12-Percobaan-Simulasi.R

User

2022-07-04

```
# LIBRARY -----------------------------------------------------------------------------------

library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(zoo)
library(tsoutliers) #untuk outlier
```

```
## Warning: package 'tsoutliers' was built under R version 4.1.3
```

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

```
library(ggplot2) #Visualisasi data
library(forecast) #untuk replace outlier, akurasi (RMSE, MAPE, dll)
library(rangerts)
library(e1071) #untuk SVR
library(dplyr)
```

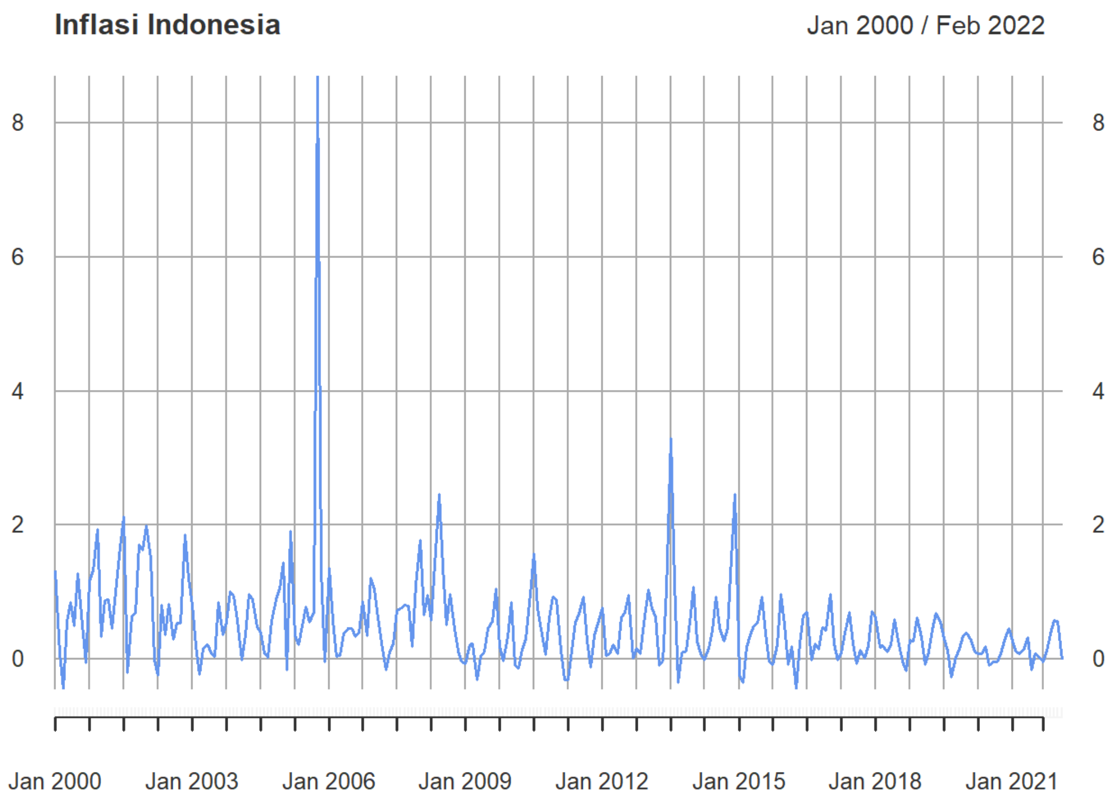
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

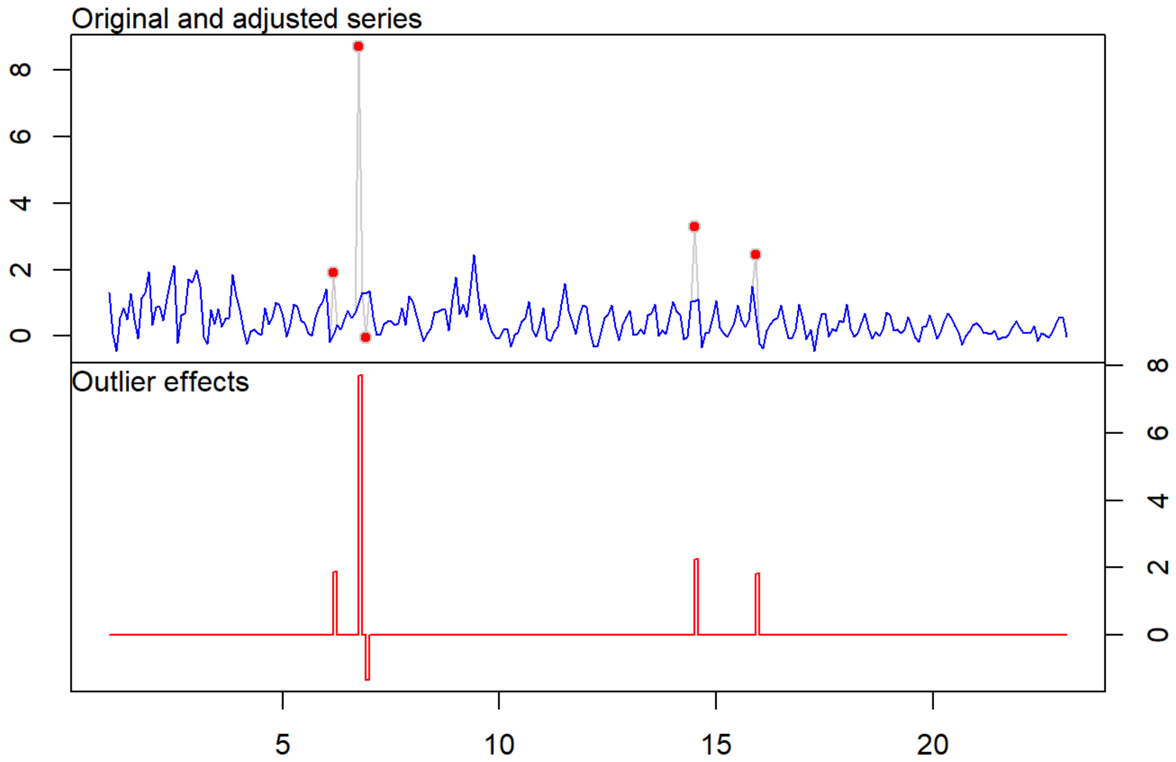
```
# devtools::install_github("hyanworkspace/rangerts", quiet = T)
# quiet = TRUE to mask c++ compilation messages, optional

# EKSPLORASI DATA ------------------------------------------------------------------------

#data inflasi
inflasi   = read.csv("https://raw.githubusercontent.com/hiasupriadi/tesis/main/inflasi.csv")
bulanan   = as.yearmon(2000 + seq(0, 265)/12) #idx bln dari th 2000, pjg data=266
Inf.Indo  = xts(inflasi$INDONESIA, order.by = bulanan) #konversi data ke format ts
plot.xts(Inf.Indo, main = "Inflasi Indonesia", col='#6495ED', lwd=1.5)
```




```
# IDENTIFIKASI OUTLIER -------------------------------------------------------------------

#outlier pada data inflasi Indonesia
out.Indo = tso(as.ts(Inf.Indo)) #cek outlier di data ke berapa
```

```
## Warning in locate.outliers.iloop(resid = resid, pars = pars, cval = cval, :
## stopped when 'maxit.iloop' was reached
```

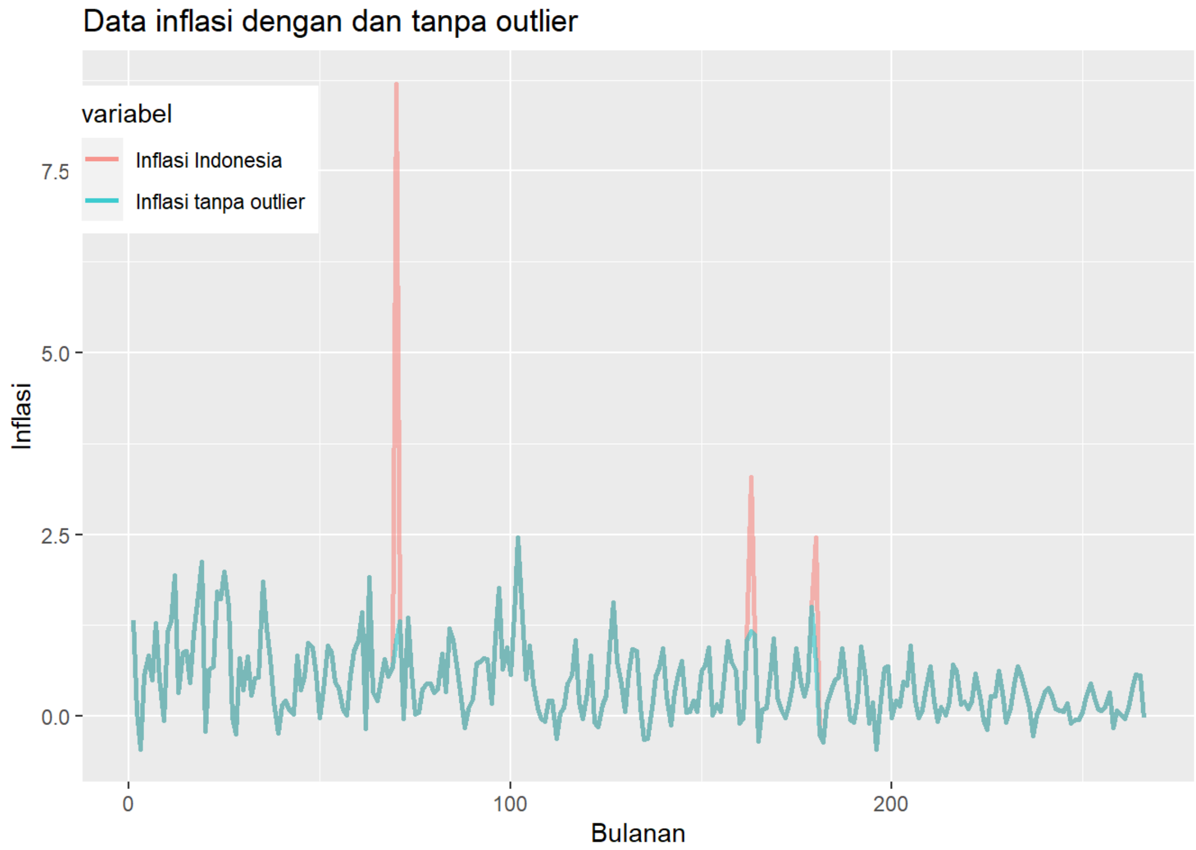
```
plot(out.Indo)
```




```
#data frame outlier Inflasi Indonesia
data.frame(out.Indo$outliers, Inf.Indo[out.Indo$outliers$ind])
```

```
##          type ind  time   coefhat     tstat Inf.Indo.out.Indo.outliers.ind.
## Mar 2005   AO  63  6:03  1.870230  5.213898                            1.91
## Oct 2005   AO  70  6:10  7.711132 21.446621                            8.70
## Dec 2005   AO  72  6:12 -1.330363 -3.696796                           -0.04
## Jul 2013   AO 163 14:07  2.242193  6.230923                            3.29
## Dec 2014   AO 180 15:12  1.814257  5.048074                            2.46
```

```
# MENGGANTI NILAI OUTLIER --------------------------------------------------------------------

Inf.Indo2 = tsclean(as.ts(Inf.Indo))
#tsclean --> Identify and replace outliers and missing values in a time series

#Visualisasi data inflasi dengan dan tanpa outlier

Gambar1  = data.frame(Bulanan=c(time(1:266)), Inf.Indo, Inf.Indo2)
ggplot(data = Gambar1, aes(x=Bulanan, y=value, color=variabel )  ) +
  ylab('Inflasi') +
  ggtitle('Data inflasi dengan dan tanpa outlier') +
  geom_line(aes(y=Inf.Indo , col="Inflasi Indonesia"), size=1, alpha=.5) +
  geom_line(aes(y=Inf.Indo2, col="Inflasi tanpa outlier"),  size=1, alpha=.5) +
  theme(legend.position=c(.1,.85))
```




```
#Outlier yang telah diganti
Rep.out = data.frame(Inf.Indo[out.Indo$outliers$ind], Inf.Indo2[out.Indo$outliers$ind])
names(Rep.out)[1] = "Outlier"
names(Rep.out)[2] = "Pengganti outlier"
Rep.out
```

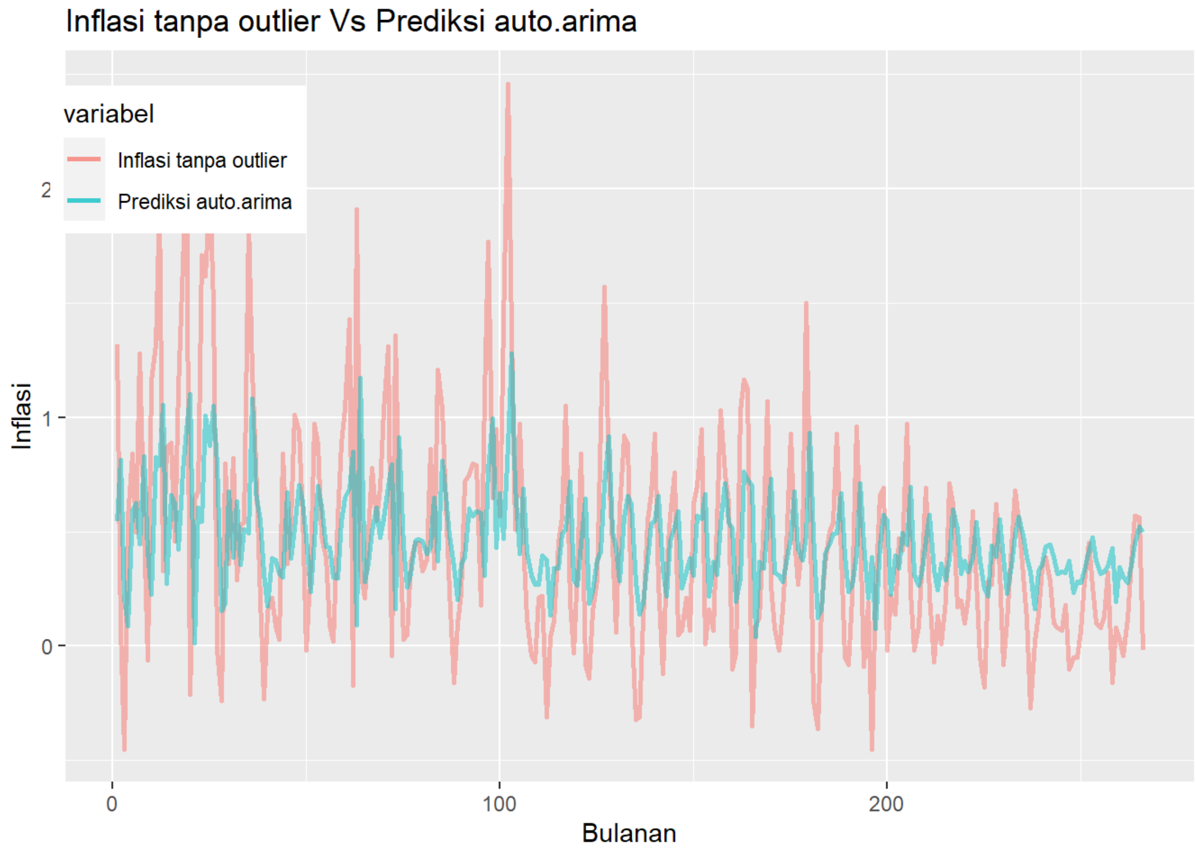
```
##          Outlier Pengganti outlier
## Mar 2005    1.91         1.9100000
## Oct 2005    8.70         1.0323272
## Dec 2005   -0.04        -0.0400000
## Jul 2013    3.29         1.1663341
## Dec 2014    2.46         0.7169995
```

```
# MENDAPATKAN MODEL ARIMA DARI SERIES YANG SUDAH CLEAN OUTLIER --------------------------------
# Menggunakan auto.arima

#model ARIMA
model.arima = auto.arima(Inf.Indo2, trace = T, seasonal = F,
                      max.d = 2,
                      max.D = 2,
                      start.p = 0,
                      start.q = 0,
                      start.P = 0,
                      start.Q = 0,
                      stationary = T)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(0,0,0)            with non-zero mean : 395.2086
##  ARIMA(0,0,0)            with non-zero mean : 395.2086
##  ARIMA(1,0,0)            with non-zero mean : 345.9569
##  ARIMA(0,0,1)            with non-zero mean : 352.0048
##  ARIMA(0,0,0)            with zero mean     : 556.0634
##  ARIMA(2,0,0)            with non-zero mean : 344.2975
##  ARIMA(3,0,0)            with non-zero mean : 344.4162
##  ARIMA(2,0,1)            with non-zero mean : 345.5851
##  ARIMA(1,0,1)            with non-zero mean : 345.9205
##  ARIMA(3,0,1)            with non-zero mean : 344.8857
##  ARIMA(2,0,0)            with zero mean     : 387.8526
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,0,0)            with non-zero mean : 348.0791
##
##  Best model: ARIMA(2,0,0)            with non-zero mean
```

```
#checkresiduals(model.arima)

#Perbandingan hasil auto.arima dengan data aktual

Gambar2  = data.frame(Bulanan=c(time(1:266)), Inf.Indo2, model.arima$fitted)
ggplot(data = Gambar2, aes(x=Bulanan, y=value, color=variabel )  ) +
  ylab('Inflasi') +
  ggtitle('Inflasi tanpa outlier Vs Prediksi auto.arima') +
  geom_line(aes(y=Inf.Indo2 , col="Inflasi tanpa outlier"), size=1, alpha=.5) +
  geom_line(aes(y=model.arima$fitted, col="Prediksi auto.arima"),  size=1, alpha=.5) +
  theme(legend.position=c(.1,.85))
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```
#Parameter ARIMA yang diperoleh dengan auto.arima
model.arima$coef
```

```
##         ar1         ar2   intercept
##  0.45021775 -0.08810694  0.46488075
```

```
mean(model.arima$residuals)
```

```
## [1] -0.001468713
```

```
#var(model.arima$residuals)
#mean(Inf.Indo2)
#var(Inf.Indo2)

#mu = intercept/(1-ar1-ar2)
#mu.teoritis = as.numeric(model.arima$coef[3]/(1-model.arima$coef[1]-model.arima$coef[2]))

# SIMULASI ------------------------------------------------------------------------------------

N                  = 50 #banyak data time series
iter               = 2
rata.Yt            = matrix(nrow=iter, ncol=N)

akurasi.autoarima2 = matrix(nrow=iter, ncol=5)
Pred.autoarima2Yt  = matrix(nrow=iter, ncol=N)

akurasi.SVR        = matrix(nrow=iter, ncol=5)
PredSVR.Yt         = matrix(nrow=iter, ncol=N)

akurasi.RF         = matrix(nrow=iter, ncol=5)
PredRF.Yt          = matrix(nrow=iter, ncol=N)

#iterasi
for(i in 1:iter){
  #Bangkitkan time series dari ARIMA(2,0,0)
  #parameter ARIMA(2,0,0)
  mu.teoritis = as.numeric(model.arima$coef[3]/(1-model.arima$coef[1]-model.arima$coef[2]))
  mu          = mu.teoritis
  ar1         = model.arima$coef[1]
  ar2         = model.arima$coef[2]
  var.res     = var(model.arima$residuals)
  at          = rnorm(N+50, 0, sqrt(var.res)) #0.2107741 adl varians dr res model arima
  Z           = matrix(nrow = N+50, ncol=1)
  Z[1]        = 0
  Z[2]        = 0

      for (j in seq(3, N+50))
        {
            Z[j] = mu + ar1*Z[j-1] + ar2*Z[j-2] + at[j] #ARIMA(2,0,0)
        }
      #Z adl series yang tidak mengandung outlier

  #Menambahkan Outlier AO di T=100
  mo100    = outliers("TC", 50+50)
  ao100    = outliers.effects(mo100, n=length(Z))

  Y        = Z+0.75*(max(Z)-min(Z))*(ao100)
  Yt       = as.vector(Y)
  Yt1      = lag(Yt, 1)
  Yt2      = lag(Yt, 2)
  #dmmy.out = as.vector(ao100)
  data     = data.frame(Yt, Yt1, Yt2)[(50+1):(N+50),]

  #.......................................................................
  #                                 Model ARIMA ----
  #.......................................................................

  #model.arimax             = arima(data$Yt, order = c(2,0,0), xreg=data$dmmy.out, #menggunakan auto.ari
ma
  #                                include.mean=T)

  model.autoarima2          = auto.arima(data$Yt, trace = F, seasonal = F,
```

```
                                   max.d = 2,
                                   max.D = 2,
                                   start.p = 0,
                                   start.q = 0,
                                   start.P = 0,
                                   start.Q = 0,
                                   stationary = T)

predict.autoarima2        = fitted(model.autoarima2)

akurasi.autoarima2[i,]    = accuracy(predict.autoarima2, data$Yt)
Pred.autoarima2Yt[i,]     = predict.autoarima2
rata.Yt[i,]               = data$Yt #nilai rata-rata Yt yang dibangkitkan

#..............................................................................
#                       Model Support Vector Regression ----
#..............................................................................

#Tune Parameter SVR -> Identifikasi parameter terbaik
tuneSVR1   = tune(svm, Yt ~ Yt1+Yt2,  data = data, kernel="radial",
                  ranges = list(epsilon = seq(0,1,0.5),
                                cost    = seq(5,100,20),
                                gamma   = seq(1,10,2),type="eps-regression"))

tuneSVR2   = tune(svm, Yt ~ Yt1+Yt2,  data = data, kernel="radial",
                  ranges = list(epsilon = seq(0,tuneSVR1$best.model$epsilon+.15,0.05),
                                cost    = seq(tuneSVR1$best.model$cost-4,
                                              tuneSVR1$best.model$cost+5,2),
                                gamma   = seq(tuneSVR1$best.model$gamma-1,
                                              tuneSVR1$best.model$gamma+1,0.2)))

model.SVR            = tuneSVR2$best.model
predict.SVR          = predict(model.SVR, data[,-4])

akurasi.SVR[i,]      = accuracy(predict.SVR,data$Yt)
PredSVR.Yt[i,]       = predict.SVR

#..............................................................................
#                           Model Random Forest ----
#..............................................................................

# Membangun Model RF
# Moving block bootstrap
# the moving mode with replacement
# thus the sample fraction is the default value = 1
hyper_gridRF = expand.grid(num.trees  = seq(50,200,50),
                           mtry       = 1:2,
                           node_size  = 1:3,
                           block.size = 2:3,
                           OOB_RMSE   = 0)

for(k in 1:nrow(hyper_gridRF)) {
  tune.RF = rangerts::rangerts(Yt~Yt1+Yt2,
                               data          = data,
                               num.trees     = hyper_gridRF$num.trees[k],
                               mtry          = hyper_gridRF$mtry[k],
                               importance    = 'impurity',
                               min.node.size = hyper_gridRF$node_size[k],
                               #case.weights = NULL,
                               replace       = T,
                               #seed          = 1,
                               bootstrap.ts  = "moving",
```

```
                               block.size    = hyper_gridRF$block.size[k])

    hyper_gridRF$OOB_RMSE[k] = sqrt(tune.RF$prediction.error)
  }

  position = which.min(hyper_gridRF$OOB_RMSE)
  head(hyper_gridRF[order(hyper_gridRF$OOB_RMSE),],5)

  # fit best model
  model.RF <- rangerts::rangerts(Yt~Yt1+Yt2,
                               data          = data,
                               num.trees     = hyper_gridRF$num.trees[position],
                               mtry          = hyper_gridRF$mtry[position],
                               importance    = 'impurity',
                               min.node.size = hyper_gridRF$node_size[position],
                               replace       = T,
                               #seed          = 1,
                               bootstrap.ts  = "moving",
                               block.size    = hyper_gridRF$block.size[position])
  #model.RF$predictions
  predict.RF          = model.RF$predictions

  akurasi.RF[i,]      = accuracy(predict.RF,data$Yt)
  PredRF.Yt[i,]       = predict.RF
  }

# AKURASI ---------------------------------------------------------------------------------------

cat("mu teoritis = ", mu, "\n")
```

```
## mu teoritis =  0.7287798
```

```
#Rata-rata akurasi Model ARIMA
acc.colname                  = colnames(accuracy(predict.autoarima2, data$Yt))

colnames(akurasi.autoarima2) = acc.colname
mean.akurasiARIMA            = colMeans(akurasi.autoarima2); mean.akurasiARIMA
```

```
##            ME          RMSE           MAE           MPE          MAPE
##   0.003241027   0.517557160   0.400513383 -18.678475383  76.157139348
```

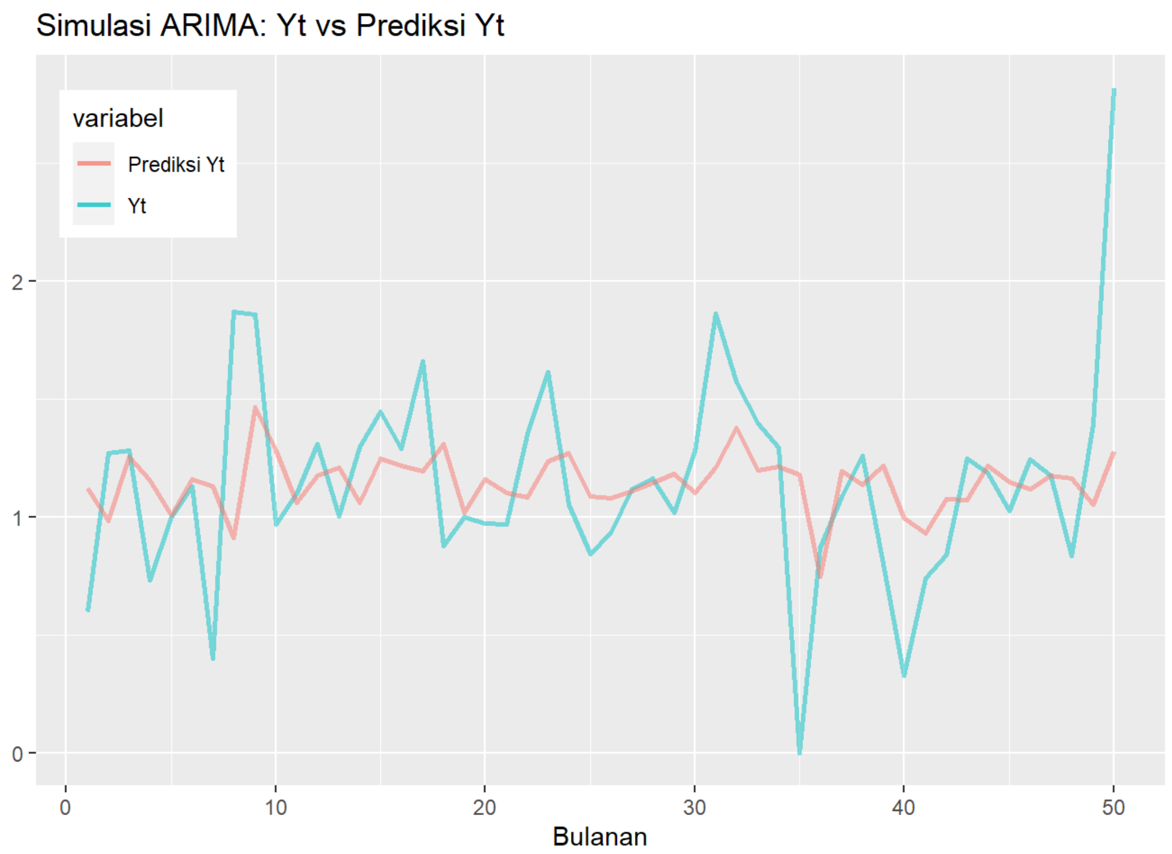
```
#Model SVR
colnames(akurasi.SVR)        = acc.colname
mean.akurasiSVR              = colMeans(akurasi.SVR); mean.akurasiSVR
```

```
##          ME        RMSE         MAE         MPE        MAPE
##   0.0144587   0.4829327   0.3940670 -20.8419764  64.8955918
```

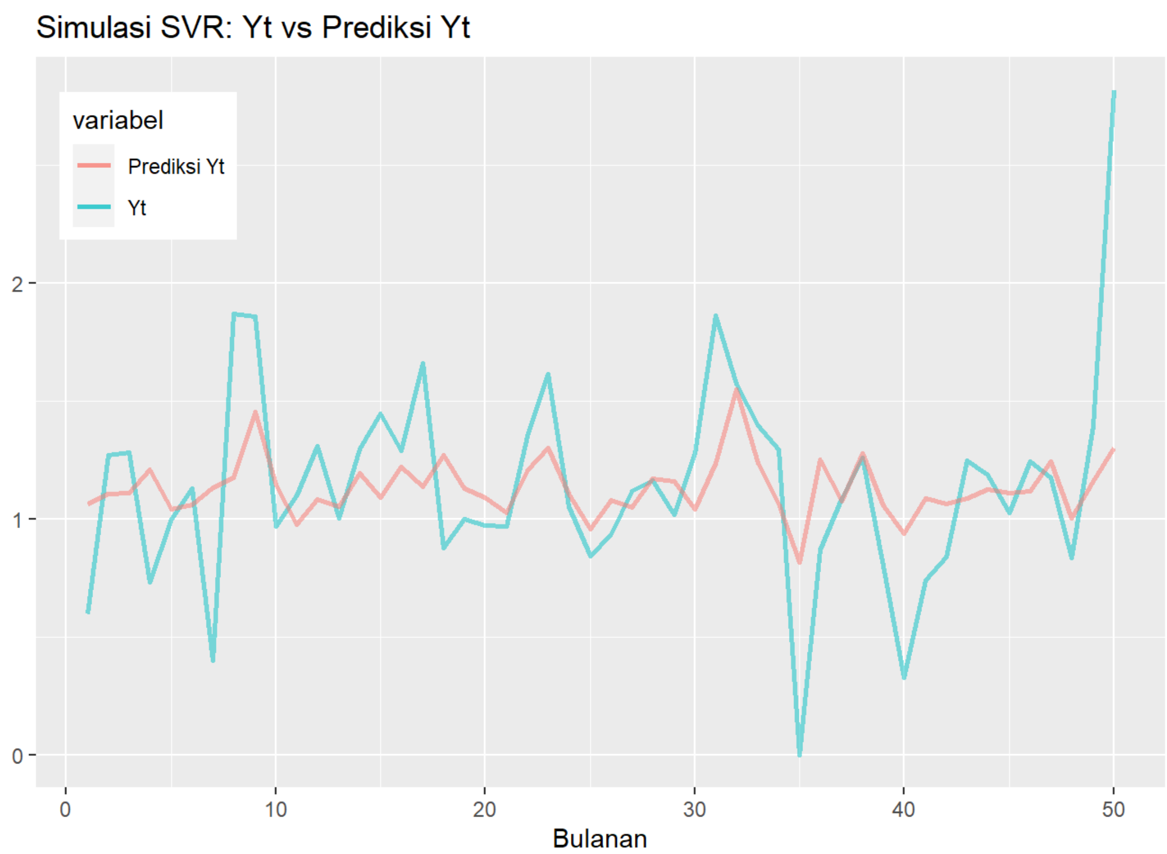
```
#Model RF
colnames(akurasi.RF)         = acc.colname
mean.akurasiRF               = colMeans(akurasi.RF); mean.akurasiRF
```

```
##           ME         RMSE          MAE          MPE         MAPE
##   0.01001528   0.58371694   0.46335320 -15.20801308  78.67300410
```

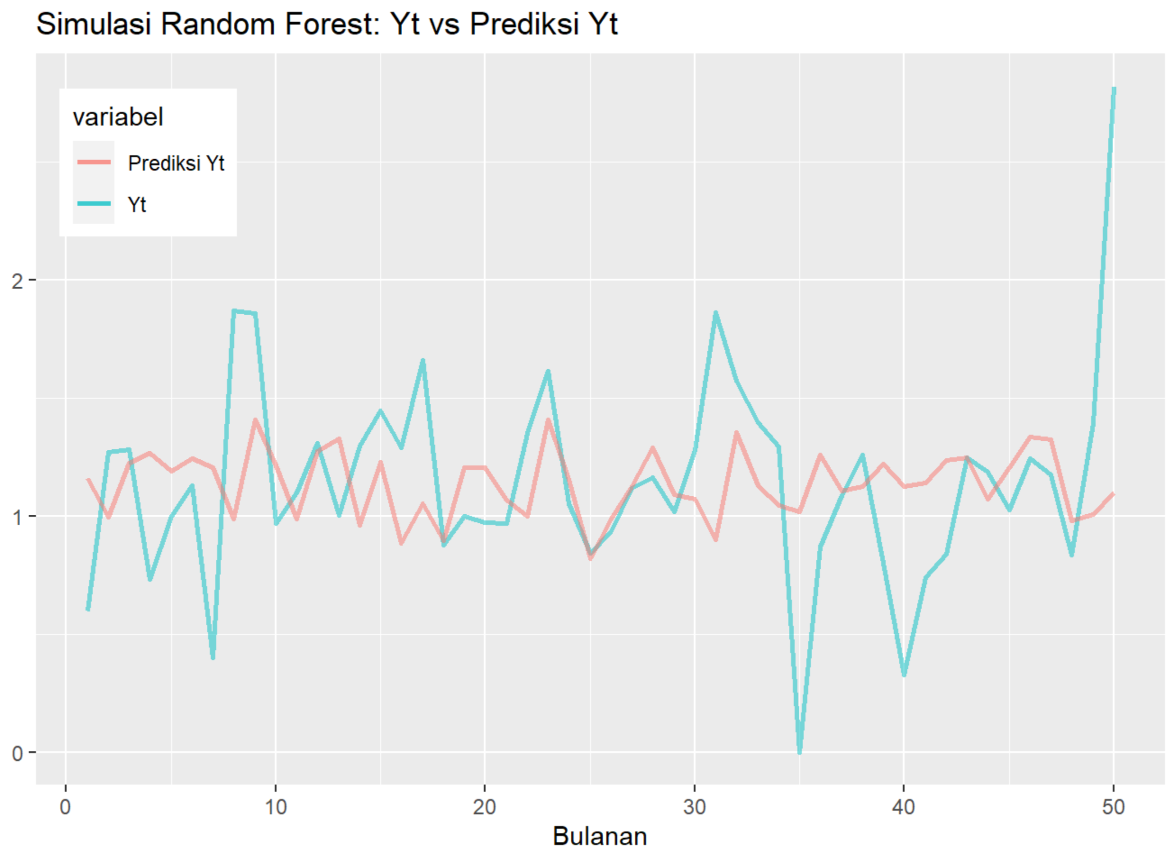
```
# VISUALISASI DATA Yt Vs PREDIKSI Yt ----------------------------------------------------------
#.............................................................................................
#Visualisasi data aktual dan prediksi untuk ARIMA
#rata-rata prediksi dengan rata-rata data aktual
rataPred.autoarima2Yt        = colMeans(Pred.autoarima2Yt)
rata.Yt                      = colMeans(rata.Yt)

#ggplot()+
# geom_line(aes(x=c(1:length(rataPred.autoarima2Yt)), y=rata.Yt), colour='blue', size=.5)+
# geom_line(aes(x=c(1:length(rataPred.autoarima2Yt)), y=rataPred.autoarima2Yt), colour='goldenrod2')+
# ggtitle('Simulasi ARIMA - Yt vs Prediksi Yt')+
# xlab('Time')+
#  ylab('Data')


Gambar3  = data.frame(Bulanan=c(time(1:N)), rata.Yt, rataPred.autoarima2Yt)
ggplot(data = Gambar3, aes(x=Bulanan, y=value, color=variabel )  ) +
  ylab('') +
  ggtitle('Simulasi ARIMA: Yt vs Prediksi Yt') +
  geom_line(aes(y=rata.Yt , col="Yt"), size=1, alpha=.5) +
  geom_line(aes(y=rataPred.autoarima2Yt, col="Prediksi Yt"),  size=1, alpha=.5) +
  theme(legend.position=c(.1,.85))
```



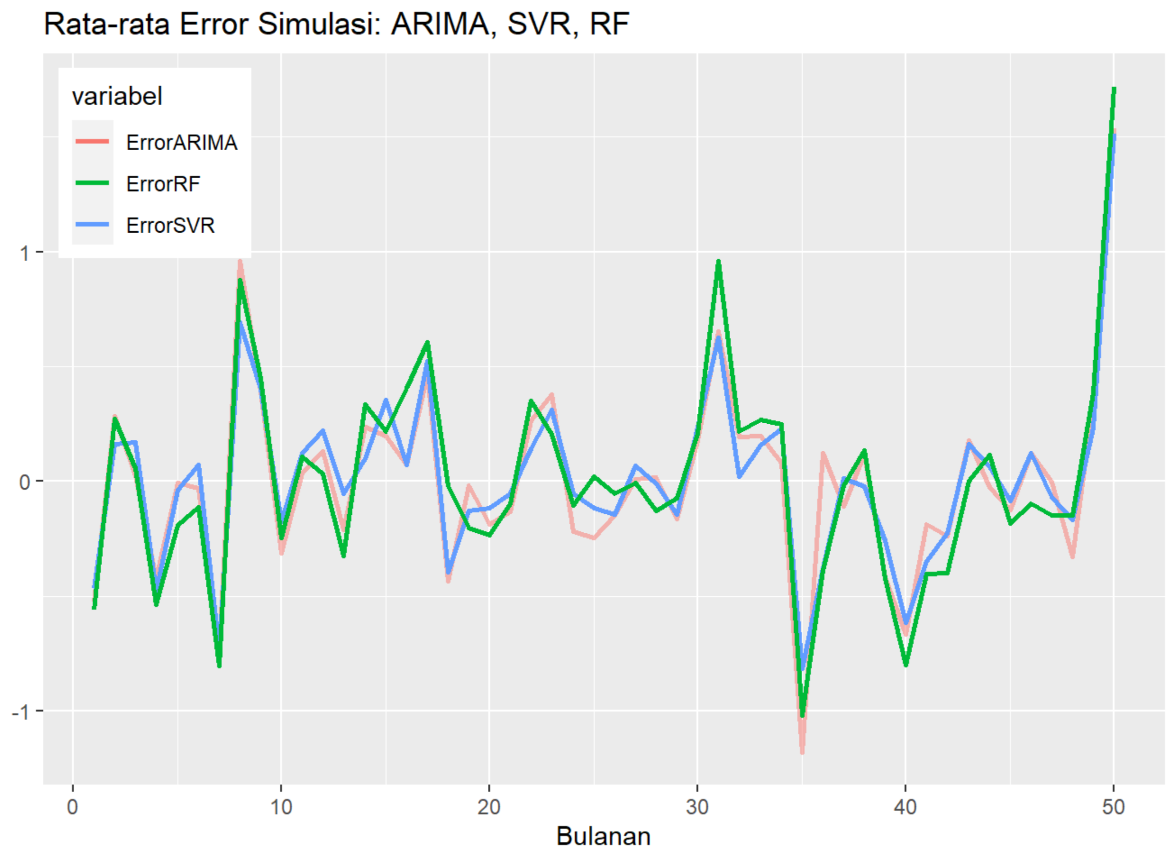
```
#.......................................................................................
#Visualisasi data aktual dan prediksi untuk SVR
#rata-rata prediksi dengan rata-rata data aktual
rataPredSVR.Yt        = colMeans(PredSVR.Yt)

#ggplot()+
#  geom_line(aes(x=c(1:length(rataPredSVR.Yt)), y=rata.Yt), colour='blue', size=.5)+
#  geom_line(aes(x=c(1:length(rataPredSVR.Yt)), y=rataPredSVR.Yt), colour='goldenrod2')+
#  ggtitle('Simulasi SVR - Yt vs Prediksi Yt')+
#  xlab('Time')+
#  ylab('Data')


Gambar4  = data.frame(Bulanan=c(time(1:N)), rata.Yt, rataPredSVR.Yt)
ggplot(data = Gambar4, aes(x=Bulanan, y=value, color=variabel )  ) +
  ylab('') +
  ggtitle('Simulasi SVR: Yt vs Prediksi Yt') +
  geom_line(aes(y=rata.Yt , col="Yt"), size=1, alpha=.5) +
  geom_line(aes(y=rataPredSVR.Yt, col="Prediksi Yt"),  size=1, alpha=.5) +
  theme(legend.position=c(.1,.85))
```



```
#..............................................................................................
#Visualisasi data aktual dan prediksi untuk SVR
#rata-rata prediksi dengan rata-rata data aktual
rataPredRF.Yt       = colMeans(PredRF.Yt)

#ggplot()+
#  geom_line(aes(x=c(1:length(rataPredRF.Yt)), y=rata.Yt), colour='blue', size=.5)+
#  geom_line(aes(x=c(1:length(rataPredRF.Yt)), y=rataPredRF.Yt), colour='goldenrod2')+
#  ggtitle('Simulasi RF - Yt vs Prediksi Yt')+
#  xlab('Time')+
#  ylab('Data')

Gambar5  = data.frame(Bulanan=c(time(1:N)), rata.Yt, rataPredRF.Yt)
ggplot(data = Gambar5, aes(x=Bulanan, y=value, color=variabel )  ) +
  ylab('') +
  ggtitle('Simulasi Random Forest: Yt vs Prediksi Yt') +
  geom_line(aes(y=rata.Yt , col="Yt"), size=1, alpha=.5) +
  geom_line(aes(y=rataPredRF.Yt, col="Prediksi Yt"),  size=1, alpha=.5) +
  theme(legend.position=c(.1,.85))
```



```
#..............................................................................................

# VISUALISASI RATA-RATA ERROR -----------------------------------------------------------------

ErrorARIMA    = rata.Yt - rataPred.autoarima2Yt
ErrorSVR      = rata.Yt - rataPredSVR.Yt
ErrorRF       = rata.Yt - rataPredRF.Yt

#ggplot()+
#  geom_line(aes(x=c(1:length(rata.Yt)), y=ErrorARIMA), colour='black', size=.5)+
#  geom_line(aes(x=c(1:length(rata.Yt)), y=ErrorSVR), colour='blue', size=.5)+
#  geom_line(aes(x=c(1:length(rata.Yt)), y=ErrorRF), colour='red')+
#  ggtitle('Rata-rata Error ARIMA, SVR, RF')+
#  xlab('Time')+
#  ylab('Data')


Gambar6  = data.frame(Bulanan=c(time(1:N)), ErrorARIMA, ErrorSVR, ErrorRF)
ggplot(data = Gambar6, aes(x=Bulanan, y=value, color=variabel )  ) +
  ylab('') +
  ggtitle('Rata-rata Error Simulasi: ARIMA, SVR, RF') +
  geom_line(aes(y=ErrorARIMA , col="ErrorARIMA"), size=1, alpha=.5) +
  geom_line(aes(y=ErrorSVR , col="ErrorSVR"), size=1, alpha=1) +
  geom_line(aes(y=ErrorRF , col="ErrorRF"), size=1, alpha=2) +
  theme(legend.position=c(.1,.85))
```




```
# SELESAI --------------------------------------------------------------------------------------
```