Pemulusan Harga Adjusted Closing IHSG Periode 1 Januari 2017 - 31 Maret 2022

Kelompok 14 - MPDW Genap 21/22

Anggota Kelompok

- G14190007 Dhiffa Fatihah Umami
- G14190026 Nenden Maulidia
- G14190043 Tony Alfan
- G14190047 Alvin Christian
- G14190071 Muhammad Hadyan Rizki

Import Library dan Dataset

Mengimport libraries yang akan digunakan

```
library(imputeTS)
library(knitr)
library(quantmod)
library(forecast)
```

Import data dan melihat 10 data pertama.

```
start <- as.POSIXct("2017-01-01")
end <- as.POSIXct("2022-03-31")
getSymbols(Symbols = "^JKSE",src = "yahoo", from = start, to = end)
```

```
## [1] "^JKSE"
```

```
data <- as.data.frame(JKSE)
data$Date <- as.Date(rownames(data))
fulldates <- data.frame(Date=seq(as.Date("2017-01-01"), as.Date("2022-03-31"), by="days"))
ihsg <- merge(fulldates,data,by="Date",all.x=T)

kable(head(ihsg,10), caption="First 10 rows of IHSG daily prices")
```

Table 1: First 10 rows of IHSG daily prices

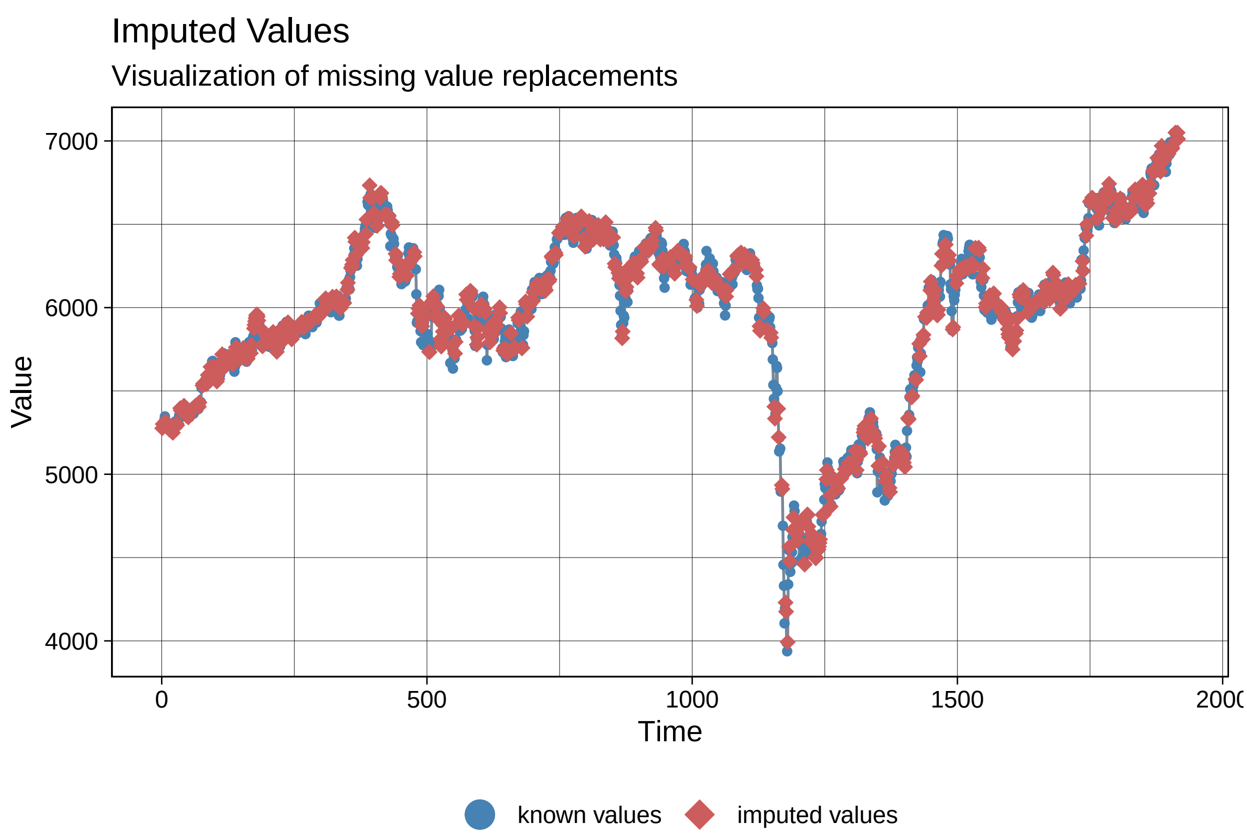
Date	JKSE.Open	JKSE.High	JKSE.Low	JKSE.Close	JKSE.Volume	JKSE.Adjusted
2017-01-01	NA	NA	NA	NA	NA	NA
2017-01-02	NA	NA	NA	NA	NA	NA
2017-01-03	5290.388	5292.181	5246.852	5275.971	33217000	5275.971
2017-01-04	5270.016	5312.956	5249.234	5301.183	53111300	5301.183

Date	JKSE.Open	JKSE.High	JKSE.Low	JKSE.Close	JKSE.Volume	JKSE.Adjusted
2017-01-05	5302.633	5328.490	5302.633	5325.504	77219400	5325.504
2017-01-06	5321.745	5350.245	5318.385	5347.022	71615300	5347.022
2017-01-07	NA	NA	NA	NA	NA	NA
2017-01-08	NA	NA	NA	NA	NA	NA
2017-01-09	5350.880	5360.061	5307.585	5316.364	97010200	5316.364
2017-01-10	5330.015	5331.133	5292.063	5309.924	70792500	5309.924

Dari 10 data pertama, dapat dilihat bahwa terdapat beberapa missing values. Pasar Saham tidak dibuka pada hari sabtu-minggu dan tanggal merah, sehingga dapat dipastikan akan terdapat missing value tiap minggunya. Maka akan dilakukan interpolasi spline untuk mengatasi missing value tersebut.

Imputasi Missing Values

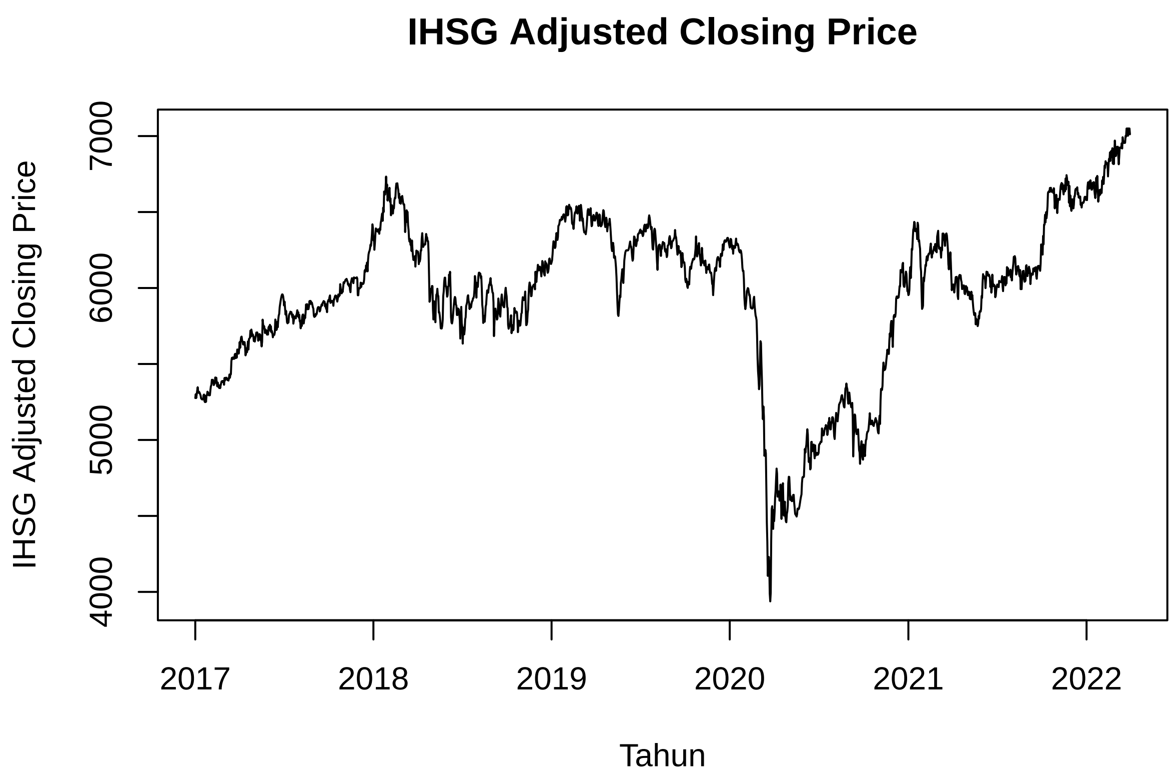
```
ihsg.imputed <- na_interpolation(ihsg, option="spline")
ggplot_na_imputations(ihsg$JKSE.Adjusted, ihsg.imputed$JKSE.Adjusted)
```



Dapat kita lihat bahwa imputasi tidak mengubah tren dari harga Adjusted Closing Price. Mari kita coba amati plot time series dari Adjusted Closing Price harian IHSG.

Eksplorasi Plot Deret Waktu

```
ihsg.ts <- ts(ihsg.imputed$JKSE.Adjusted)
plot(ihsg.imputed$JKSE.Adjusted~ihsg.imputed$Date,
     type = "l", xlab = "Tahun", main = "IHSG Adjusted Closing Price",
     ylab = "IHSG Adjusted Closing Price")
```



Train-test split data

Membagi dataset menjadi data training untuk melatih model dan data testing untuk validasi performa model.

```
ihsg.ts <- ts(ihsg.imputed$JKSE.Adjusted)
ihsg.seasonal <- ts(ihsg.imputed$JKSE.Adjusted, frequency=28)

train.prop <- floor(nrow(ihsg.imputed)*0.8)
test.prop <- nrow(ihsg.imputed)-train.prop

ts.train <- head(ihsg.ts, train.prop)
ts.test <- tail(ihsg.ts, test.prop)
seasonal.train <- head(ihsg.seasonal, train.prop)
seasonal.test <- tail(ihsg.seasonal, test.prop)
```

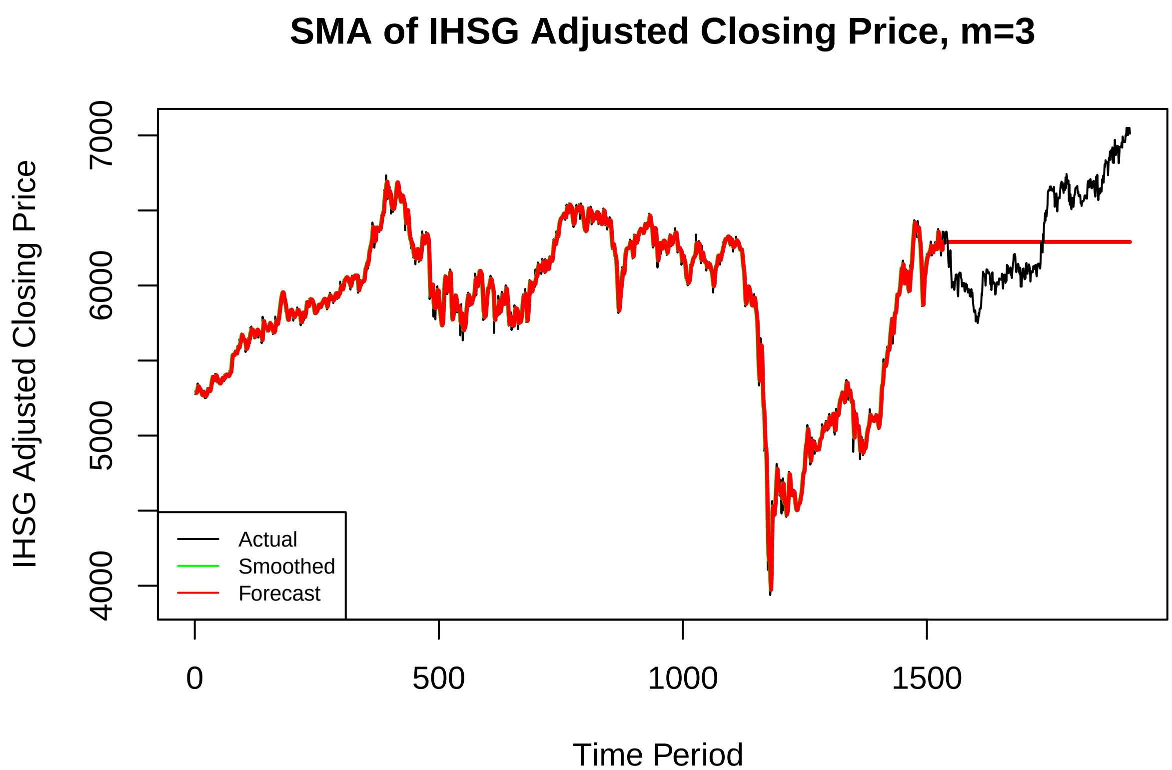
Single Moving Average

Menetapkan parameter m = 3 untuk moving average.

```
m = 3
```

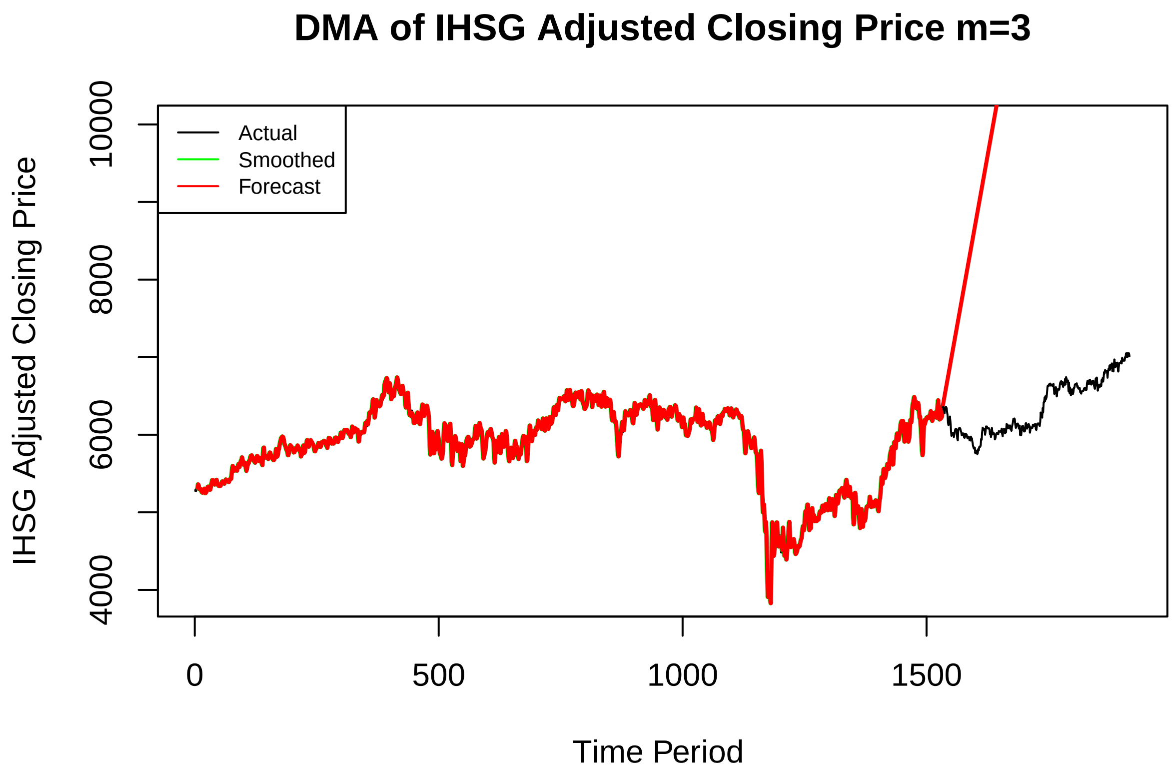
```
data.sma <- SMA(ts.train, n=m)
data.fc <- c(NA,data.sma)
data.gab <- data.frame(cbind(actual=c(ts.train,rep(NA,test.prop)),smoothing=c(data.sma,rep(NA,test.prop
                             forecast=c(data.fc,rep(data.fc[length(data.fc)],test.prop-1))))
```

```
ts.plot(data.gab[,1], xlab="Time Period ", ylab="IHSG Adjusted Closing Price",
        main= "SMA of IHSG Adjusted Closing Price, m=3", ylim=c(3900,7050))
lines(data.gab[,2],col="green",lwd=2)
lines(data.gab[,3],col="red",lwd=2)
lines(ts.test)
legend("bottomleft",c("Actual","Smoothed","Forecast"), lty=1,
       col=c("black","green","red"), cex=0.7)
```



Double Moving Average

```r
dma <- SMA(data.sma, n = m)
At <- 2*data.sma - dma
Bt <- 2/(m-1)*(data.sma - dma)
data.dma<- At+Bt
data.fc2<- c(NA, data.dma)

t = 1:test.prop+1
f = c()

for (i in t) {
  f[i] = At[length(At)] + Bt[length(Bt)]*(i)
}

data.gab2 <- data.frame(cbind(aktual = c(ts.train,rep(NA,test.prop+1)),
                              pemulusan1 = c(data.sma,rep(NA,test.prop+1)),
                              pemulusan2 = c(data.dma, rep(NA,test.prop+1)),
                              At = c(At, rep(NA,test.prop+1)),
                              Bt = c(Bt,rep(NA,test.prop+1)),
                              forecast = c(data.fc2, f[-1])))

ts.plot(data.gab2[,1], xlab="Time Period ", ylab="IHSG Adjusted Closing Price",
        main= "DMA of IHSG Adjusted Closing Price m=3", ylim=c(3900,10000))
lines(data.gab2[,3],col="green",lwd=2)
lines(data.gab2[,6],col="red",lwd=2)
lines(ts.test)
legend("topleft",c("Actual","Smoothed","Forecast"), lty=1,
       col=c("black","green","red"), cex=0.7)
```

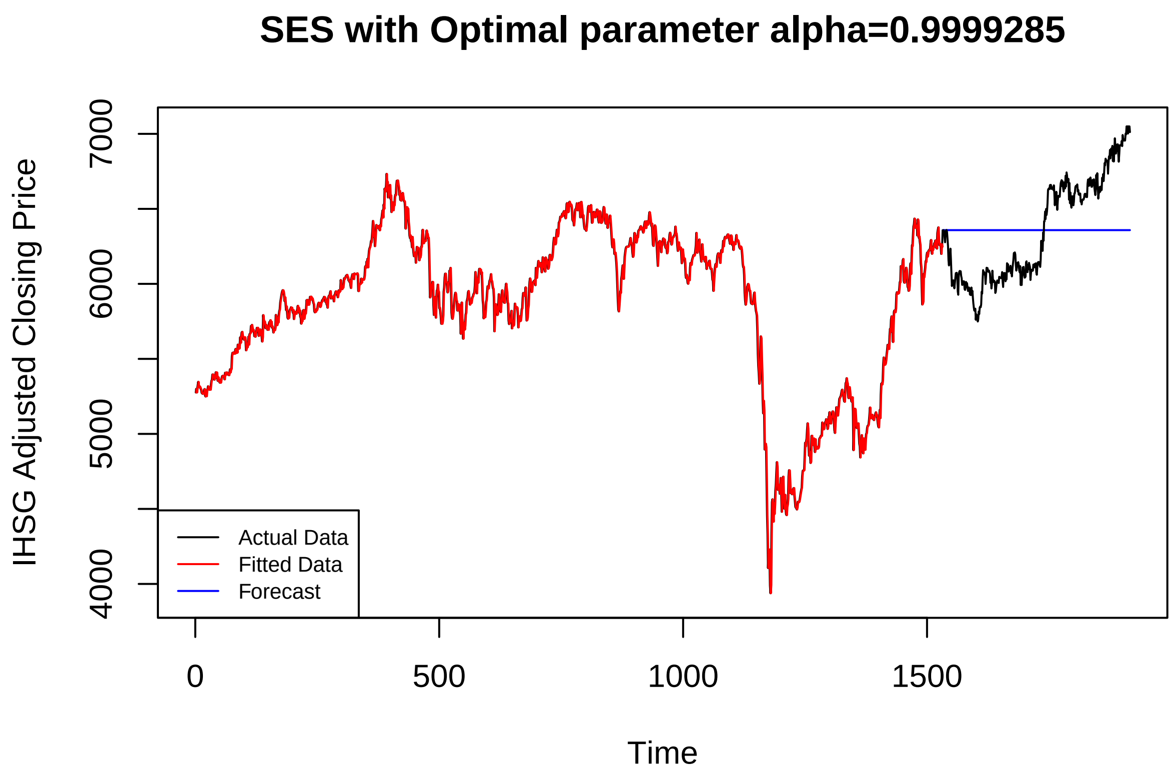


Single Exponential Smoothing

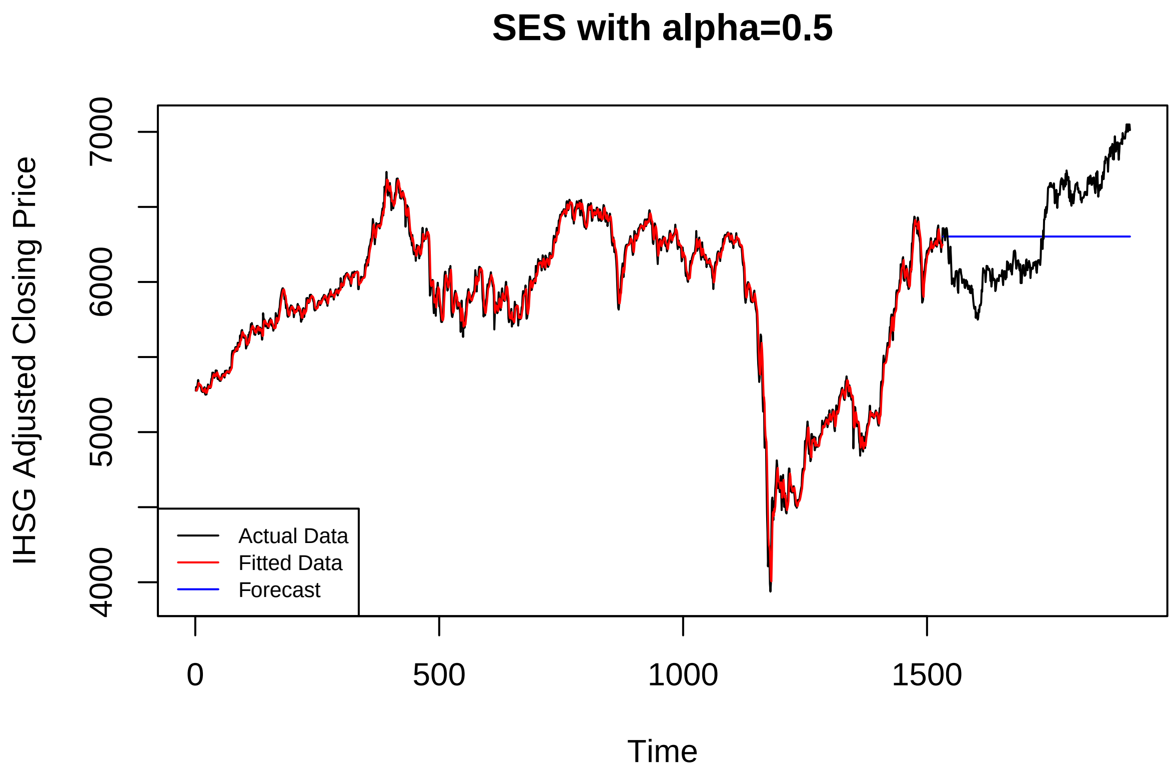
```
ses.1 <- HoltWinters(ts.train, gamma = F, beta = F, alpha = 0.5)
ses.2 <- HoltWinters(ts.train, gamma = F, beta = F, alpha = 0.9)
ses.opt <- HoltWinters(ts.train, gamma = F, beta = F)

ses.opt #optimum parameter for ses a = 0.9999414
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = ts.train, beta = F, gamma = F)
##
## Smoothing parameters:
##  alpha: 0.9999414
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 6358.203
```

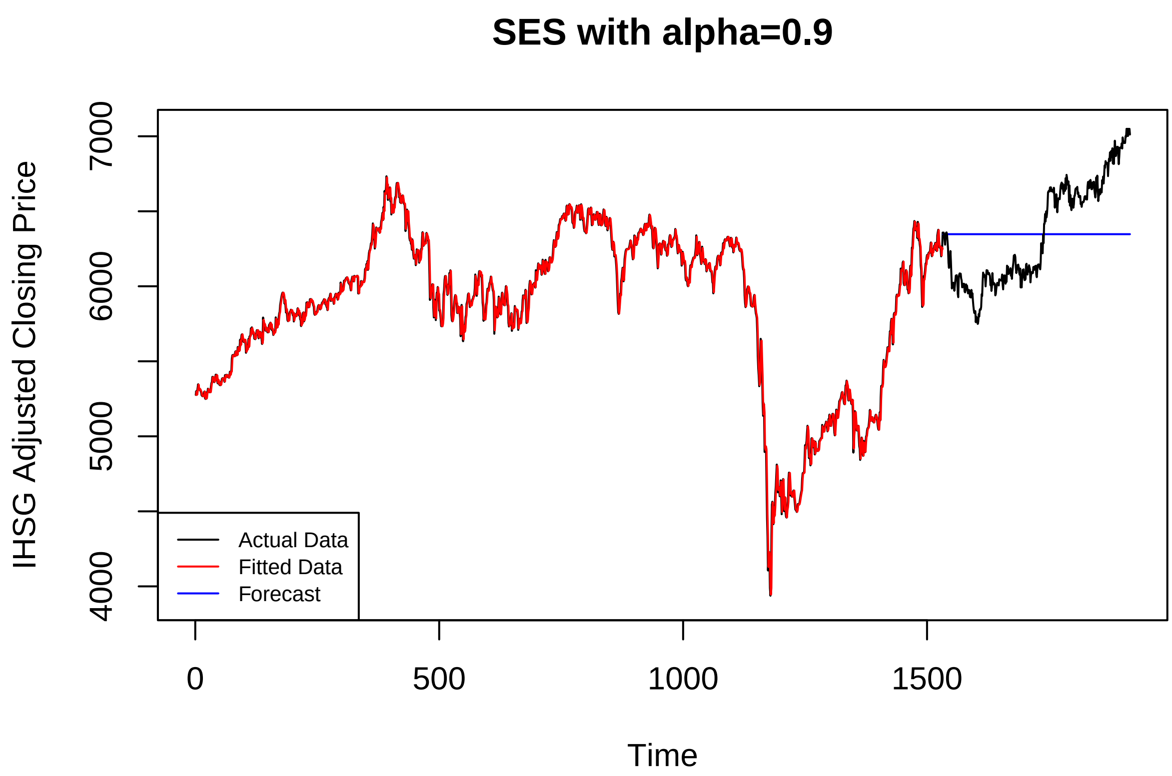
```
fc.ses1 <- predict(ses.1, n.ahead = test.prop)
fc.ses2 <- predict(ses.2, n.ahead = test.prop)
fc.sesopt <- predict(ses.opt, n.ahead = test.prop)

plot(ts.train,main="SES with Optimal parameter alpha=0.9999285",type="l",col="black",pch=12,
     ylab="IHSG Adjusted Closing Price", xlim=c(0,1916), ylim=c(3900,7050))
lines(ses.opt$fitted[,2],type="l",col="red")
lines(fc.sesopt,type="l",col="blue")
lines(ts.test,type="l")
legend("bottomleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```




```
plot(ts.train,main="SES with alpha=0.5",type="l",col="black",pch=12,
     ylab="IHSG Adjusted Closing Price", xlim=c(0,1916), ylim=c(3900,7050))
lines(ses.1$fitted[,2],type="l",col="red")
lines(fc.ses1,type="l",col="blue")
lines(ts.test,type="l")
legend("bottomleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```




```
plot(ts.train,main="SES with alpha=0.9",type="l",col="black",pch=12,
     ylab="IHSG Adjusted Closing Price",
     xlim=c(0,1916),ylim=c(3900,7050))
lines(ses.2$fitted[,2],type="l",col="red")
lines(fc.ses2,type="l",col="blue")
lines(ts.test,type="l")
legend("bottomleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```

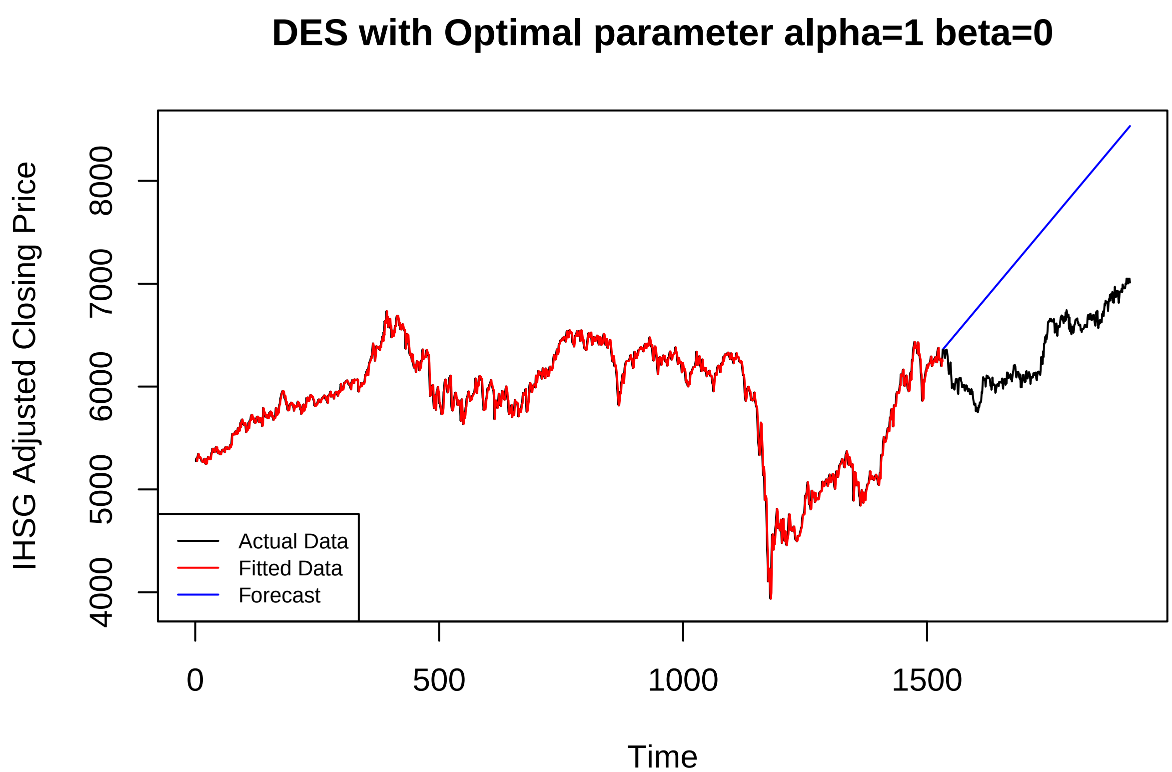


Double Exponential Smoothing

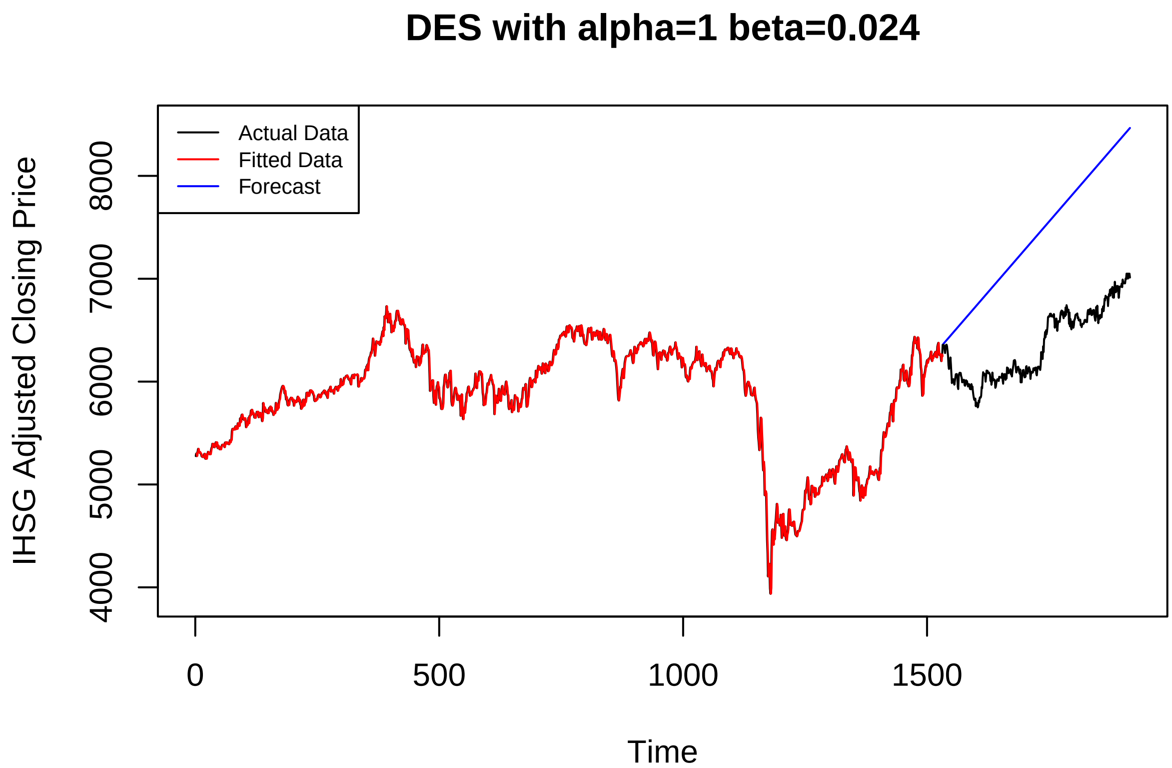
```
des.1 <- HoltWinters(ts.train, alpha = 1, beta=0.024, gamma=F)
des.2 <- HoltWinters(ts.train, alpha = 0.86, beta=0.01, gamma=F)
des.opt <- HoltWinters(ts.train, gamma=F)

des.opt #optimum parameter for des a=1, b=0.01499698
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = ts.train, gamma = F)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0.01499698
##  gamma: FALSE
##
## Coefficients:
##          [,1]
## a 6358.208984
## b    5.661422
```

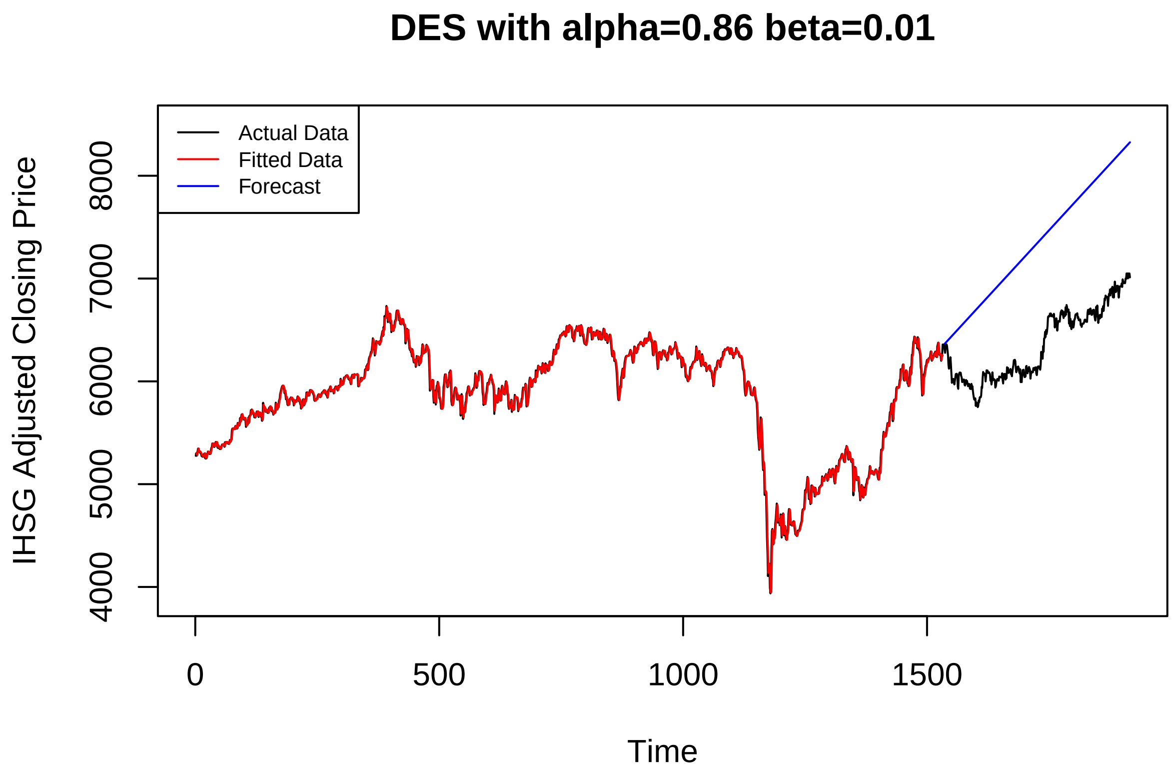
```
fc.des1 <- predict(des.1, n.ahead = test.prop)
fc.des2 <- predict(des.2, n.ahead = test.prop)
fc.desopt <- predict(des.opt, n.ahead = test.prop)

plot(ts.train,main="DES with Optimal parameter alpha=1 beta=0",
     type="l",col="black",pch=12, ylab="IHSG Adjusted Closing Price",
     xlim=c(0,1916),ylim=c(3900,8500))
lines(des.opt$fitted[,2],type="l",col="red")
lines(fc.desopt,type="l",col="blue")
lines(ts.test,type="l")
legend("bottomleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```




```
plot(ts.train,main="DES with alpha=1 beta=0.024",
     type="l",col="black",pch=12, ylab="IHSG Adjusted Closing Price",
     xlim=c(0,1916),ylim=c(3900,8500))
lines(des.1$fitted[,2],type="l",col="red")
lines(fc.des1,type="l",col="blue")
lines(ts.test,type="l")
legend("topleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```

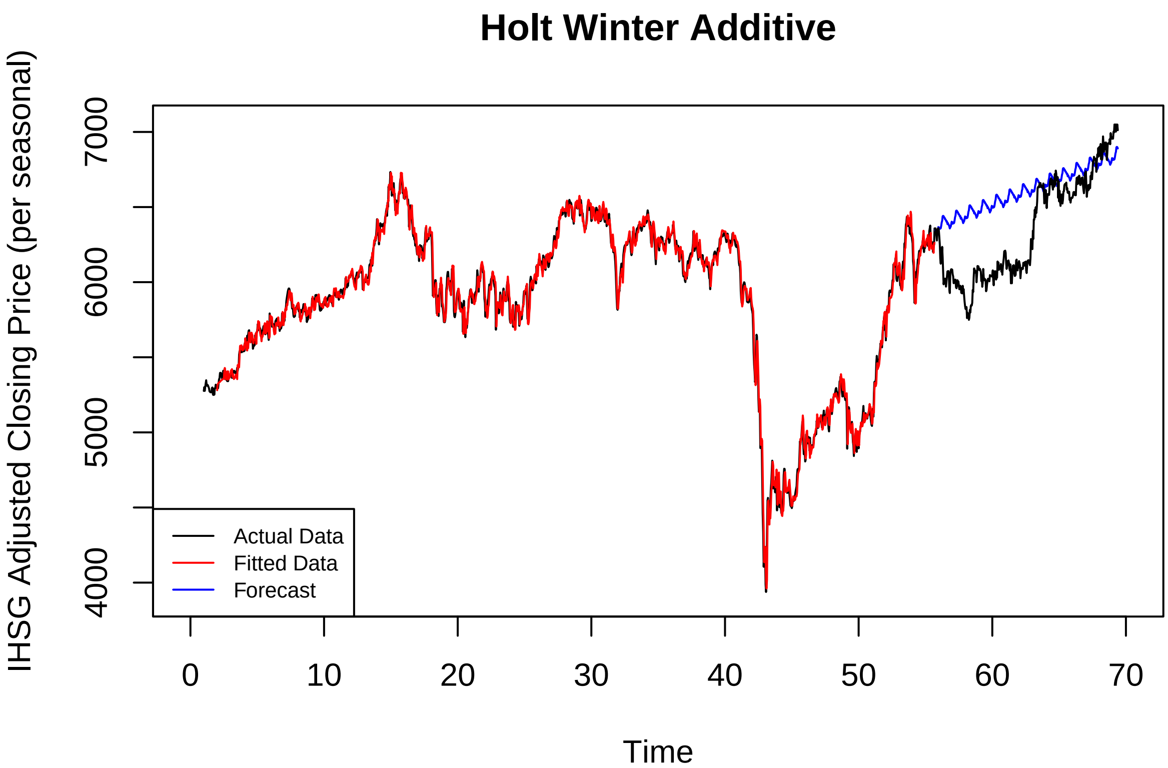



```r
plot(ts.train,main="DES with alpha=0.86 beta=0.01",
     type="l",col="black",pch=12, ylab="IHSG Adjusted Closing Price",
     xlim=c(0,1916),ylim=c(3900,8500))
lines(des.2$fitted[,2],type="l",col="red")
lines(fc.des2,type="l",col="blue")
lines(ts.test,type="l")
legend("topleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```



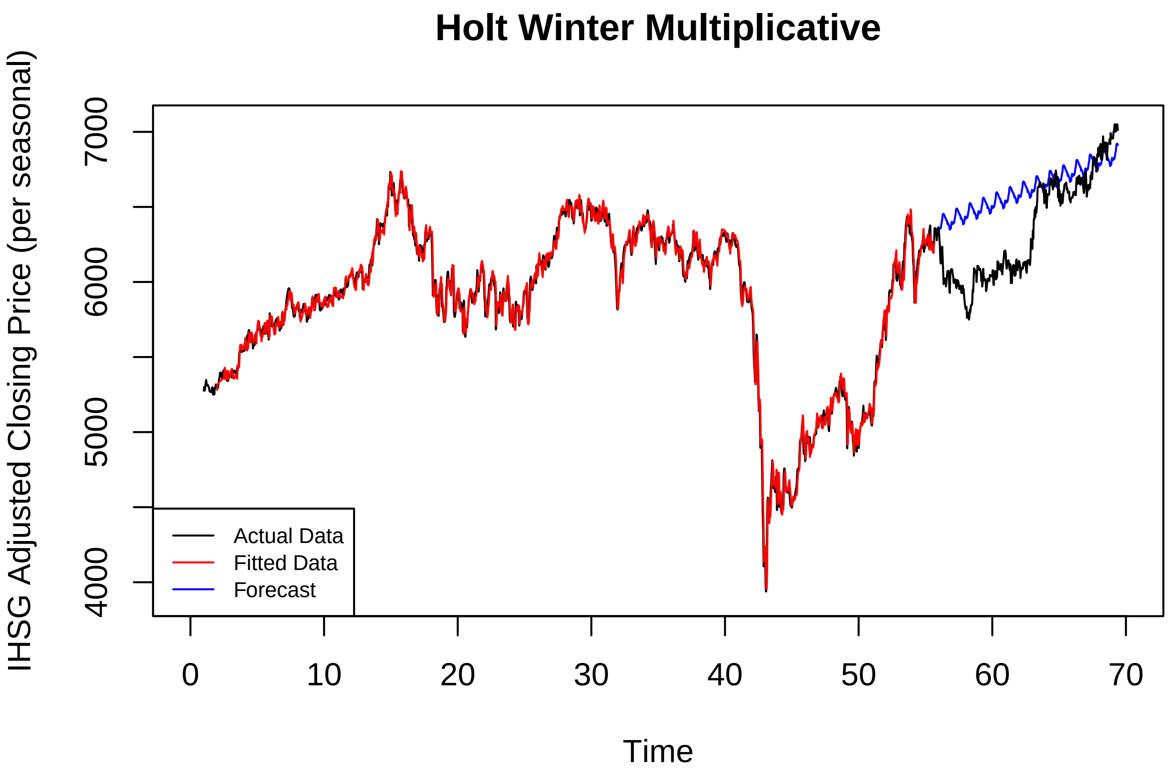
Holtwinter Additive

```
HWA <- HoltWinters(seasonal.train, seasonal = "additive")
fc.HWA <- forecast(HWA, h=test.prop)
predictHWA <- predict(HWA, n.ahead=test.prop)

plot(seasonal.train,main="Holt Winter Additive",type="l",col="black",pch=12,
     ylim=c(3900,7050),xlim=c(0,70),
     ylab="IHSG Adjusted Closing Price (per seasonal)")
lines(HWA$fitted[,2],type="l",col="red")
lines(predictHWA,type="l",col="blue")
lines(seasonal.test,type="l")
legend("bottomleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1, cex=0.7)
```



Holtwinter Multiplicative

```
HWM <- HoltWinters(seasonal.train, seasonal = "multiplicative")
fc.HWM <- forecast(HWM, h=test.prop)
predictHWM <- predict(HWM, n.ahead=test.prop)

plot(seasonal.train,main="Holt Winter Multiplicative",type="l",col="black",pch=12,
     ylim=c(3900,7050),xlim=c(0,70),
     ylab="IHSG Adjusted Closing Price (per seasonal)")
lines(HWM$fitted[,2],type="l",col="red")
lines(predictHWM,type="l",col="blue")
lines(seasonal.test,type="l")
legend("bottomleft",c("Actual Data","Fitted Data","Forecast"),
       col=c("black","red","blue"),lty=1,cex=0.7)
```



Membandingkan nilai RMSE dari data training

```
error.sma <- ts.train - data.fc[1:length(ts.train)]
RMSE.sma <- sqrt(mean(error.sma[test.prop+1:length(ts.test)]^2))
error.dma = ts.train-data.fc2[1:length(ts.train)]
RMSE.dma = sqrt(mean(error.dma[(m*2):length(ts.train)]^2))

RMSE.ses1 <- sqrt(ses.1$SSE/length(ts.train))
RMSE.ses2 <- sqrt(ses.2$SSE/length(ts.train))
RMSE.sesopt <- sqrt(ses.opt$SSE/length(ts.train))
RMSE.des1 <- sqrt(des.1$SSE/length(ts.train))
RMSE.des2 <- sqrt(des.2$SSE/length(ts.train))
RMSE.desopt <- sqrt(des.opt$SSE/length(ts.train))

RMSE.HWA <- sqrt(HWA$SSE/length(seasonal.train))
RMSE.HWM <- sqrt(HWM$SSE/length(seasonal.train))

err <- data.frame(metode=c("SMA","DMA","SES 1","SES 2","SES opt",
                           "DES 1", "DES 2", "DES opt",
                           "HW Additive", "HW Multiplicative"),
                  RMSE=c(RMSE.sma, RMSE.dma, RMSE.ses1, RMSE.ses2, RMSE.sesopt,
                         RMSE.des1, RMSE.des2, RMSE.desopt, RMSE.HWA, RMSE.HWM))
kable(err)
```

metode	RMSE
SMA	64.89666
DMA	66.31930
SES 1	57.28156
SES 2	47.65417
SES opt	46.61853
DES 1	46.87520
DES 2	48.47003
DES opt	46.84327
HW Additive	48.82866
HW Multiplicative	49.24107

Membandingkan nilai RMSE dari data testing

```
test.RMSE.SMA <- sqrt(mean((tail(data.gab$forecast, test.prop)-ts.test)^2))
test.RMSE.DMA <- sqrt(mean((tail(data.gab2$forecast, test.prop)-ts.test)^2))

test.RMSE.SES1 <- sqrt(mean((fc.ses1-ts.test)^2))
test.RMSE.SES2 <- sqrt(mean((fc.ses2-ts.test)^2))
test.RMSE.SESopt <- sqrt(mean((fc.sesopt-ts.test)^2))
test.RMSE.DES1 <- sqrt(mean((fc.des1-ts.test)^2))
test.RMSE.DES2 <- sqrt(mean((fc.des2-ts.test)^2))
test.RMSE.DESopt <- sqrt(mean((fc.desopt-ts.test)^2))

test.RMSE.HWA <- sqrt(mean((fc.HWA$mean[1:test.prop]-ts.test)^2))
test.RMSE.HWM <- sqrt(mean((fc.HWM$mean[1:test.prop]-ts.test)^2))

test.err <- data.frame(metode=c("SMA","DMA","SES 1","SES 2","SES opt",
                                "DES 1", "DES 2", "DES opt",
                                "HW Additive","HW Multiplicative"),
                       RMSE=c(test.RMSE.SMA, test.RMSE.DMA,
                              test.RMSE.SES1, test.RMSE.SES2, test.RMSE.SESopt,
                              test.RMSE.DES1, test.RMSE.DES2, test.RMSE.DESopt,
                              test.RMSE.HWA, test.RMSE.HWM))
kable(test.err)
```

metode	RMSE
SMA	354.2710
DMA	7660.7545
SES 1	352.0963
SES 2	347.9871
SES opt	347.8415
DES 1	1109.8895
DES 2	1026.5869
DES opt	1146.8849
HW Additive	342.6330
HW Multiplicative	345.4335

Membandingkan nilai MAPE dari data testing

```
MAPE.sma <- mean(abs((ts.test-tail(data.gab$forecast, test.prop))/ts.test))*100
MAPE.dma <- mean(abs((ts.test-tail(data.gab2$forecast, test.prop))/ts.test))*100

MAPE.ses1 <- mean(abs((fc.ses1 - ts.test)/ts.test)) * 100
MAPE.ses2 <- mean(abs((fc.ses2 - ts.test)/ts.test)) * 100
MAPE.sesopt <- mean(abs((fc.sesopt - ts.test)/ts.test)) * 100

MAPE.des1 <- mean(abs((fc.des1 - ts.test)/ts.test)) * 100
MAPE.des2 <- mean(abs((fc.des2 - ts.test)/ts.test)) * 100
MAPE.desopt <- mean(abs((fc.desopt - ts.test)/ts.test)) * 100

MAPE.HWA <- mean(abs((fc.HWA$mean - seasonal.test)/seasonal.test)) * 100
MAPE.HWM <- mean(abs((fc.HWM$mean - seasonal.test)/seasonal.test)) * 100

MAPE <- data.frame(metode=c("SMA","DMA","SES 1","SES 2","SES opt",
                            "DES 1", "DES 2", "DES opt",
                            "HW Additive","HW Multiplicative"),
                   MAPE=c(MAPE.sma, MAPE.dma,
                          MAPE.ses1, MAPE.ses2, MAPE.sesopt,
                          MAPE.des1, MAPE.des2, MAPE.desopt,
                          MAPE.HWA, MAPE.HWM))
kable(MAPE)
```

metode	MAPE
SMA	4.928133
DMA	104.119740
SES 1	4.943376
SES 2	5.005881
SES opt	5.024747
DES 1	16.516330
DES 2	15.323370
DES opt	17.028077
HW Additive	4.547781
HW Multiplicative	4.589989

Kesimpulan

Berdasarkan nilai RMSE dan MAPE dari data testing, terlihat bahwa metode pemulusan HoltWinter Seasonal (terutama Additive) merupakan metode pemulusan yang paling baik dalam melakukan peramalan Adjusted Closing Price IHSG dalam periode 1 Januari 2017 hingga 31 Maret 2022.