likelihood = -354.82,  aic = 717.63
```

```
mod_4 <- arima(pm10_train, order = c(1,0,2), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_4
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1     ma2      sar1
##       0.9076  -0.3391  0.0886  -0.8579
## s.e.  0.0738   0.1306  0.1139   0.0528
##
## sigma^2 estimated as 56.26:  log likelihood = -354.52,  aic = 719.05
```

```
mod_auto = auto.arima(pm10_train, max.p=4,max.q=4, max.P = 4, max.Q = 4,sea
sonal =TRUE, stationary = FALSE)
mod_auto
```

```
## Series: pm10_train
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     mean
##       1.5584  -0.6896  -0.7487  55.4196
## s.e.  0.1557   0.1142   0.1732   1.6896
##
## sigma^2 estimated as 83.21:  log likelihood=-390.45
## AIC=790.91   AICc=791.49   BIC=804.32
```

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(1,0,1) \times (1,1,0)_6$ namun akan dipilih model $SARIMA(1,0,0) \times (1,1,0)_6$ karena model ini menghasilkan galat validasi yang lebih kecil berdasarkan analisis lebih lanjut. Berikutnya, akan dilakukan uji signifikansi dari model.

4. Signifikansi dari Koefisien Parameter dan Pembuatan Model

```
coeftest(mod_2)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.780254   0.071269  10.948 < 2.2e-16 ***
## sar1 -0.809858   0.054791 -14.781 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

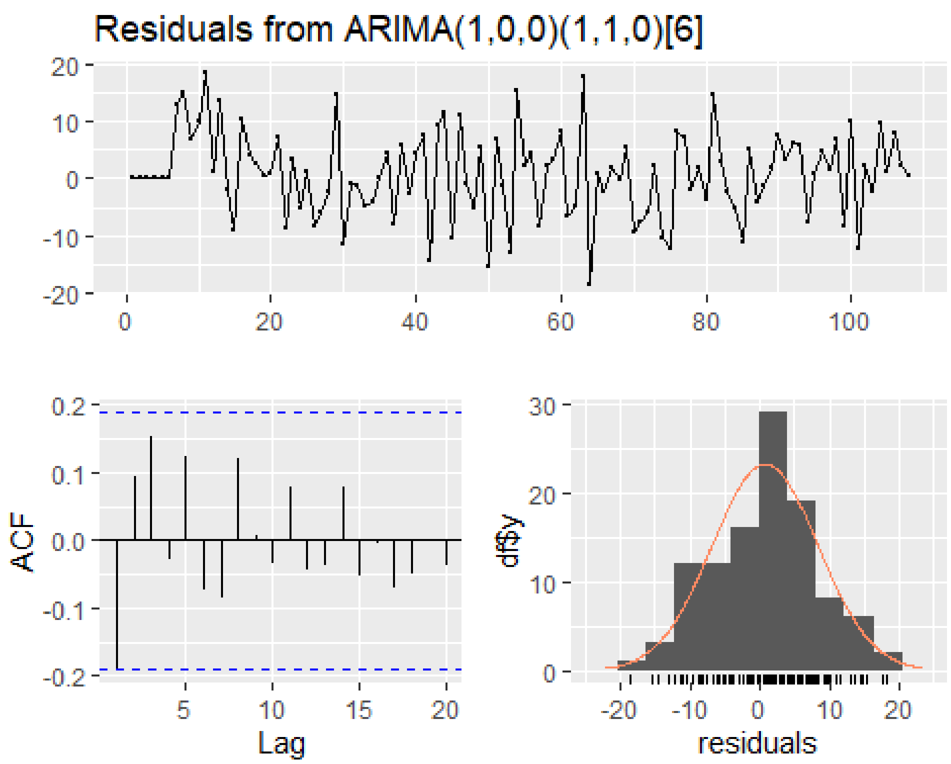
Dapat dilihat bahwa semua parameter signifikan sehingga dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memodelkan data. Berikutnya, akan dilihat performa model.

```
accuracy(mod_2)

##                     ME    RMSE      MAE       MPE     MAPE      MASE
ACF1
## Training set 0.7523712 7.59177 5.976177 0.1943559 11.30042 0.7834028 -0.
1917483
```

Dapat dilihat bahwa RMSE dari model sebesar 7.592 yang sangat kecil dibandingkan dengan standar deviasi data (= 12.89). Jadi, dapat disimpulkan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

5. Uji Diagnostik

```
checkresiduals(mod_2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[6]
## Q* = 12.98, df = 8, p-value = 0.1125
##
## Model df: 2.   Total lags used: 10
```

Perhatikan bahwa p-value = 0.1125 > $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data saling bebas. Hal ini juga diperkuat dari plot ACF yang tidak signifikan. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan plot residual terlihat memiliki rataan nol dan variansi yang konstan.

```
data_residuals <- residuals(mod_2)
ks.test(residuals(mod_2), "pnorm", mean=mean(data_residuals), sd=sd(data_re
siduals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  residuals(mod_2)
## D = 0.059726, p-value = 0.8358
## alternative hypothesis: two-sided
```

Uji Kolmogorov-Smirnov dengan hipotesis:
H_0: data berdistribusi normal
H_1: data tidak berdistribusi normal
Diperoleh p-value = 0.8358 > $\alpha = 0.05$ maka H_0 tidak ditolak dan dapat disimpulkan bahwa residual model berdistribusi normal.

```
ArchTest(data_residuals)

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  data_residuals
## Chi-squared = 7.1079, df = 12, p-value = 0.8504
```

Karena p-value = $0.8504 > \alpha = 0.05$, dapat disimpulkan bahwa data tidak perlu dimodelkan dengan model heteroskedastik. Hal ini didukung pula oleh variansi galat yang terlihat konstan pada plot galat.

5.1 Forecasting

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

```
validation <- forecast(pm10_train,model=mod_2,h=length(pm10_validation))
actual <- as.vector(pm10_validation)
rmse_validation <- sqrt(mean((as.vector(validation$mean)-actual)^2))
rmse_validation

## [1] 10.90673
```

Dapat dilihat bahwa RMSE dari model ≈ 10.907 yang masih lebih kecil dari standar deviasi data. Dengan demikian, dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memprediksi data.

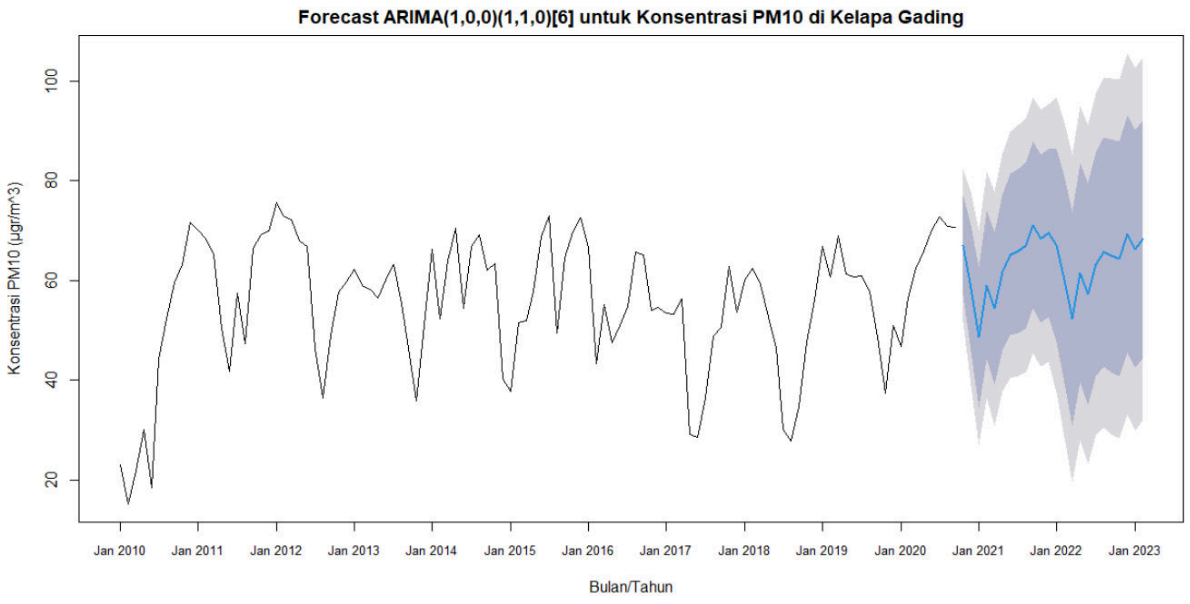
```
ape_validation <- abs((as.vector(validation$mean) -actual)/actual)*100
mape_validation <- mean(ape_validation)
mape_validation

## [1] 19.45461
```

Diperoleh juga MAPE sebesar ≈ 19.455

```
fc <- forecast(pm10,model=mod_2, h=24)
summary(fc)

##
## Forecast method: ARIMA(1,0,0)(1,1,0)[6]
##
```

```
## Model Information:
## Series: object
## ARIMA(1,0,0)(1,1,0)[6]
##
## Coefficients:
##          ar1     sar1
##       0.7803  -0.8099
## s.e.  0.0000   0.0000
##
## sigma^2 estimated as 61.03:  log likelihood=-440.19
## AIC=882.38   AICc=882.41   BIC=885.21
##
## Error measures:
##                     ME     RMSE      MAE        MPE     MAPE   MASE
ACF1
## Training set 0.5019865 7.565238 5.961525 -0.3123265 11.50245 0.8064 -0.1
741342
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## 133       56.38941 46.37811 66.40072 41.07844  71.70039
## 134       39.70816 27.00999 52.40634 20.28798  59.12834
## 135       48.43504 34.34991 62.52016 26.89369  69.97638
## 136       44.26175 29.39549 59.12802 21.52576  66.99774
## 137       55.76738 40.44505 71.08970 32.33390  79.20086
## 138       61.71389 46.12044 77.30733 37.86576  85.56201
## 139       61.07299 44.93353 77.21245 36.38982  85.75617
## 140       62.62455 46.16155 79.08756 37.44656  87.80254
## 141       69.43444 52.77754 86.09134 43.95991  94.90897
## 142       61.34541 44.57157 78.11926 35.69203  86.99879
## 143       59.03569 42.19105 75.88033 33.27403  84.79735
## 144       61.77427 44.88667 78.66187 35.94692  87.60162
## 145       57.42937 38.09756 76.76117 27.86392  86.99481
## 146       44.18210 23.50327 64.86093 12.55656  75.80764
## 147       52.51886 31.06134 73.97638 19.70241  85.33531
## 148       47.58104 25.66300 69.49909 14.06028  81.10180
## 149       56.44419 34.25047 78.63792 22.50181  90.38658
## 150       61.76857 39.40867 84.12847 27.57206  95.96509
## 151       60.41390 37.44018 83.38762 25.27863  95.54917
## 152       59.14418 35.80467 82.48368 23.44949  94.83887
## 153       66.23861 42.67920 89.79802 30.20760 102.26962
## 154       58.74424 35.05195 82.43653 22.51001  94.97848
## 155       58.55543 34.78261 82.32826 22.19804  94.91283
## 156       61.78294 37.96122 85.60465 25.35076  98.21511
```

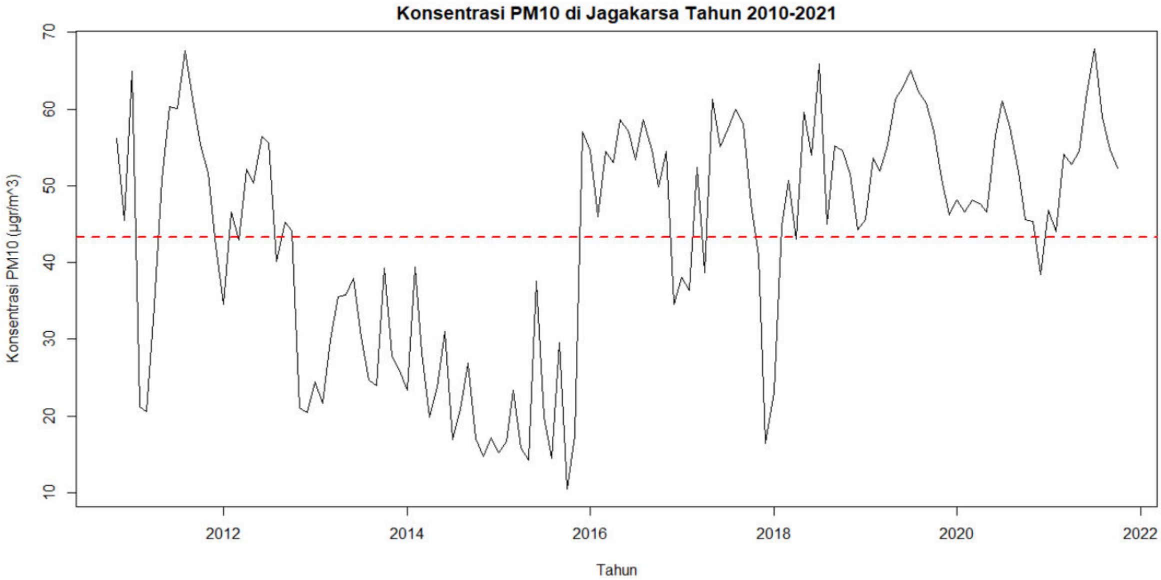
```
plot(forecast(mod_2, h=24, level=c(80,95)), xaxt='n',ylab="Konsentrasi PM10
(µgr/m^3)", xlab="Bulan/Tahun", main="Forecast ARIMA(1,0,0)(1,1,0)[6] untuk
Konsentrasi PM10 di Kelapa Gading")
axis(1, at=seq(1, 140,10), labels=format(seq(as.Date("2010/1/1"), by = "yea
r", length.out = 14),format("%b %Y")), cex.axis=0.8, xpd=TRUE)
```



4) *Syntax* pemodelan PM10 di stasiun DKI 3

Library yang digunakan

```
# 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)
# Untuk melakukan uji heteroskedastik
library(FinTS)

# Memanggil Data
setwd("E:/Praktikum ADW")
library(readr)
data <- read_csv("ispu_stasiun.csv",
    col_types = cols(`bulan/tahun` = col_date(format = "%m/%Y"),
        `DKI 1` = col_number(), `DKI 2` = col_number(),
        `DKI 3` = col_number(), `DKI 4` = col_number()))
# Membuat grafik data
plot(data$`bulan/tahun`,data$`DKI 3`, type = 'l', main = 'Konsentrasi PM10
di Jagakarsa Tahun 2010-2021', xlab = "Tahun", ylab = "Konsentrasi PM10 (µg
r/m^3)")
# Membuat garis rataan
abline(h=mean(data$`DKI 3`),lwd=2,
lty = 2, col ='red')
```



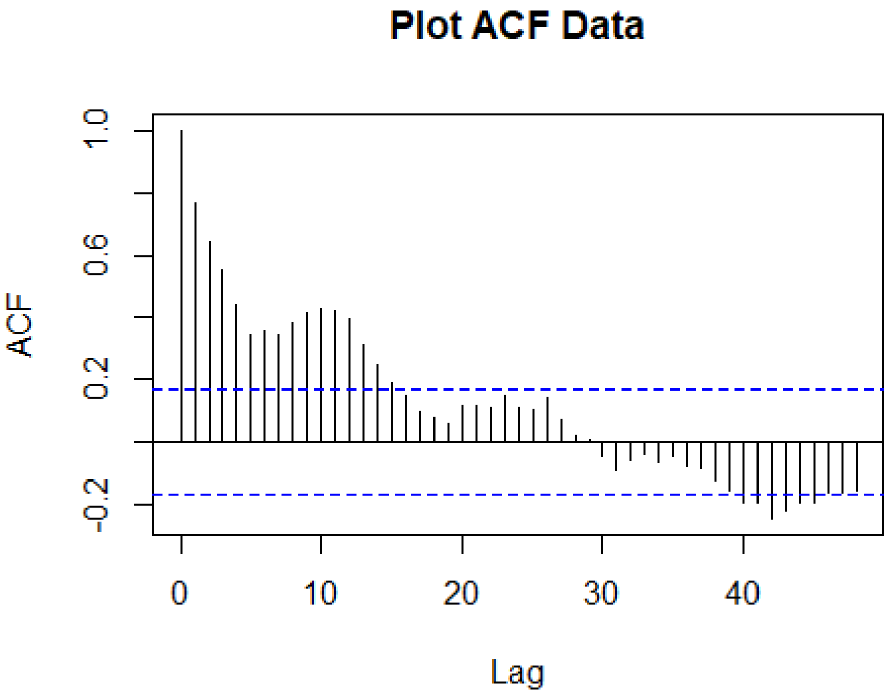

```r
summary(data$`DKI 3`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.52   30.21   46.57   43.25   55.11   67.90
```

```r
sd(data$`DKI 3`)
```

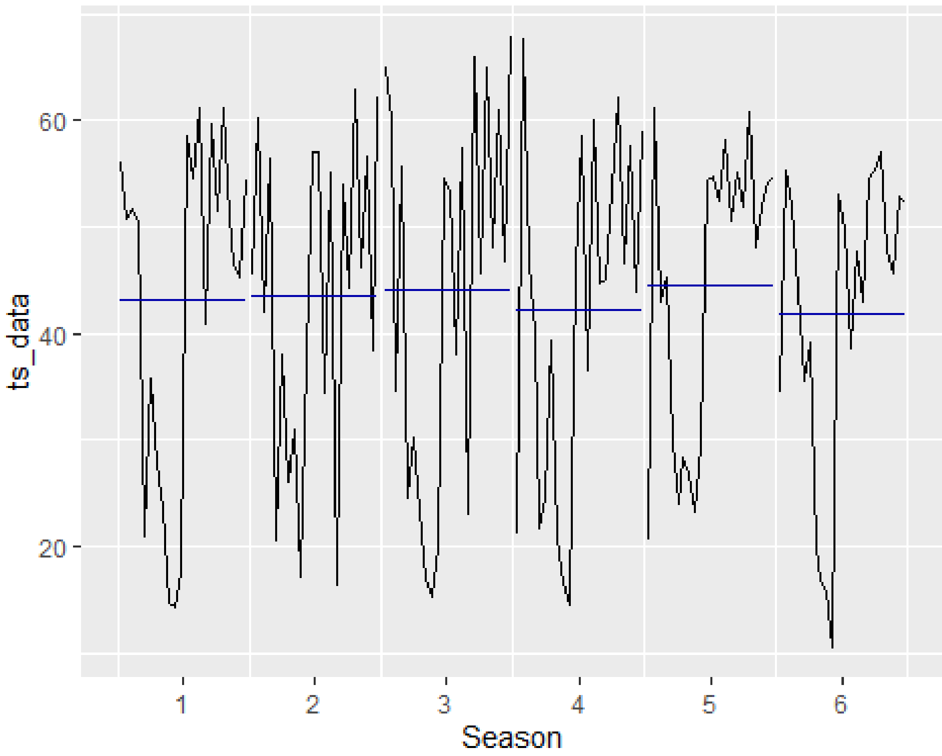
```
## [1] 15.24656
```

```r
acf(data$`DKI 3`, lag = 48, main="Plot ACF Data")
```

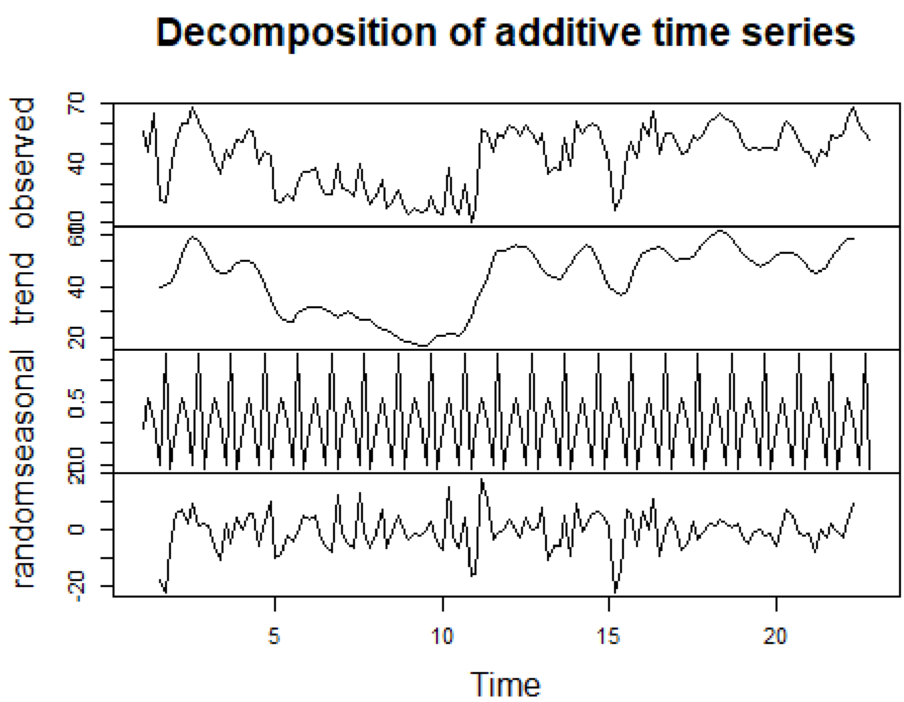


Seperti data di daerah stasiun DKI lainnya, akan diasumsikan bahwa data memiliki pola musiman dengan $s = 6$

```
# Membuat data menjadi time series
ts_data <- ts(data$`DKI 3`, frequency = 6)
#Plot per subseries
ggsubseriesplot(ts_data)
```




```
plot(decompose(ts_data))
```



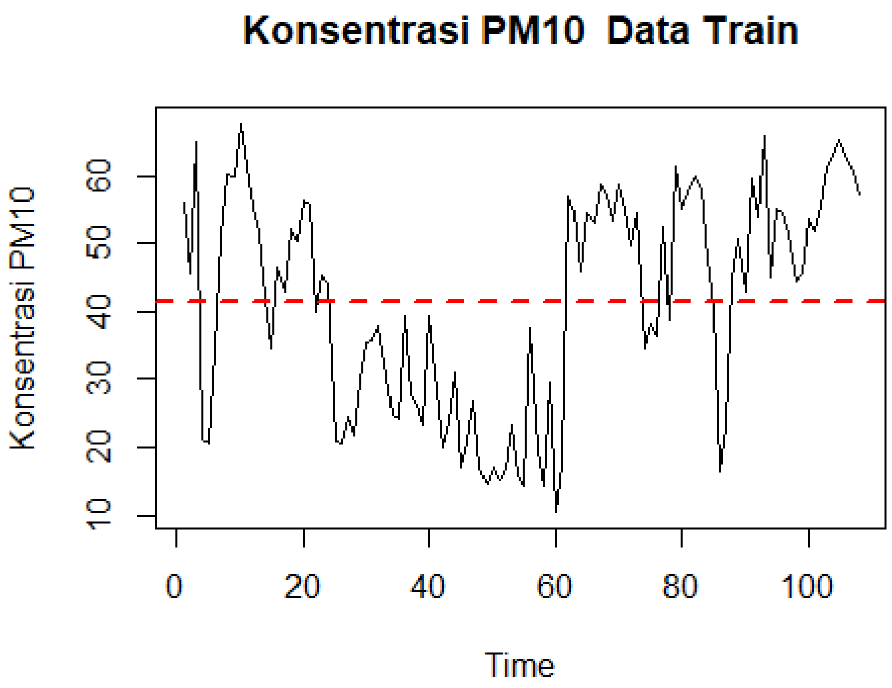
Dari plot subseries dan dekomposisi terlihat bahwa memang terdapat pola musiman dari data tetapi tidak begitu nampak di plot ACF yang mungkin disebabkan oleh data yang belum stasioner.

Pada data terdapat 132 observasi yang berarti terdapat 11 tahun. Dengan demikian, jumlah observasi yang akan menjadi data train adalah sebanyak 0.8×11 tahun ≈ 9 tahun = 114 observasi dan sisa datanya akan menjadi data validation

```
# Membuat data train dan validation dan mengubahnya menjadi data time serie
s
pm10 <- ts(data$`DKI 3`)
# data yang akan digunakan untuk membuat model
pm10_train <- ts(pm10[1:108])
# data yang akan digunakan untuk memvalidasi model
pm10_validation <-ts(pm10[108:132])
```

2. Identifikasi Model

```
# Membuat grafik data
plot(pm10_train , type = 'l', main = 'Konsentrasi PM10  Data Train', ylab =
"Konsentrasi PM10")
# Membuat garis rataan
abline(h=mean(pm10_train),lwd=2, lty = 2, col ='red')
```

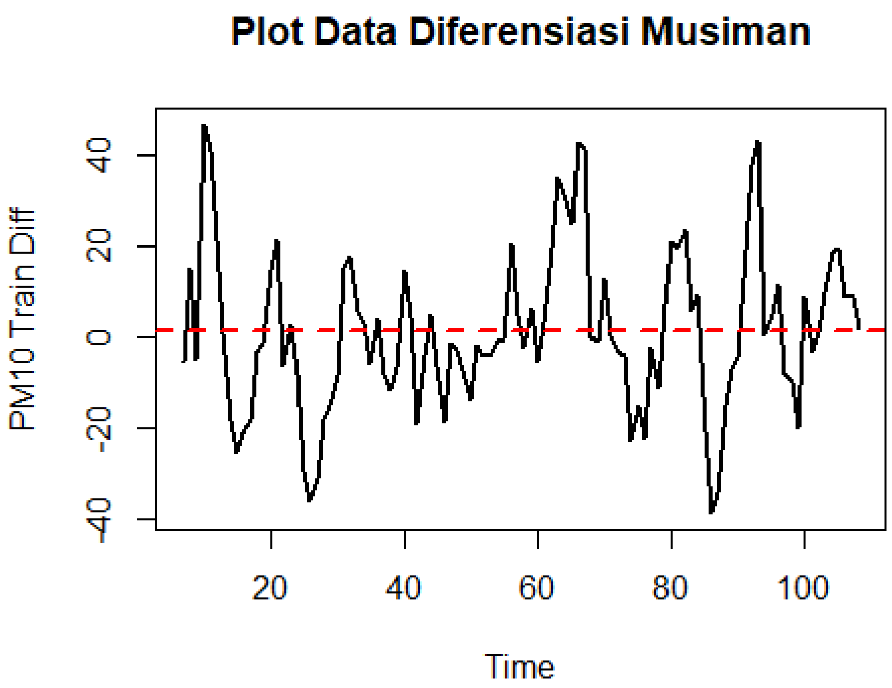



```
adf.test(pm10_train)

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train
```

```
## Dickey-Fuller = -2.7355, Lag order = 4, p-value = 0.2716
## alternative hypothesis: stationary
```

Diperoleh nilai p-value = 0.1913 > $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data belum stasioner sehingga akan dilakukan diferensiasi

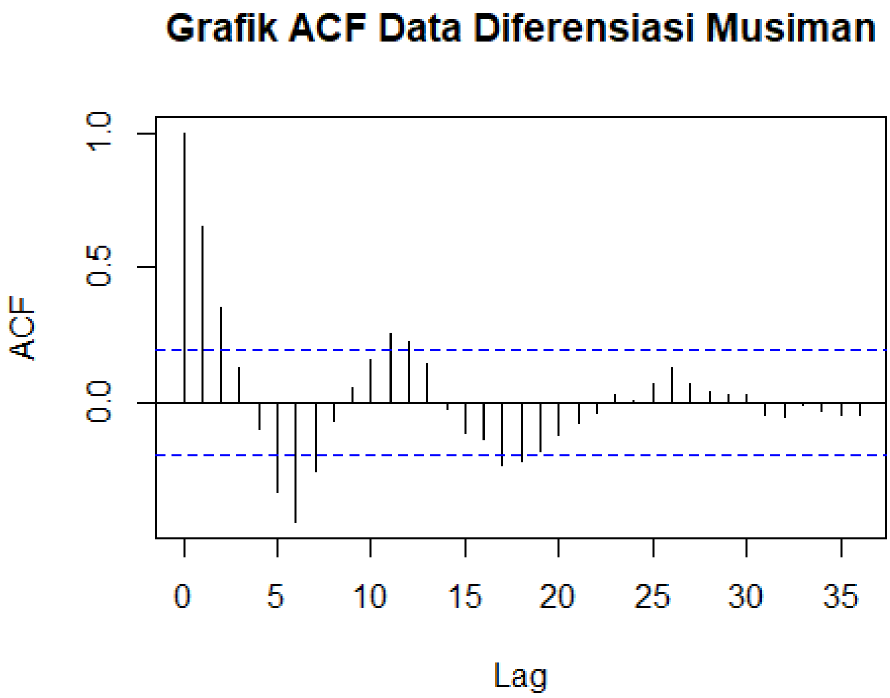
```
pm10_train_diff <- diff(pm10_train, lag = 6)
plot(pm10_train_diff,lwd = 2,main = 'Plot Data Diferensiasi Musiman', ylab=
"PM10 Train Diff")
abline(h=mean(pm10_train_diff), lwd=2,lty = 2, col ='red')
```



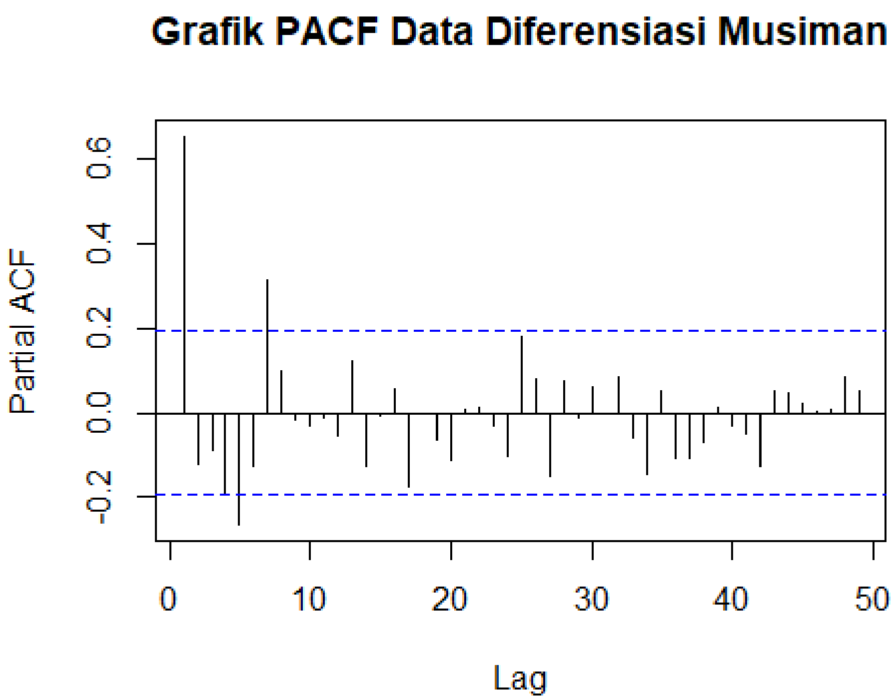

```
adf.test(pm10_train_diff)

## Warning in adf.test(pm10_train_diff): p-value smaller than printed p-val
ue

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train_diff
## Dickey-Fuller = -6.5861, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Diperoleh bahwa p-value < $\alpha = 0.05$ maka dapat disimpulkan bahwa data telah stasioner dan dapat langsugn ditentukan modelnya.

```
acf(pm10_train_diff, main = "Grafik ACF Data Diferensiasi Musiman", lag.max
= 36)
```




```
pacf(pm10_train_diff, main = "Grafik PACF Data Diferensiasi Musiman", lag.max = 49)
```



Dari grafik-grafik tersebut dapat disimpulkan beberapa model yang akan dicoba, yaitu: 1. $SARIMA(0,0,2) \times (1,1,1)_6$ 2. $SARIMA(1,0,0) \times (1,1,1)_6$ 3. $SARIMA(1,0,2) \times (1,1,1)_6$ 4. $SARIMA(0,0,2) \times (1,1,2)_6$ 5. $SARIMA(1,0,0) \times (1,1,2)_6$ 6. $SARIMA(1,0,2) \times (1,1,2)_6$ 7. $sARIMA(1,0,0) \times (0,1,1)_6$

3. PENAKSIRAN PARAMETER

```
mod_1 <- arima(pm10_train, order = c(0,0,2), seasonal = list(order = c(1,1,
1), period = 6),
               method = 'ML')
mod_1

##
## Call:
## arima(x = pm10_train, order = c(0, 0, 2), seasonal = list(order = c(1, 1
, 1),
##     period = 6), method = "ML")
##
## Coefficients:
##          ma1     ma2    sar1     sma1
##       0.6826  0.3052  0.1293  -0.6792
## s.e.  0.0969  0.1049  0.6059   0.5808
##
## sigma^2 estimated as 142:  log likelihood = -399.1,  aic = 808.2

mod_2 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(1,1,
1), period = 6),
               method = 'ML')
mod_2

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(1, 1
, 1),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1    sar1     sma1
##       0.7312  0.0760  -0.8076
## s.e.  0.0746  0.3487   0.3936
##
## sigma^2 estimated as 123:  log likelihood = -393.23,  aic = 794.46

mod_3 <- arima(pm10_train, order = c(1,0,2), seasonal = list(order = c(1,1,
1), period = 6),
               method = 'ML')
mod_3

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 2), seasonal = list(order = c(1, 1
, 1),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1      ma2     sar1     sma1
##       0.8466  -0.1775  -0.0696  -0.0282  -0.7894
## s.e.  0.1230   0.1721   0.1573   0.2722   0.2893
##
## sigma^2 estimated as 121.6:  log likelihood = -392.72,  aic = 797.45
```

```r
mod_4 <- arima(pm10_train, order = c(0,0,2), seasonal = list(order = c(1,1,
2), period = 6),
               method = 'ML')
mod_4
```

```
##
## Call:
## arima(x = pm10_train, order = c(0, 0, 2), seasonal = list(order = c(1, 1
, 2),
##     period = 6), method = "ML")
##
## Coefficients:
##          ma1     ma2     sar1    sma1     sma2
##       0.6436  0.2773  -0.9989  0.4708  -0.5113
## s.e.  0.0983  0.0856   0.0111  0.1251   0.1162
##
## sigma^2 estimated as 136.4:  log likelihood = -398.32,  aic = 808.63
```

```r
mod_5 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(1,1,
2), period = 6),
               method = 'ML')
```

```
## Warning in arima(pm10_train, order = c(1, 0, 0), seasonal = list(order =
c(1, :
## possible convergence problem: optim gave code = 1
```

```r
mod_5
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(1, 1
, 2),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1     sar1    sma1     sma2
##       0.7190  -0.9940  0.2877  -0.6669
## s.e.  0.0764   0.0341  0.1496   0.1370
##
## sigma^2 estimated as 120.7:  log likelihood = -392.48,  aic = 794.96
```

```r
mod_6 <- arima(pm10_train, order = c(1,0,2), seasonal = list(order = c(1,1,
2), period = 6),
               method = 'ML')
```

```
## Warning in arima(pm10_train, order = c(1, 0, 2), seasonal = list(order =
c(1, :
## possible convergence problem: optim gave code = 1
```

```r
mod_6
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 2), seasonal = list(order = c(1, 1
, 2),
##     period = 6), method = "ML")
```

```
##
## Coefficients:
##          ar1      ma1      ma2     sar1    sma1     sma2
##       0.8801  -0.2479  -0.0899  -0.9972  0.1764  -0.7851
## s.e.  0.1130   0.1662   0.1558   0.0203  0.1846   0.1766
##
## sigma^2 estimated as 114.8:  log likelihood = -391.5,  aic = 797
```

```
mod_7 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(0,1,
1), period = 6),
               method = 'ML')
mod_7
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(0, 1
, 1),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1     sma1
##       0.7272  -0.7294
## s.e.  0.0736   0.1233
##
## sigma^2 estimated as 124.4:  log likelihood = -393.26,  aic = 792.51
```

```
mod_auto = auto.arima(pm10_train, max.p=4,max.q=4, max.P = 4, max.Q = 4,sea
sonal =TRUE, stationary = FALSE)
mod_auto
```

```
## Series: pm10_train
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.5339  -0.8537
## s.e.  0.1367   0.0845
##
## sigma^2 estimated as 110.6:  log likelihood=-402.79
## AIC=811.59   AICc=811.82   BIC=819.6
```

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(1,0,0) \times (0,1,1)_6$ sehingga akan dipilih model tersebut untuk dianalisis lebih lanjut. Berikutnya, akan dilakukan uji signifikansi dari model.

4. Signifikansi dari Koefisien Parameter dan Pembuatan Model

```
coeftest(mod_7)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.727201   0.073618  9.8781 < 2.2e-16 ***
## sma1 -0.729376   0.123282 -5.9163 3.293e-09 ***
```

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

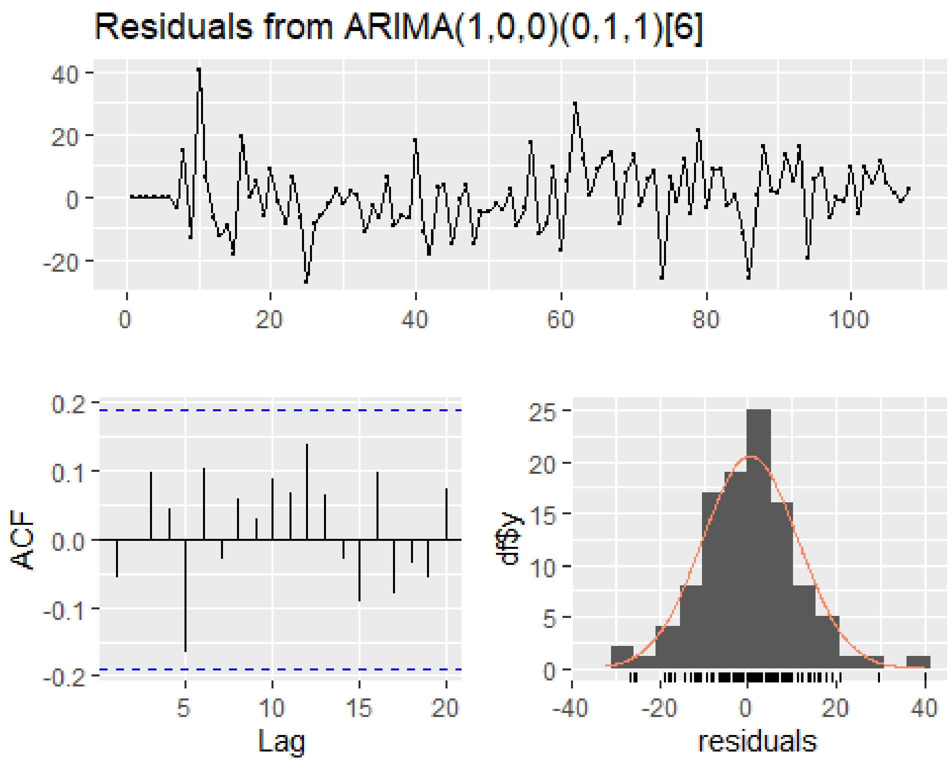
Dapat dilihat bahwa semua parameter signifikan sehingga dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memodelkan data. Berikutnya, akan dilihat performa model.

```
accuracy(mod_7)

##                    ME     RMSE      MAE       MPE    MAPE      MASE
ACF1
## Training set 0.418365 10.83773 8.096342 -7.058697 24.1523 0.9990044 -0.0
5699761
```

Dapat dilihat bahwa RMSE dari model sebesar 10.838 yang sangat kecil dibandingkan dengan standar deviasi data (= 15.247). Jadi, dapat disimpulkan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

5. Uji Diagnostik

```
checkresiduals(mod_7)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[6]
## Q* = 7.6364, df = 8, p-value = 0.4698
##
## Model df: 2.   Total lags used: 10
```

Perhatikan bahwa p-value = 0.4698 > $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data saling bebas. Hal ini juga diperkuat dari plot ACF yang tidak signifikan. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan plot residual terlihat memiliki rataan nol dan variansi yang konstan.

```
data_residuals <- residuals(mod_7)
ks.test(residuals(mod_7), "pnorm", mean=mean(data_residuals), sd=sd(data_re
siduals))

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  residuals(mod_7)
## D = 0.058919, p-value = 0.8476
## alternative hypothesis: two-sided
```

Uji Kolmogorov-Smirnov dengan hipotesis:
H_0: data berdistribusi normal
H_1: data tidak berdistribusi normal
Diperoleh p-value = 0.8476 > $\alpha = 0.05$ maka H_0 tidak ditolak dan dapat disimpulkan bahwa residual model berdistribusi normal.

```
ArchTest(data_residuals)

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  data_residuals
## Chi-squared = 9.377, df = 12, p-value = 0.6704
```

Karena p-value = 0.6704 > $\alpha = 0.05$, dapat disimpulkan bahwa data tidak perlu dimodelkan dengan model heteroskedastik. Hal ini didukung pula oleh variansi galat yang terlihat konstan pada plot galat.

5.1 Forecasting

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

```
validation <- forecast(pm10_train,model=mod_7,h=length(pm10_validation))
actual <- as.vector(pm10_validation)
rmse_validation <- sqrt(mean((as.vector(validation$mean)-actual)^2))
rmse_validation

## [1] 7.055449
```

Dapat dilihat bahwa RMSE dari model ≈ 7.0554 yang lebih kecil dibandingkan dengan RMSE yang diperoleh sebelumnya. Dengan demikian, dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memprediksi data.

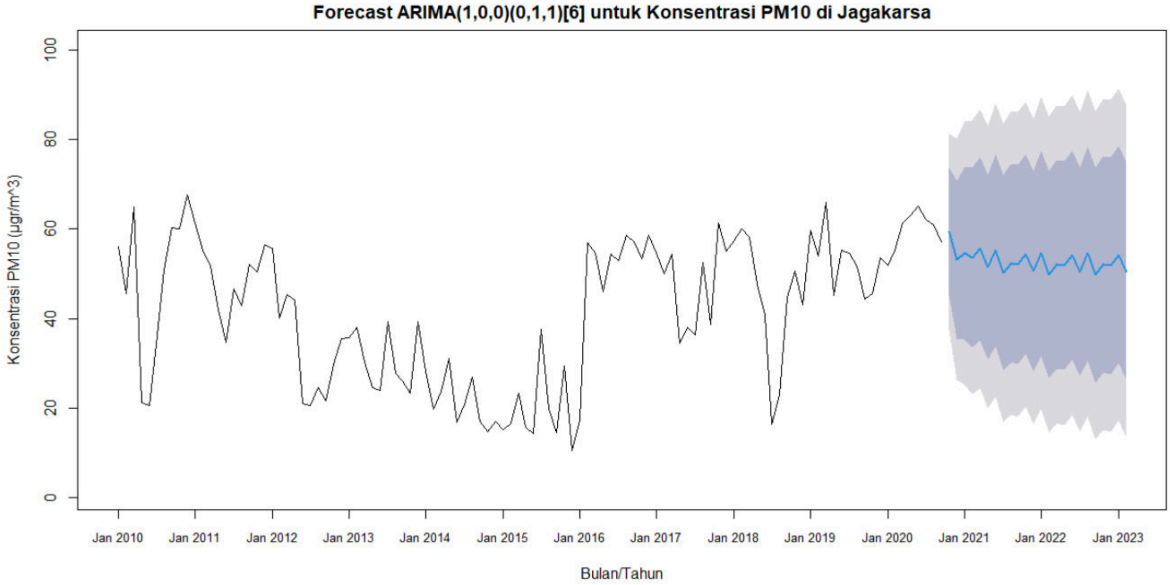
```
ape_validation <- abs((as.vector(validation$mean) -actual)/actual)*100
mape_validation <- mean(ape_validation)
mape_validation
```

```
## [1] 11.82231
```

Diperoleh juga MAPE sebesar ≈ 11.822

```
fc <- forecast(pm10,model=mod_7, h=24)
summary(fc)

##
## Forecast method: ARIMA(1,0,0)(0,1,1)[6]
##
## Model Information:
## Series: object
## ARIMA(1,0,0)(0,1,1)[6]
##
## Coefficients:
##          ar1     sma1
##       0.7272  -0.7294
## s.e.  0.0000   0.0000
##
## sigma^2 estimated as 124.4:  log likelihood=-475.17
## AIC=952.33   AICc=952.37   BIC=955.17
##
## Error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3605194 10.06241 7.393646 -5.88345 21.24156 0.9932244
##                     ACF1
## Training set -0.04643071
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 133       52.53454 38.24276 66.82632 30.67715 74.39193
## 134       52.44390 34.77276 70.11504 25.41823 79.46958
## 135       56.91170 37.69219 76.13120 27.51800 86.30540
## 136       53.01259 33.02270 73.00248 22.44069 83.58449
## 137       53.73659 33.35107 74.12211 22.55962 84.91356
## 138       50.69116 30.09949 71.28283 19.19892 82.18340
## 139       51.37477 29.93200 72.81754 18.58089 84.16865
## 140       51.60052 29.72105 73.47999 18.13876 85.06228
## 141       56.29839 34.19147 78.40530 22.48878 90.10800
## 142       52.56659 30.34034 74.79284 18.57447 86.55871
## 143       53.41226 31.12316 75.70136 19.32402 87.50050
## 144       50.45531 28.13304 72.77758 16.31634 84.59427
## 145       51.20326 28.38074 74.02578 16.29922 86.10729
## 146       51.47580 28.39311 74.55848 16.17388 86.77772
## 147       56.20769 32.98860 79.42677 20.69716 91.71821
## 148       52.50063 29.20974 75.79153 16.88028 88.12098
## 149       53.36430 30.03552 76.69307 17.68601 89.04258
## 150       50.42043 27.07164 73.76922 14.71154 86.12932
## 151       51.17789 27.38598 74.96981 14.79130 87.56449
## 152       51.45735 27.43440 75.48030 14.71742 88.19728
## 153       56.19427 32.05004 80.33850 19.26886 93.11969
## 154       52.49088 28.28276 76.69900 15.46776 89.51400
## 155       53.35720 29.11536 77.59904 16.28251 90.43189
## 156       50.41527 26.15562 74.67492 13.31334 87.51720
```

```
plot(forecast(mod_7, h=24, level=c(80,95)), xaxt='n',ylab="Konsentrasi PM10
(µgr/m^3)", xlab="Bulan/Tahun", main="Forecast ARIMA(1,0,0)(0,1,1)[6] untuk
Konsentrasi PM10 di Jagakarsa", asp=0.6)
axis(1, at=seq(1, 140,10), labels=format(seq(as.Date("2010/1/1"), by = "yea
r", length.out = 14),format("%b %Y")), cex.axis=0.8, xpd=TRUE)
```

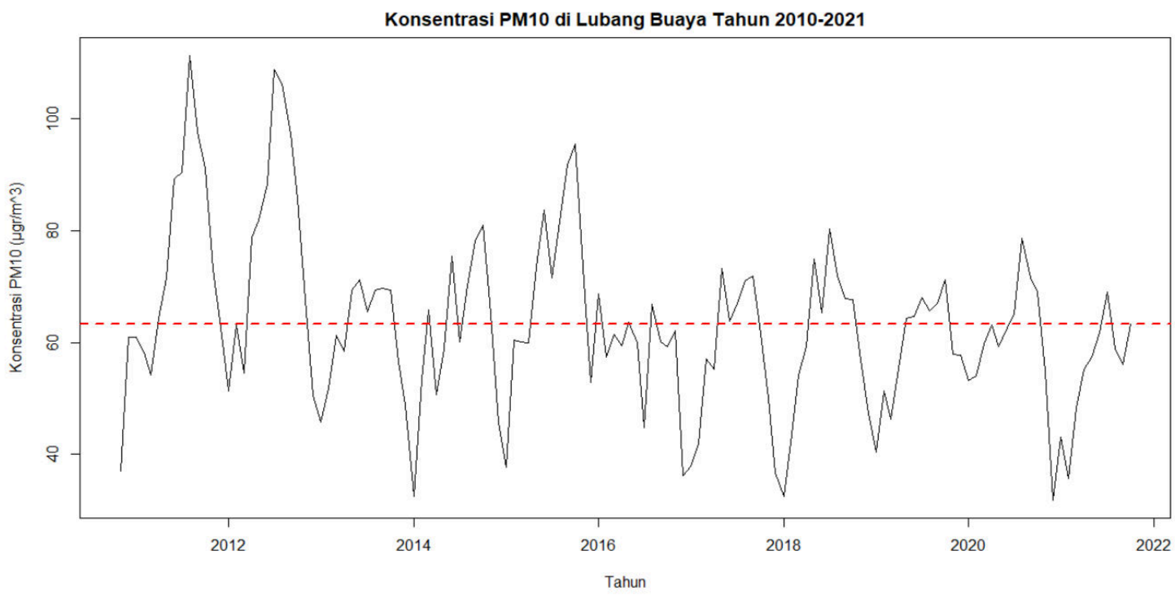


5) *Syntax* pemodelan PM10 di stasiun DKI 4

Library yang digunakan

```
# 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)
# Untuk melakukan uji heteroskedastik
library(FinTS)

# Memanggil Data
setwd("E:/Praktikum ADW")
library(readr)
data <- read_csv("ispu_stasiun.csv",
    col_types = cols(`bulan/tahun` = col_date(format = "%m/%Y"),
        `DKI 1` = col_number(), `DKI 2` = col_number(),
        `DKI 3` = col_number(), `DKI 4` = col_number()))
# Membuat grafik data
plot(data$`bulan/tahun`,data$`DKI 4`, type = 'l', main = 'Konsentrasi PM10
di Lubang Buaya Tahun 2010-2021', xlab = "Tahun", ylab = "Konsentrasi PM10
(µgr/m^3)")
# Membuat garis rataan
```

```
abline(h=mean(data$`DKI 4`),lwd=2,
lty = 2, col ='red')
```



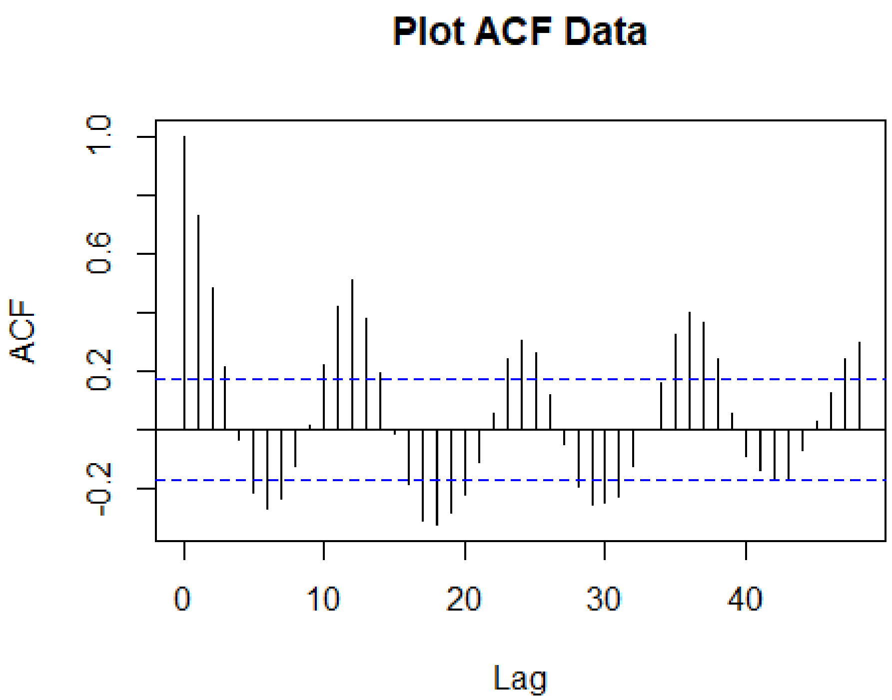

```
summary(data$`DKI 4`)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    31.81   54.83   61.93   63.36   71.09  111.43
```

```
sd(data$`DKI 4`)
```

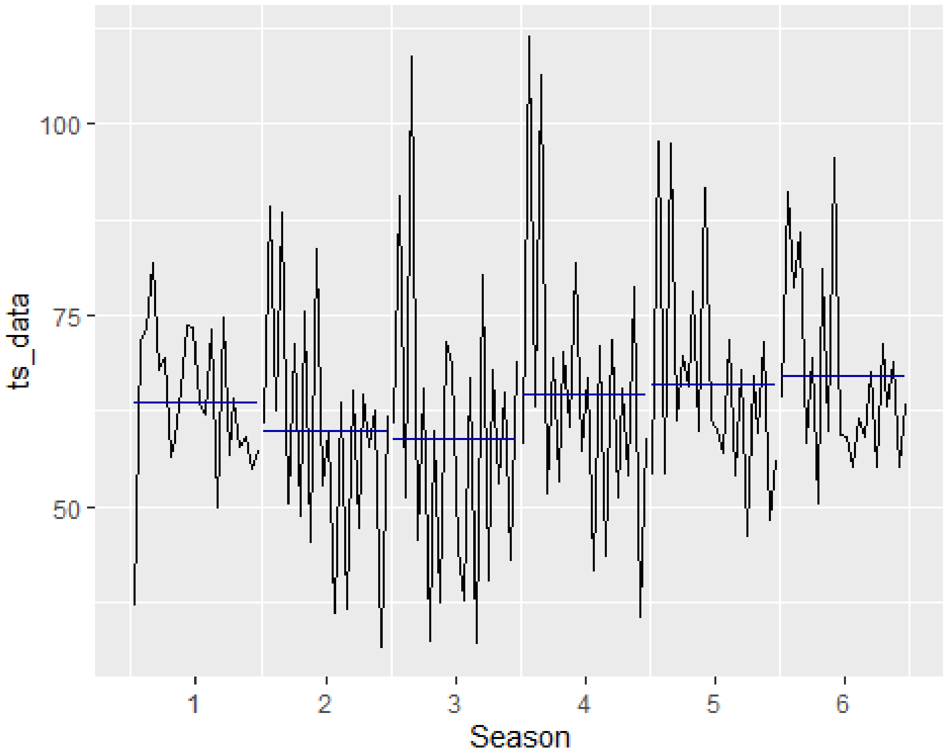
```
## [1] 15.40222
```

```
acf(data$`DKI 4`, lag = 48, main="Plot ACF Data")
```

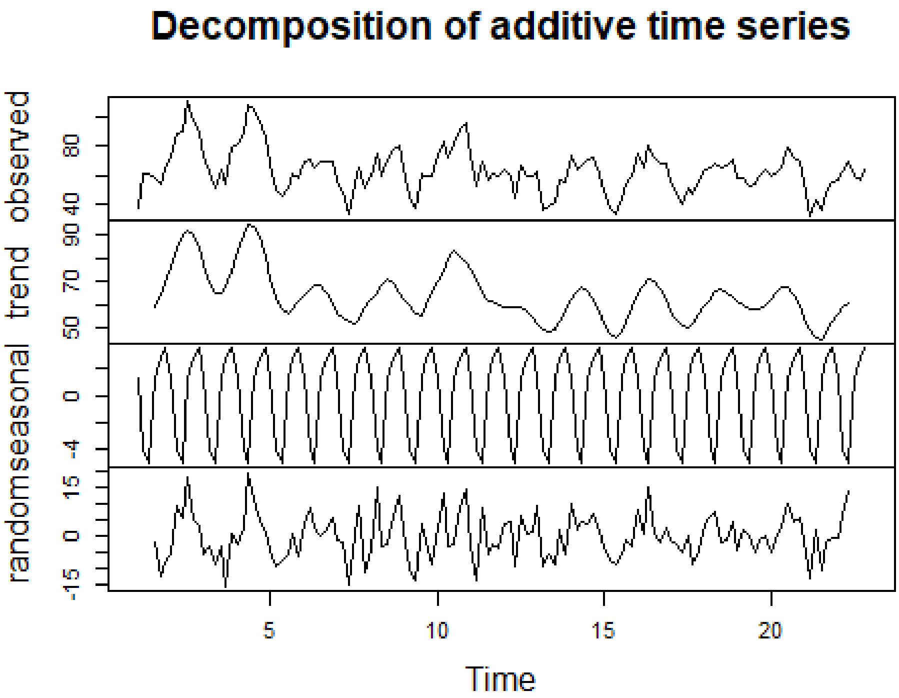


Pada grafik ACF dapat dilihat bahwa nilai ACF setiap lag kelipatan 6 lebih tinggi dari yang lain sehingga akan diasumsikan bahwa data memiliki pola musiman dengan $s = 6$

```
# Membuat data menjadi time series
ts_data <- ts(data$`DKI 4`, frequency = 6)
#Plot per subseries
ggsubseriesplot(ts_data)
```




```
plot(decompose(ts_data))
```

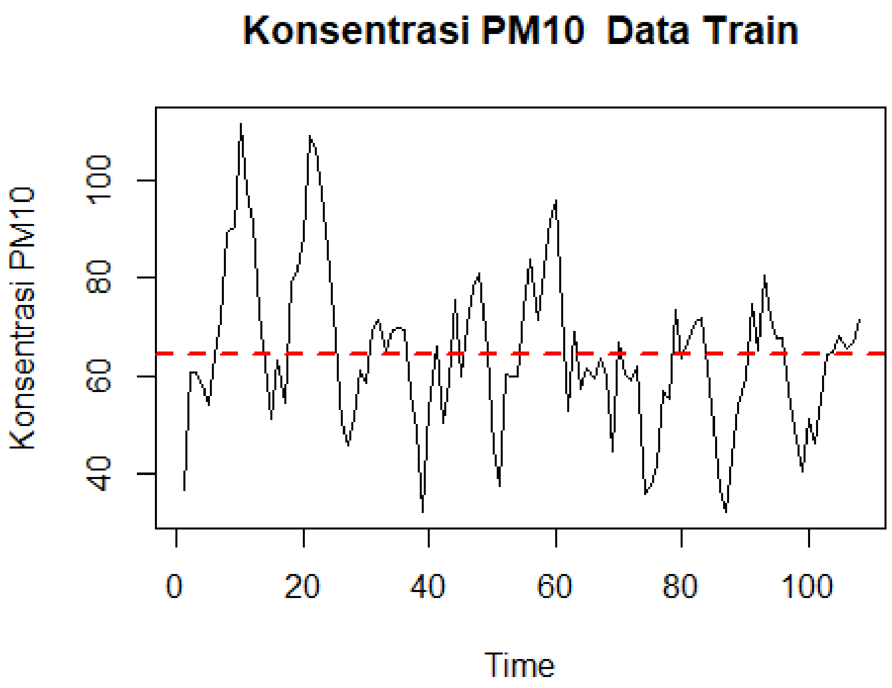


Pada data terdapat 132 observasi yang berarti terdapat 11 tahun. Dengan demikian, jumlah observasi yang akan menjadi data train adalah sebanyak 0.8×11 tahun ≈ 9 tahun = 108 observasi dan sisa datanya akan menjadi data validation

```
# Membuat data train dan validation dan mengubahnya menjadi data time serie
s
pm10 <- ts(data$`DKI 4`)
# data yang akan digunakan untuk membuat model
pm10_train <- ts(pm10[1:108])
# data yang akan digunakan untuk memvalidasi model
pm10_validation <-ts(pm10[108:132])
```

2. Identifikasi Model

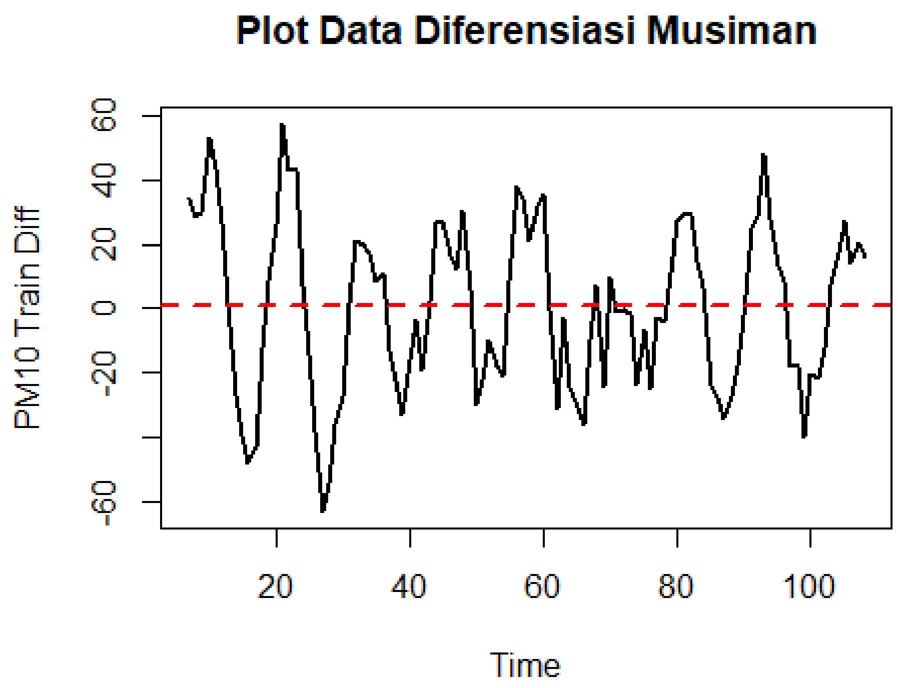
```
# Membuat grafik data
plot(pm10_train , type = 'l', main = 'Konsentrasi PM10  Data Train', ylab =
"Konsentrasi PM10")
# Membuat garis rataan
abline(h=mean(pm10_train),lwd=2, lty = 2, col ='red')
```



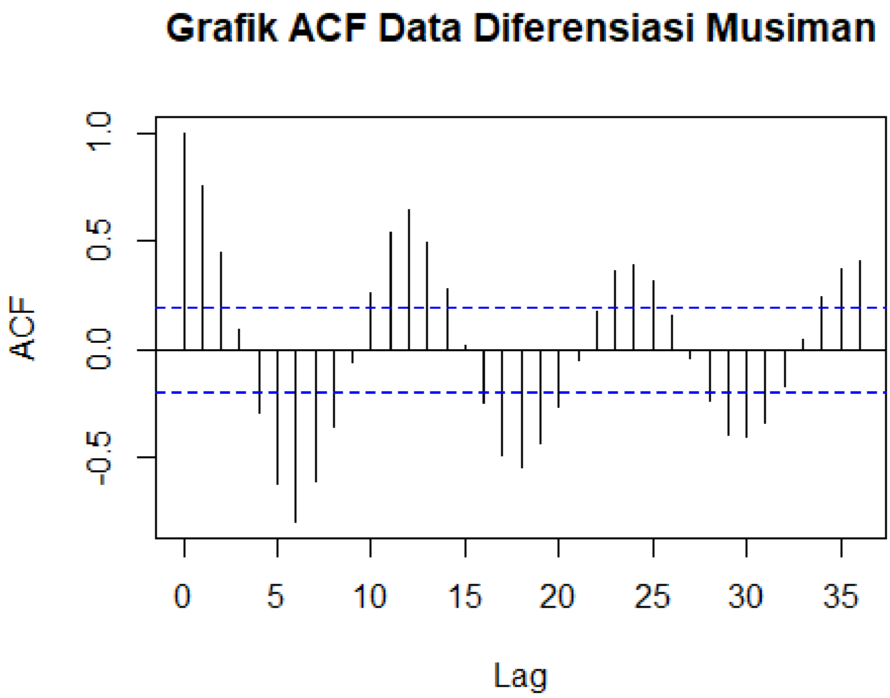

```
adf.test(pm10_train)

## Warning in adf.test(pm10_train): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train
## Dickey-Fuller = -6.2525, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Diperoleh nilai p-value = 0.01 < $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data telah stasioner dalam rataan. sehingga akan dilakukan diferensiasi musiman

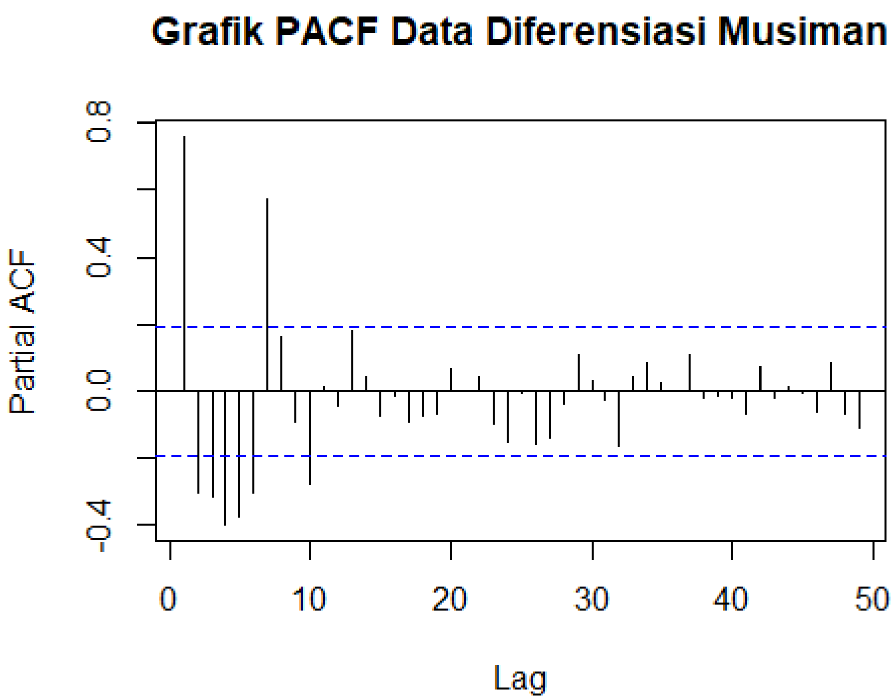
```
pm10_train_diff <- diff(pm10_train, lag = 6)
plot(pm10_train_diff,lwd = 2,main = 'Plot Data Diferensiasi Musiman', ylab=
"PM10 Train Diff")
abline(h=mean(pm10_train_diff), lwd=2,lty = 2, col ='red')
```




```
adf.test(pm10_train_diff)

## Warning in adf.test(pm10_train_diff): p-value smaller than printed p-val
ue

##
##  Augmented Dickey-Fuller Test
##
## data:  pm10_train_diff
## Dickey-Fuller = -9.8303, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

acf(pm10_train_diff, main = "Grafik ACF Data Diferensiasi Musiman", lag.max
= 36)
```




```
pacf(pm10_train_diff, main = "Grafik PACF Data Diferensiasi Musiman", lag.max = 49)
```



Dari grafik-grafik tersebut dapat disimpulkan beberapa model yang akan dicoba, yaitu: 1. $SARIMA(0,0,2) \times (1,1,0)_6$ 2. $SARIMA(1,0,0) \times (1,1,0)_6$ 3. $SARIMA(1,0,1) \times (1,1,0)_6$ 4. $SARIMA(1,0,2) \times (1,1,0)_6$

3. PENAKSIRAN PARAMETER

```
mod_1 <- arima(pm10_train, order = c(0,0,2), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_1

##
## Call:
## arima(x = pm10_train, order = c(0, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ma1     ma2      sar1
##       0.5572  0.2836   -0.7970
## s.e.  0.1072  0.0927    0.0597
##
## sigma^2 estimated as 117.8:  log likelihood = -391.18,  aic = 790.36
```

```
mod_2 <- arima(pm10_train, order = c(1,0,0), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_2

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 0), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1     sar1
##       0.6886  -0.7936
## s.e.  0.0774   0.0567
##
## sigma^2 estimated as 103.8:  log likelihood = -384.73,  aic = 775.46
```

```
mod_3 <- arima(pm10_train, order = c(1,0,1), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_3

##
## Call:
## arima(x = pm10_train, order = c(1, 0, 1), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1     sar1
##       0.8000  -0.2227  -0.8308
## s.e.  0.0837   0.1309   0.0558
##
## sigma^2 estimated as 100.5:  log likelihood = -383.5,  aic = 774.99
```

```
mod_4 <- arima(pm10_train, order = c(1,0,2), seasonal = list(order = c(1,1,
0), period = 6),
               method = 'ML')
mod_4
```

```
##
## Call:
## arima(x = pm10_train, order = c(1, 0, 2), seasonal = list(order = c(1, 1
, 0),
##     period = 6), method = "ML")
##
## Coefficients:
##          ar1      ma1     ma2     sar1
##       0.7675  -0.2982  0.2019  -0.8321
## s.e.  0.0898   0.1598  0.1667   0.0547
##
## sigma^2 estimated as 98.61:  log likelihood = -382.67,  aic = 775.34
```

```
mod_auto = auto.arima(pm10_train, max.p=4,max.q=4, max.P = 4, max.Q = 4,sea
sonal =TRUE, stationary = FALSE)
mod_auto
```

```
## Series: pm10_train
## ARIMA(0,1,0)
##
## sigma^2 estimated as 132.3:  log likelihood=-413.17
## AIC=828.34   AICc=828.38   BIC=831.02
```

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(1,0,1) \times (1,1,0)_6$ namun akan dipilih model $SARIMA(1,0,0) \times (1,1,0)_6$ untuk dianalisis lebih lanjut karena memiliki nilai AIC yang tidak berbeda jauh tetapi memiliki parameter yang lebih sedikit. Berikutnya, akan dilakukan uji signifikansi dari model.

4. Signifikansi dari Koefisien Parameter dan Pembuatan Model

```
coeftest(mod_2)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.688591   0.077381   8.8987 < 2.2e-16 ***
## sar1 -0.793650   0.056734 -13.9891 < 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. Berikutnya, akan dilihat performa model.

```
accuracy(mod_2)
```

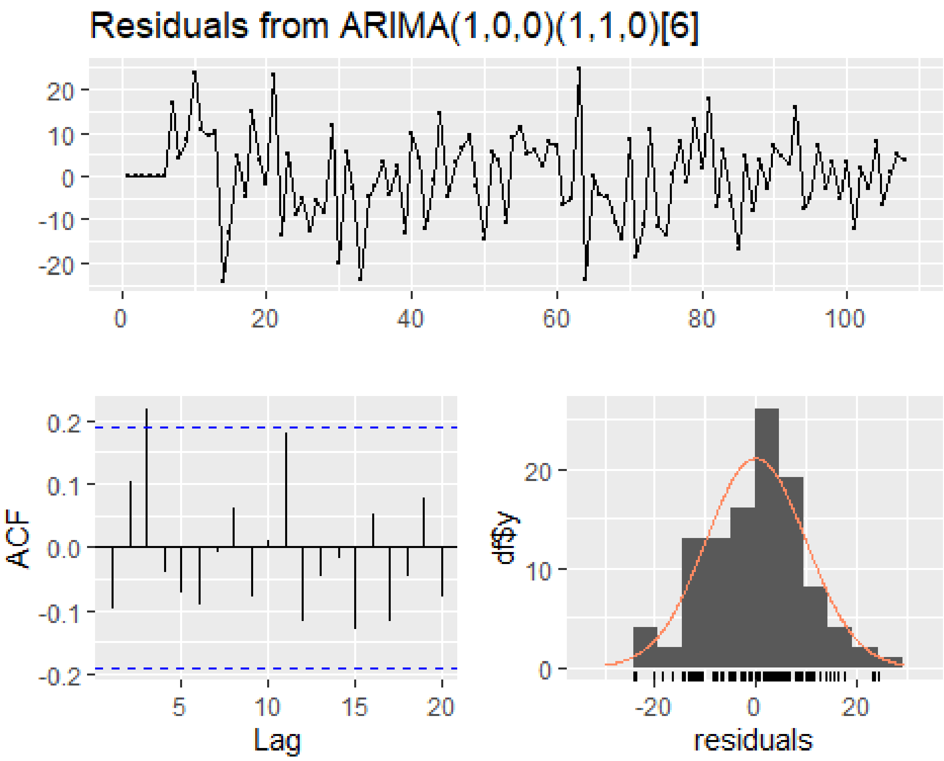
```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.02701485 9.902168 7.840702 -1.742368 12.70888 0.8308288
```

```
##                        ACF1
## Training set -0.09641638
```

Dapat dilihat bahwa RMSE dari model sebesar 9.9022 yang sangat kecil dibandingkan dengan standar deviasi data (= 15.402). Jadi, dapat disimpulkan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

5. Uji Diagnostik

```
checkresiduals(mod_2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[6]
## Q* = 10.514, df = 8, p-value = 0.2308
##
## Model df: 2.   Total lags used: 10
```

Perhatikan bahwa p-value = 0.2308 > $\alpha = 0.05$ sehingga dapat disimpulkan bahwa data saling bebas. Hal ini juga diperkuat dari plot ACF yang tidak signifikan. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan plot residual terlihat memiliki rataan nol dan variansi yang konstan.

```
data_residuals <- residuals(mod_2)
ks.test(residuals(mod_2), "pnorm", mean=mean(data_residuals), sd=sd(data_re
siduals))

##
##  One-sample Kolmogorov-Smirnov test
##
```

```
## data:  residuals(mod_2)
## D = 0.058803, p-value = 0.8493
## alternative hypothesis: two-sided
```

Uji Kolmogorov-Smirnov dengan hipotesis:
H_0: data berdistribusi normal
H_1: data tidak berdistribusi normal
Diperoleh p-value = 0.8493 > $\alpha = 0.05$ maka H_0 tidak ditolak dan dapat disimpulkan bahwa residual model berdistribusi normal.

```
ArchTest(data_residuals)

##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  data_residuals
## Chi-squared = 10.678, df = 12, p-value = 0.5567
```

Karena p-value = 0.5567 > $\alpha = 0.05$, dapat disimpulkan bahwa data tidak perlu dimodelkan dengan model heteroskedastik. Hal ini didukung pula oleh variansi galat yang terlihat konstan pada plot galat.

5.1 Forecasting

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

```
validation <- forecast(pm10_train,model=mod_2,h=length(pm10_validation))
actual <- as.vector(pm10_validation)
rmse_validation <- sqrt(mean((as.vector(validation$mean)-actual)^2))
rmse_validation

## [1] 8.600851
```

Dapat dilihat bahwa RMSE dari model ≈ 8.6008 yang lebih kecil dibandingkan dengan RMSE yang diperoleh sebelumnya. Dengan demikian, dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memprediksi data.

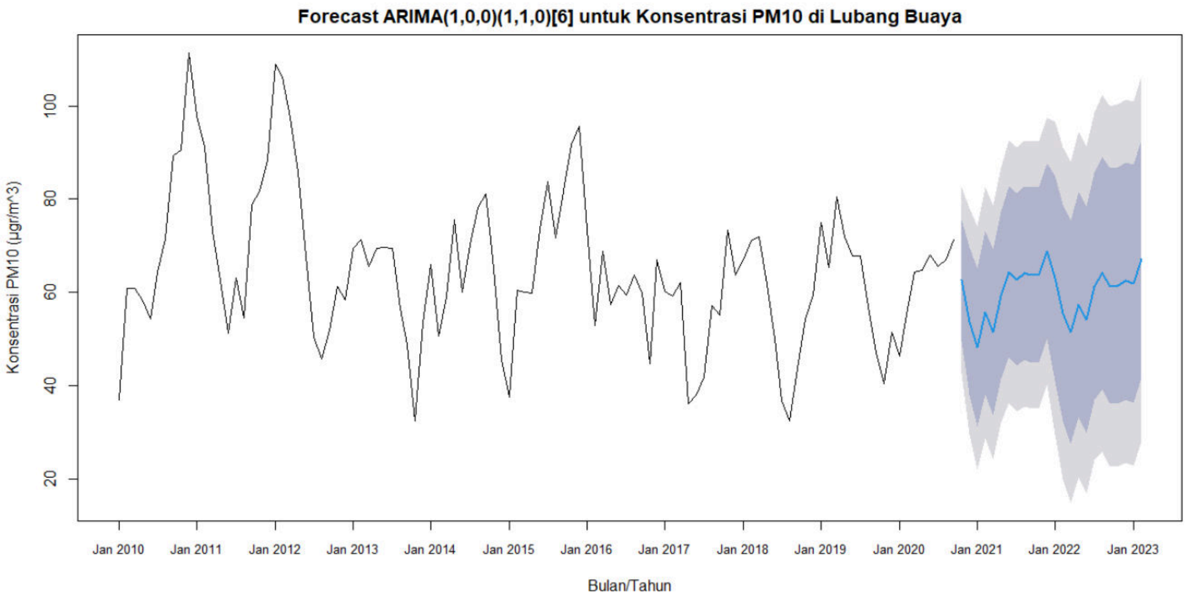
```
ape_validation <- abs((as.vector(validation$mean) -actual)/actual)*100
mape_validation <- mean(ape_validation)
mape_validation

## [1] 12.91173
```

Diperoleh juga MAPE sebesar ≈ 12.912

```
fc <- forecast(pm10,model=mod_2, h=24)
summary(fc)

##
## Forecast method: ARIMA(1,0,0)(1,1,0)[6]
##
## Model Information:
## Series: object
## ARIMA(1,0,0)(1,1,0)[6]
```

```
##
## Coefficients:
##          ar1      sar1
##       0.6886   -0.7936
## s.e.  0.0000    0.0000
##
## sigma^2 estimated as 103.8:  log likelihood=-471.14
## AIC=944.27   AICc=944.31   BIC=947.11
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.1590777 9.693121 7.599728 -2.147416 12.76708 0.8436042
##                    ACF1
## Training set -0.1081916
##
## Forecasts:
##     Point Forecast    Lo 80    Hi 80     Lo 95     Hi 95
## 133       53.47222 40.41419 66.53026 33.501686  73.44276
## 134       36.63251 20.77811 52.48691 12.385303  60.87972
## 135       47.44305 30.42255 64.46355 21.412446  73.47366
## 136       39.81651 22.27016 57.36286 12.981684  66.65134
## 137       49.51791 31.72766 67.30817 22.310068  76.72576
## 138       56.52531 38.62056 74.43005 29.142364  83.90825
## 139       56.34496 37.97978 74.71013 28.257847  84.43207
## 140       56.40971 37.83021 74.98922 27.994813  84.82462
## 141       64.47890 45.79862 83.15917 35.909884  93.04791
## 142       54.94791 36.22004 73.67577 26.306113  83.58970
## 143       54.71468 35.96429 73.46506 26.038438  83.39092
## 144       61.90939 43.14834 80.67045 33.216837  90.60195
## 145       54.04275 32.11294 75.97257 20.503992  87.58151
## 146       40.69821 17.41614 63.98027  5.091362  76.30505
## 147       50.94784 27.05133 74.84436 14.401277  87.49441
## 148       42.93161 18.74920 67.11401  5.947809  79.91540
## 149       50.58526 26.26847 74.90205 13.395945  87.77458
## 150       57.63287 33.25262 82.01311 20.346496  94.91923
## 151       55.86752 30.85190 80.88315 17.609426  94.12562
## 152       53.16601 27.85469 78.47733 14.455684  91.87634
## 153       61.68564 36.23531 87.13596 22.762719 100.60855
## 154       52.46757 26.95159 77.98354 13.444253  91.49088
## 155       53.86204 28.31500 79.40908 14.791210  92.93287
## 156       61.02657 35.46481 86.58832 21.933228 100.11990
```

```
plot(forecast(mod_2, h=24, level=c(80,95)), xaxt='n',ylab="Konsentrasi PM10
(µgr/m^3)", xlab="Bulan/Tahun", main="Forecast ARIMA(1,0,0)(1,1,0)[6] untuk
Konsentrasi PM10 di Lubang Buaya")
axis(1, at=seq(1, 140,10), labels=format(seq(as.Date("2010/1/1"), by = "yea
r", length.out = 14),format("%b %Y")), cex.axis=0.8, xpd=TRUE)
```


Forecast ARIMA(1,0,0)(1,1,0)[6] untuk Konsentrasi PM10 di Lubang Buaya
Konsentrasi PM10 (µgr/m^3)
20
40
60
80
100
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019
Jan 2020
Jan 2021
Jan 2022
Jan 2023
Bulan/Tahun