Tugas-Praktikum-2-MPDW.R

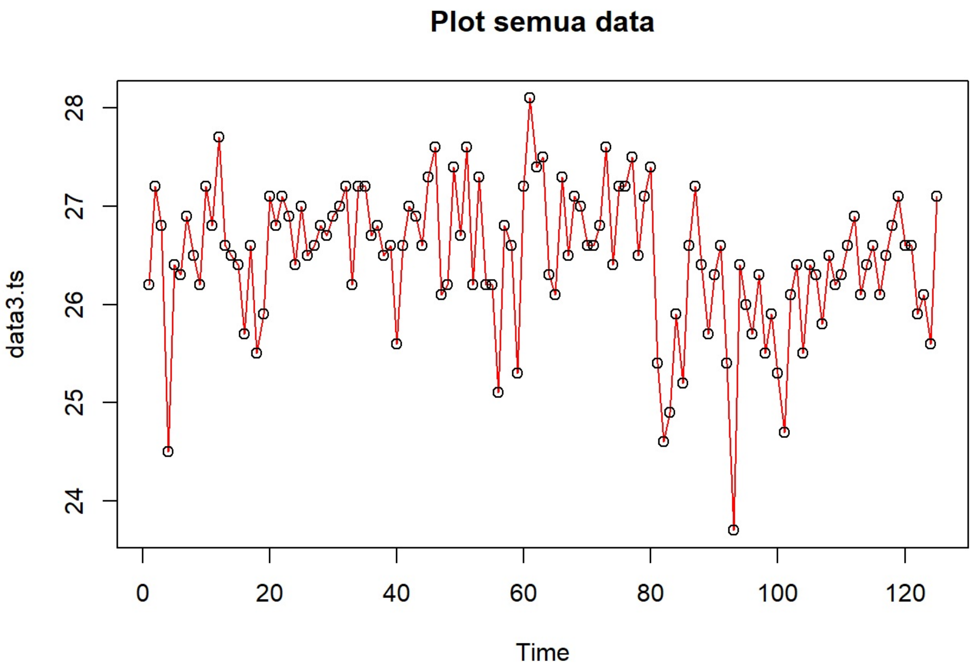
VIVOBOOK ASUS

2025-09-01

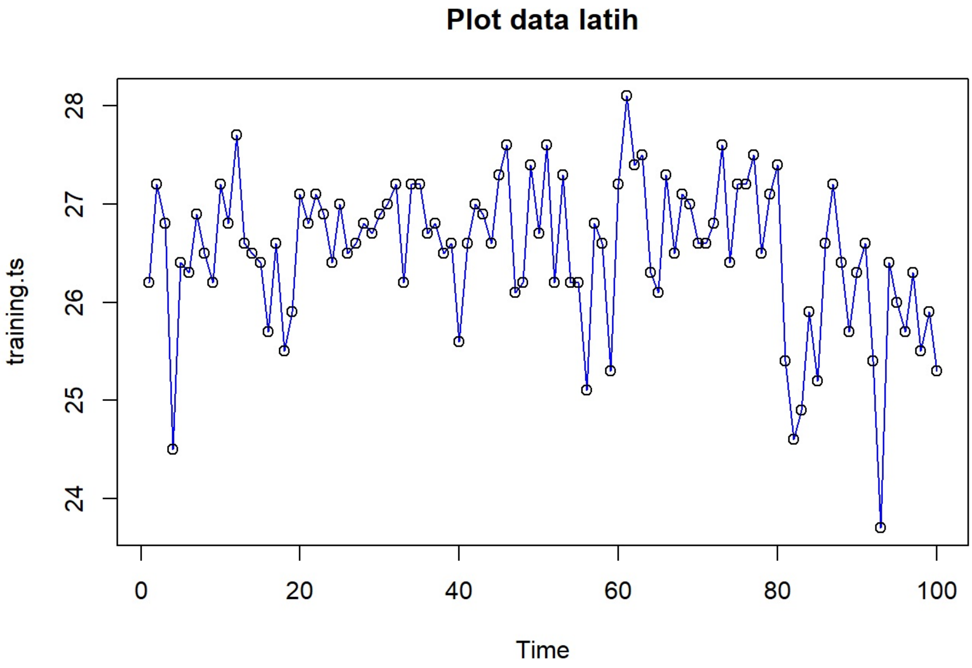
```
library(rio)

data3 <- import("https://raw.githubusercontent.com/dianiraraspp/mpdw/main/Pertemuan%202/Data%20TAVG%20376-500%20Di
ani%20Raras%20Puspita.xlsx")

head(data3)
```

```
##   Periode Nilai
## 1     376  26.2
## 2     377  27.2
## 3     378  26.8
## 4     379  24.5
## 5     380  26.4
## 6     381  26.3
```

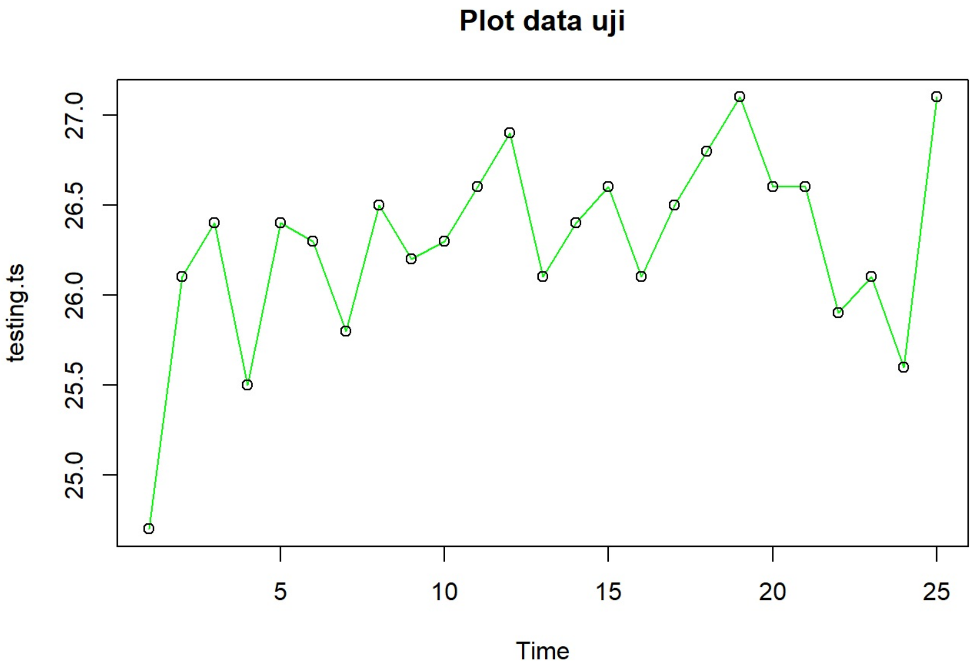
```
data3.ts <- ts(data3[,2], frequency = 1)

# Membagi data: 100 training, 25 testing
training <- data3[1:100, 2]
testing  <- data3[101:125, 2]

# Time series harian (tanpa musiman, frequency = 1)
training.ts <- ts(training, frequency = 1)
testing.ts  <- ts(testing, frequency = 1)

# Plot semua data
plot(data3.ts, col="red", main="Plot semua data")
points(data3.ts)
```



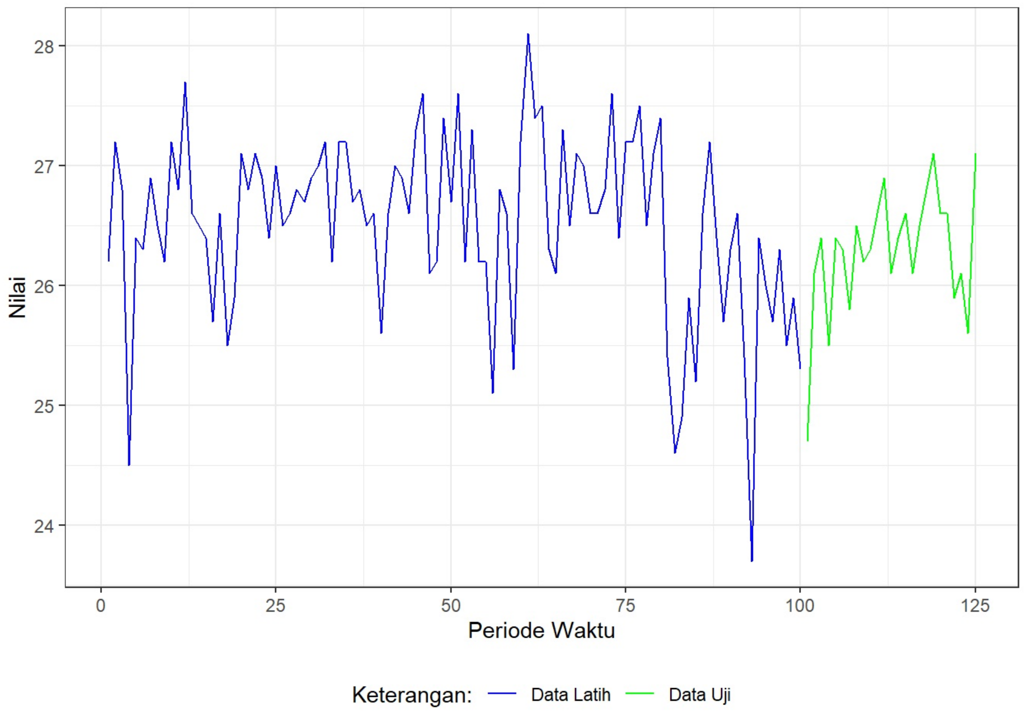

```
# Plot training
plot(training.ts, col="blue", main="Plot data latih")
points(training.ts)
```




```
# Plot testing
plot(testing.ts, col="green", main="Plot data uji")
points(testing.ts)
```



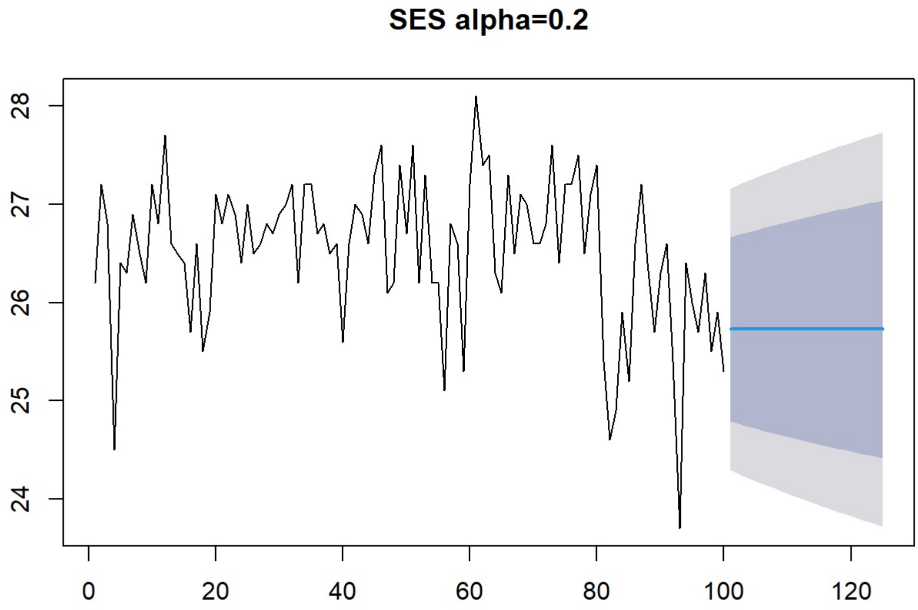
```r
library(ggplot2)

# Buat kolom Periode untuk setiap subset
training_df <- data.frame(
  Periode = 1:length(training),
  Yt = training
)

testing_df <- data.frame(
  Periode = (length(training)+1):(length(training)+length(testing)),
  Yt = testing
)

# Plot gabungan dengan ggplot2
ggplot() +
  geom_line(data = training_df, aes(x = Periode, y = Yt, col = "Data Latih")) +
  geom_line(data = testing_df, aes(x = Periode, y = Yt, col = "Data Uji")) +
  labs(x = "Periode Waktu", y = "Nilai", color = "Legend") +
  scale_colour_manual(name="Keterangan:",
                      breaks = c("Data Latih", "Data Uji"),
                      values = c("blue", "green")) +
  theme_bw() +
  theme(legend.position = "bottom",
        plot.caption = element_text(hjust=0.5, size=12))
```




```r
library(forecast)
```

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

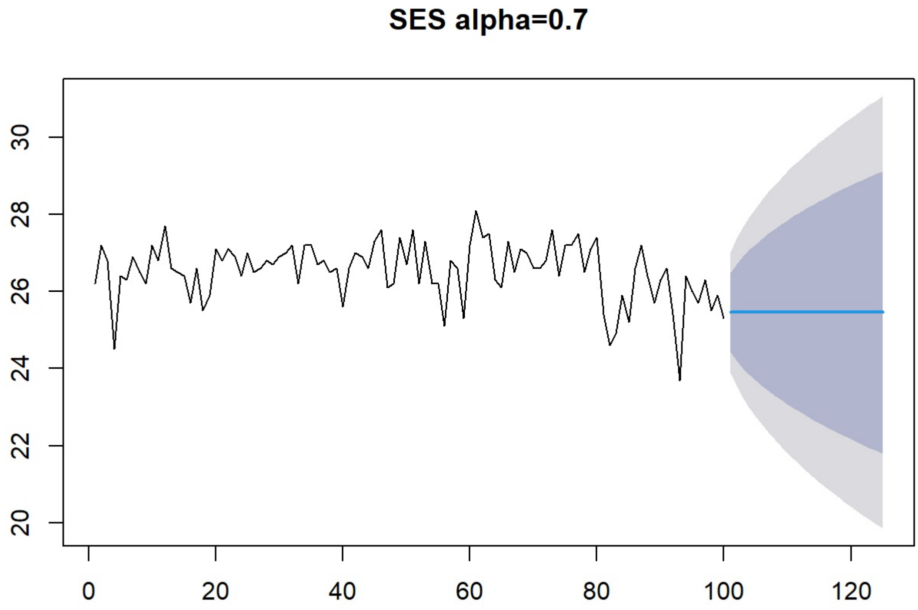
```r
# SES dengan alpha = 0.2
ses.1 <- ses(training.ts, h = length(testing.ts), alpha = 0.2)
plot(ses.1, main="SES alpha=0.2")
```




```
ses.1
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101       25.72628 24.78941 26.66316 24.29345 27.15911
## 102       25.72628 24.77085 26.68171 24.26508 27.18748
## 103       25.72628 24.75265 26.69991 24.23724 27.21532
## 104       25.72628 24.73479 26.71778 24.20992 27.24264
## 105       25.72628 24.71724 26.73533 24.18308 27.26948
## 106       25.72628 24.69999 26.75258 24.15670 27.29586
## 107       25.72628 24.68302 26.76954 24.13075 27.32181
## 108       25.72628 24.66633 26.78623 24.10522 27.34734
## 109       25.72628 24.64989 26.80267 24.08009 27.37247
## 110       25.72628 24.63371 26.81886 24.05533 27.39723
## 111       25.72628 24.61776 26.83481 24.03094 27.42162
## 112       25.72628 24.60203 26.85053 24.00689 27.44567
## 113       25.72628 24.58652 26.86604 23.98317 27.46939
## 114       25.72628 24.57122 26.88134 23.95977 27.49279
## 115       25.72628 24.55612 26.89644 23.93668 27.51588
## 116       25.72628 24.54122 26.91134 23.91388 27.53868
## 117       25.72628 24.52650 26.92607 23.89137 27.56119
## 118       25.72628 24.51195 26.94061 23.86913 27.58344
## 119       25.72628 24.49758 26.95498 23.84715 27.60542
## 120       25.72628 24.48338 26.96919 23.82542 27.62714
## 121       25.72628 24.46933 26.98323 23.80394 27.64862
## 122       25.72628 24.45544 26.99712 23.78270 27.66986
## 123       25.72628 24.44170 27.01086 23.76169 27.69087
## 124       25.72628 24.42811 27.02445 23.74090 27.71166
## 125       25.72628 24.41466 27.03791 23.72032 27.73224
```

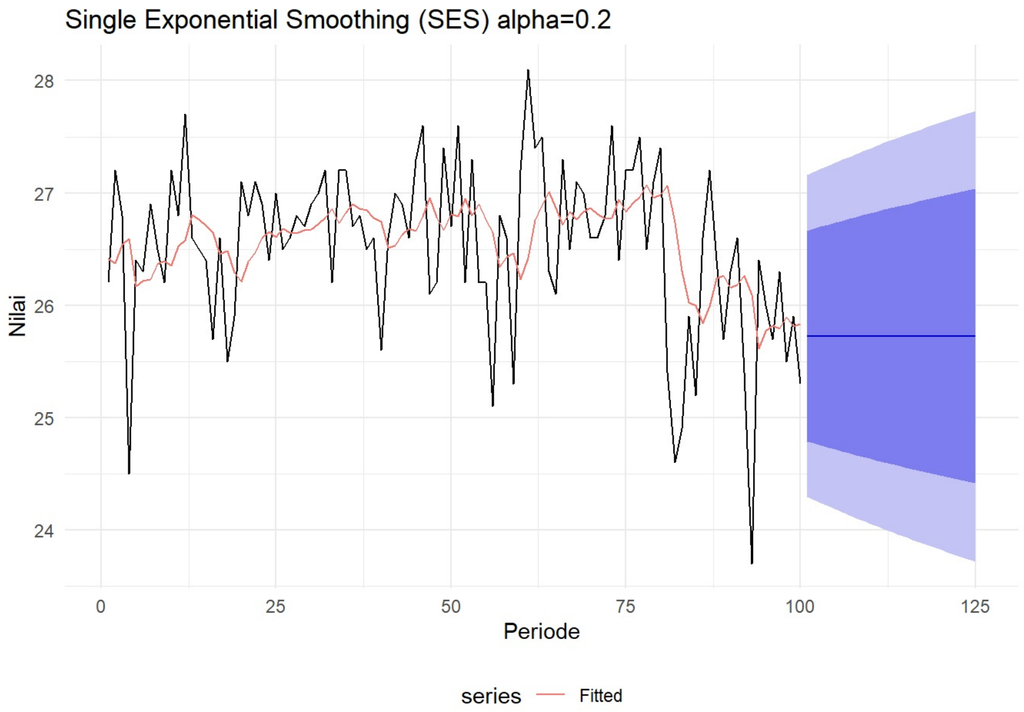
```
# SES dengan alpha = 0.7
ses.2 <- ses(training.ts, h = length(testing.ts), alpha = 0.7)
plot(ses.2, main="SES alpha=0.7")
```



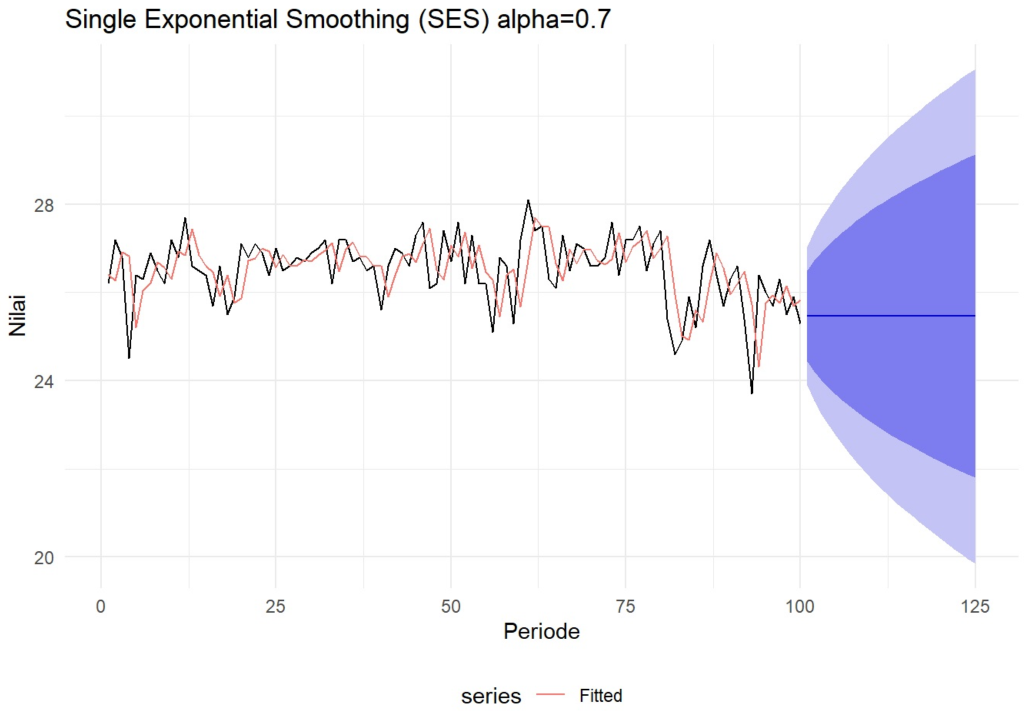

```
ses.2
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101        25.4613 24.43456 26.48805 23.89103 27.03157
## 102        25.4613 24.20800 26.71460 23.54455 27.37806
## 103        25.4613 24.01655 26.90606 23.25174 27.67087
## 104        25.4613 23.84765 27.07496 22.99343 27.92917
## 105        25.4613 23.69483 27.22778 22.75971 28.16289
## 106        25.4613 23.55421 27.36839 22.54466 28.37795
## 107        25.4613 23.42328 27.49933 22.34441 28.57820
## 108        25.4613 23.30026 27.62235 22.15627 28.76634
## 109        25.4613 23.18388 27.73873 21.97828 28.94433
## 110        25.4613 23.07316 27.84945 21.80895 29.11365
## 111        25.4613 22.96735 27.95525 21.64713 29.27547
## 112        25.4613 22.86585 28.05675 21.49191 29.43070
## 113        25.4613 22.76818 28.15443 21.34253 29.58008
## 114        25.4613 22.67393 28.24868 21.19838 29.72423
## 115        25.4613 22.58276 28.33985 21.05895 29.86366
## 116        25.4613 22.49439 28.42822 20.92380 29.99881
## 117        25.4613 22.40858 28.51403 20.79256 30.13005
## 118        25.4613 22.32511 28.59750 20.66491 30.25770
## 119        25.4613 22.24381 28.67880 20.54057 30.38203
## 120        25.4613 22.16451 28.75809 20.41930 30.50331
## 121        25.4613 22.08708 28.83553 20.30087 30.62173
## 122        25.4613 22.01138 28.91122 20.18511 30.73750
## 123        25.4613 21.93731 28.98529 20.07183 30.85078
## 124        25.4613 21.86477 29.05784 19.96088 30.96173
## 125        25.4613 21.79366 29.12895 19.85212 31.07048
```

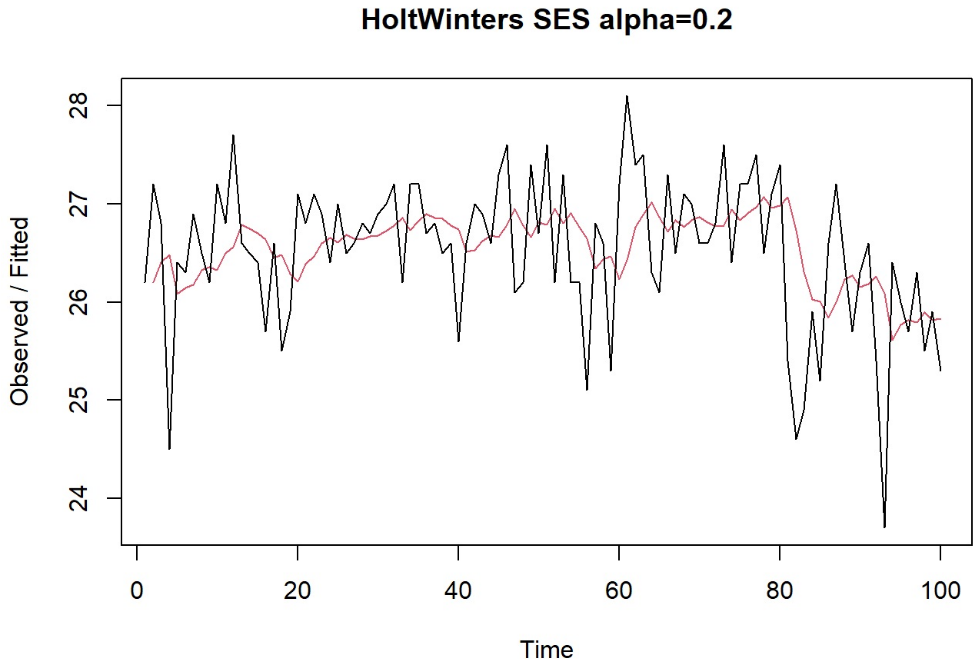
```
library(forecast)
library(ggplot2)

autoplot(ses.1) +
  autolayer(fitted(ses.1), series = "Fitted") +
  ylab("Nilai") +
  xlab("Periode") +
  ggtitle("Single Exponential Smoothing (SES) alpha=0.2") +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
autoplot(ses.2) +
  autolayer(fitted(ses.2), series = "Fitted") +
  ylab("Nilai") +
  xlab("Periode") +
  ggtitle("Single Exponential Smoothing (SES) alpha=0.7") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



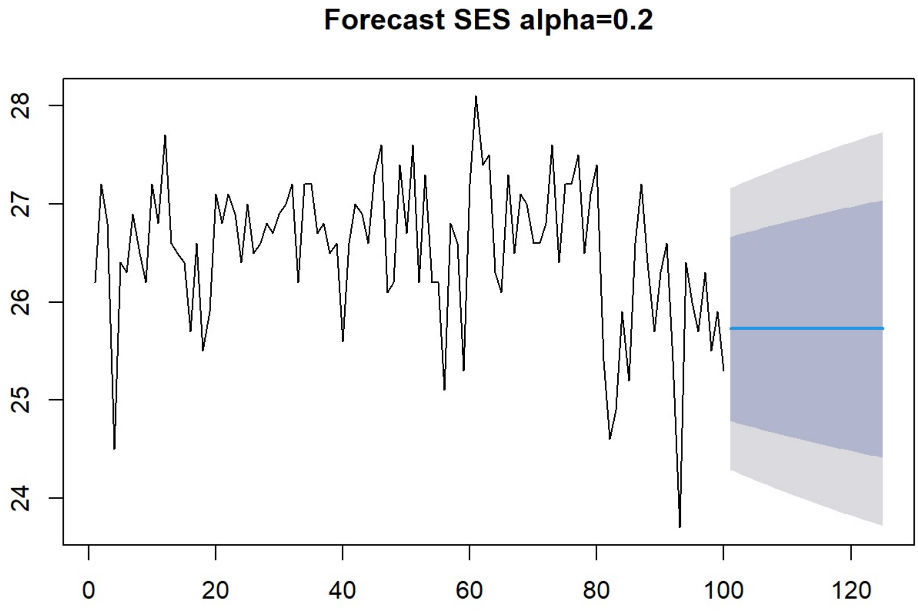

```
library(forecast)

# SES dengan HoltWinters, alpha = 0.2
ses1 <- HoltWinters(training.ts, alpha = 0.2, beta = FALSE, gamma = FALSE)
plot(ses1, main = "HoltWinters SES alpha=0.2")
```



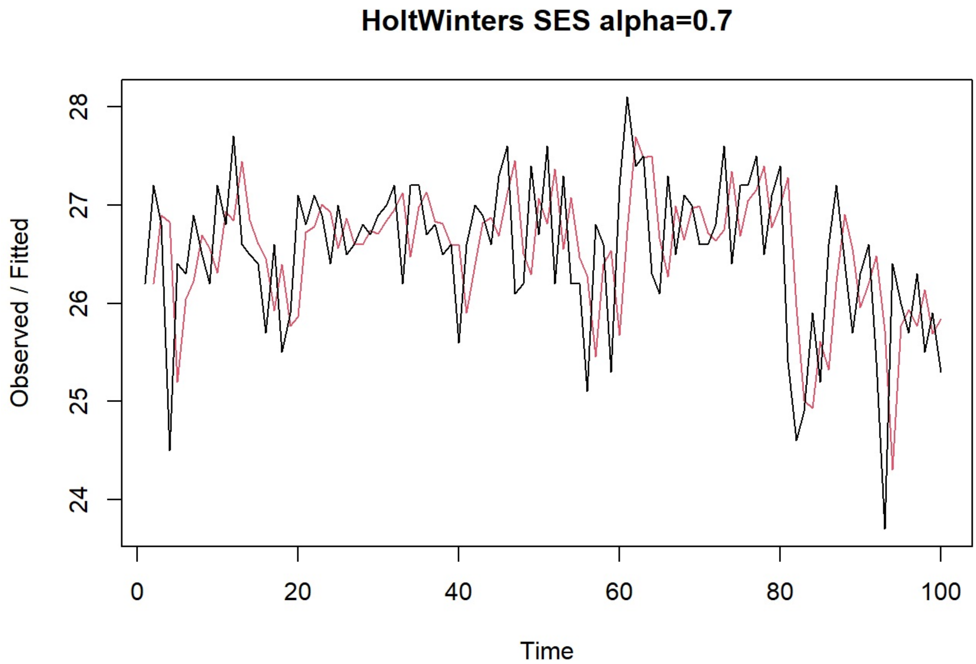

```
# Forecast
ramalan1 <- forecast(ses1, h = length(testing.ts))
ramalan1
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101       25.72628 24.78876 26.66380 24.29247 27.16009
## 102       25.72628 24.77020 26.68237 24.26408 27.18849
## 103       25.72628 24.75198 26.70058 24.23622 27.21634
## 104       25.72628 24.73411 26.71846 24.20888 27.24368
## 105       25.72628 24.71654 26.73602 24.18202 27.27054
## 106       25.72628 24.69928 26.75328 24.15562 27.29694
## 107       25.72628 24.68231 26.77026 24.12966 27.32290
## 108       25.72628 24.66560 26.78696 24.10411 27.34845
## 109       25.72628 24.64916 26.80341 24.07896 27.37360
## 110       25.72628 24.63296 26.81961 24.05419 27.39838
## 111       25.72628 24.61699 26.83557 24.02977 27.42279
## 112       25.72628 24.60126 26.85130 24.00571 27.44685
## 113       25.72628 24.58574 26.86682 23.98198 27.47059
## 114       25.72628 24.57043 26.88213 23.95856 27.49400
## 115       25.72628 24.55532 26.89724 23.93545 27.51711
## 116       25.72628 24.54040 26.91216 23.91264 27.53992
## 117       25.72628 24.52567 26.92689 23.89011 27.56245
## 118       25.72628 24.51112 26.94144 23.86785 27.58471
## 119       25.72628 24.49674 26.95582 23.84586 27.60670
## 120       25.72628 24.48252 26.97004 23.82412 27.62844
## 121       25.72628 24.46847 26.98409 23.80262 27.64994
## 122       25.72628 24.45457 26.99799 23.78137 27.67120
## 123       25.72628 24.44082 27.01174 23.76034 27.69222
## 124       25.72628 24.42722 27.02534 23.73954 27.71303
## 125       25.72628 24.41376 27.03881 23.71895 27.73361
```

```
plot(ramalan1, main = "Forecast SES alpha=0.2")
```



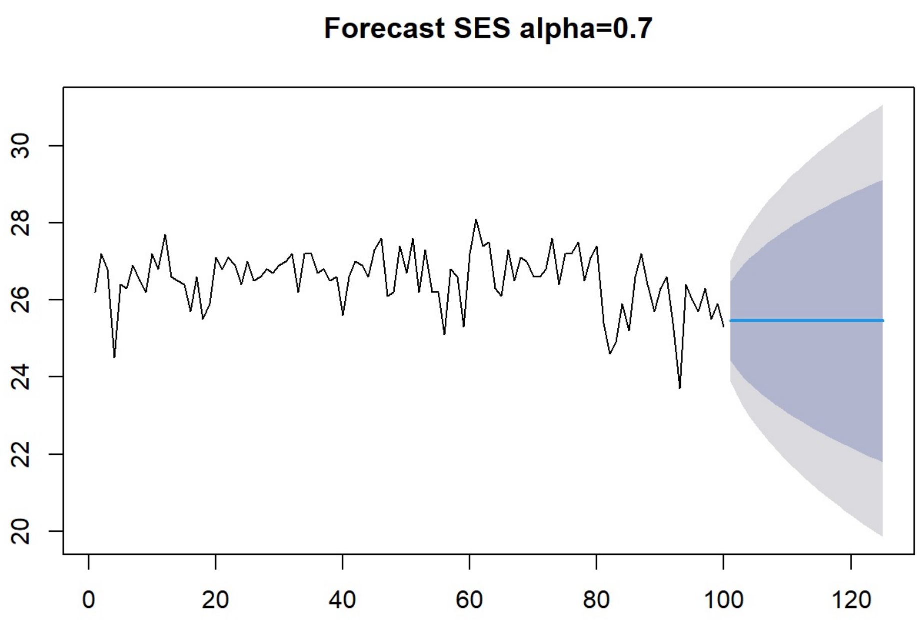

```
# SES dengan HoltWinters, alpha = 0.7
ses2 <- HoltWinters(training.ts, alpha = 0.7, beta = FALSE, gamma = FALSE)
plot(ses2, main = "HoltWinters SES alpha=0.7")
```



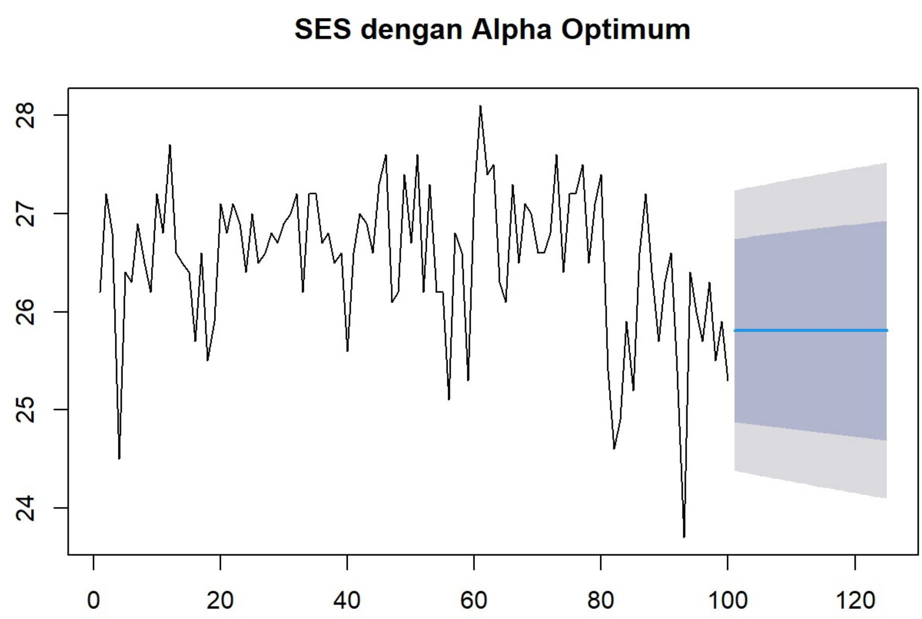

```
# Forecast
ramalan2 <- forecast(ses2, h = length(testing.ts))
ramalan2
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101        25.4613 24.43423 26.48838 23.89053 27.03208
## 102        25.4613 24.20760 26.71501 23.54393 27.37868
## 103        25.4613 24.01608 26.90653 23.25103 27.67158
## 104        25.4613 23.84713 27.07548 22.99263 27.92997
## 105        25.4613 23.69426 27.22835 22.75884 28.16377
## 106        25.4613 23.55360 27.36901 22.54372 28.37889
## 107        25.4613 23.42262 27.49999 22.34340 28.57920
## 108        25.4613 23.29956 27.62305 22.15520 28.76740
## 109        25.4613 23.18314 27.73947 21.97715 28.94545
## 110        25.4613 23.07239 27.85022 21.80777 29.11483
## 111        25.4613 22.96655 27.95606 21.64590 29.27670
## 112        25.4613 22.86502 28.05759 21.49062 29.43198
## 113        25.4613 22.76731 28.15530 21.34120 29.58141
## 114        25.4613 22.67303 28.24958 21.19700 29.72561
## 115        25.4613 22.58183 28.34078 21.05752 29.86508
## 116        25.4613 22.49343 28.42918 20.92233 30.00027
## 117        25.4613 22.40759 28.51502 20.79105 30.13156
## 118        25.4613 22.32410 28.59851 20.66336 30.25925
## 119        25.4613 22.24277 28.67984 20.53898 30.38362
## 120        25.4613 22.16345 28.75916 20.41767 30.50494
## 121        25.4613 22.08599 28.83662 20.29921 30.62340
## 122        25.4613 22.01027 28.91234 20.18340 30.73920
## 123        25.4613 21.93617 28.98643 20.07008 30.85252
## 124        25.4613 21.86361 29.05900 19.95910 30.96351
## 125        25.4613 21.79247 29.13013 19.85031 31.07229
```

```
plot(ramalan2, main = "Forecast SES alpha=0.7")
```




```
library(forecast)

# SES dengan alpha optimum
ses.opt <- ses(training.ts, h = length(testing.ts), alpha = NULL)
plot(ses.opt, main="SES dengan Alpha Optimum")
```



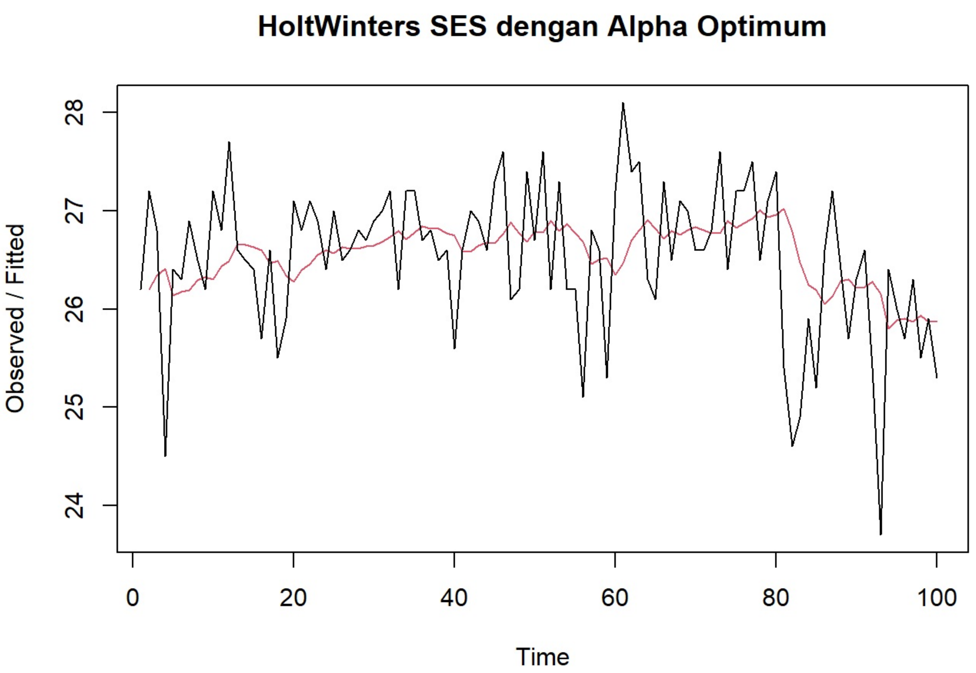

```
ses.opt
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101        25.8096 24.87660 26.74261 24.38270 27.23651
## 102        25.8096 24.86812 26.75109 24.36973 27.24948
## 103        25.8096 24.85971 26.75950 24.35687 27.26234
## 104        25.8096 24.85138 26.76783 24.34412 27.27508
## 105        25.8096 24.84312 26.77609 24.33149 27.28772
## 106        25.8096 24.83492 26.78428 24.31896 27.30025
## 107        25.8096 24.82680 26.79241 24.30654 27.31267
## 108        25.8096 24.81874 26.80046 24.29421 27.32499
## 109        25.8096 24.81075 26.80845 24.28199 27.33721
## 110        25.8096 24.80282 26.81638 24.26987 27.34934
## 111        25.8096 24.79496 26.82425 24.25784 27.36137
## 112        25.8096 24.78715 26.83206 24.24590 27.37331
## 113        25.8096 24.77940 26.83980 24.23405 27.38516
## 114        25.8096 24.77172 26.84749 24.22229 27.39692
## 115        25.8096 24.76408 26.85512 24.21062 27.40859
## 116        25.8096 24.75651 26.86270 24.19903 27.42018
## 117        25.8096 24.74898 26.87022 24.18752 27.43168
## 118        25.8096 24.74151 26.87769 24.17610 27.44311
## 119        25.8096 24.73410 26.88511 24.16476 27.45445
## 120        25.8096 24.72673 26.89248 24.15349 27.46572
## 121        25.8096 24.71941 26.89980 24.14230 27.47691
## 122        25.8096 24.71214 26.90706 24.13118 27.48803
## 123        25.8096 24.70492 26.91429 24.12014 27.49907
## 124        25.8096 24.69775 26.92146 24.10917 27.51004
## 125        25.8096 24.69062 26.92859 24.09826 27.52094
```

```
# HoltWinters dengan alpha optimum (tanpa trend & seasonality)
HWopt <- HoltWinters(training.ts, gamma = FALSE, beta = FALSE, alpha = NULL)
HWopt
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = training.ts, alpha = NULL, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1443504
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 25.79303
```

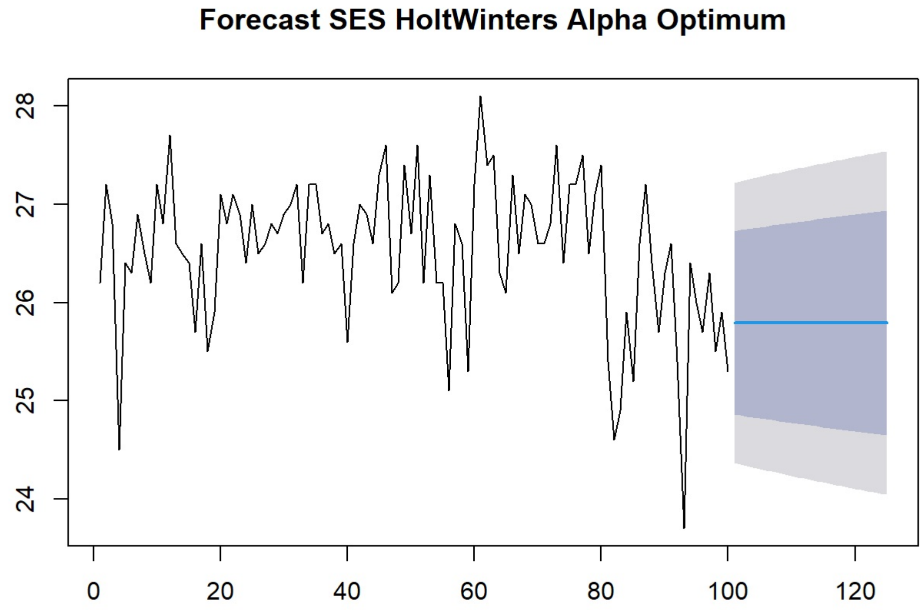
```
plot(HWopt, main="HoltWinters SES dengan Alpha Optimum")
```




```
# Ramalan dengan HoltWinters alpha optimum
ramalanopt <- forecast(HWopt, h = length(testing.ts))
ramalanopt
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101       25.79303 24.85869 26.72737 24.36407 27.22198
## 102       25.79303 24.84900 26.73705 24.34926 27.23679
## 103       25.79303 24.83941 26.74664 24.33460 27.25145
## 104       25.79303 24.82992 26.75613 24.32009 27.26596
## 105       25.79303 24.82053 26.76553 24.30572 27.28034
## 106       25.79303 24.81122 26.77483 24.29148 27.29457
## 107       25.79303 24.80200 26.78405 24.27738 27.30867
## 108       25.79303 24.79286 26.79319 24.26341 27.32264
## 109       25.79303 24.78381 26.80224 24.24956 27.33649
## 110       25.79303 24.77484 26.81121 24.23584 27.35021
## 111       25.79303 24.76594 26.82011 24.22224 27.36381
## 112       25.79303 24.75713 26.82893 24.20875 27.37730
## 113       25.79303 24.74838 26.83767 24.19538 27.39067
## 114       25.79303 24.73971 26.84634 24.18212 27.40393
## 115       25.79303 24.73111 26.85494 24.16897 27.41708
## 116       25.79303 24.72258 26.86347 24.15592 27.43013
## 117       25.79303 24.71412 26.87193 24.14298 27.44307
## 118       25.79303 24.70572 26.88033 24.13014 27.45592
## 119       25.79303 24.69739 26.88866 24.11739 27.46866
## 120       25.79303 24.68912 26.89693 24.10474 27.48131
## 121       25.79303 24.68091 26.90514 24.09219 27.49386
## 122       25.79303 24.67276 26.91329 24.07973 27.50632
## 123       25.79303 24.66467 26.92138 24.06736 27.51870
## 124       25.79303 24.65664 26.92941 24.05507 27.53098
## 125       25.79303 24.64866 26.93739 24.04287 27.54318
```

```
plot(ramalanopt, main="Forecast SES HoltWinters Alpha Optimum")
```




```
# Keakuratan Metode pada data training

# SES alpha = 0.2
SSE1 <- ses1$SSE
MSE1 <- ses1$SSE / length(training.ts)
RMSE1 <- sqrt(MSE1)

akurasi1 <- matrix(c(SSE1, MSE1, RMSE1), nrow = 3, ncol = 1)
row.names(akurasi1) <- c("SSE", "MSE", "RMSE")
colnames(akurasi1) <- c("Akurasi alpha=0.2")
akurasi1
```

```
##      Akurasi alpha=0.2
## SSE         52.5027388
## MSE          0.5250274
## RMSE         0.7245877
```

```
# SES alpha = 0.7
SSE2 <- ses2$SSE
MSE2 <- ses2$SSE / length(training.ts)
RMSE2 <- sqrt(MSE2)

akurasi2 <- matrix(c(SSE2, MSE2, RMSE2), nrow = 3, ncol = 1)
row.names(akurasi2) <- c("SSE", "MSE", "RMSE")
colnames(akurasi2) <- c("Akurasi alpha=0.7")
akurasi2
```

```
##      Akurasi alpha=0.7
## SSE         62.9558431
## MSE          0.6295584
## RMSE         0.7934472
```

```
# Cara Manual SES HoltWinters

# Alpha = 0.2
fitted1 <- ramalan1$fitted
residuals1 <- ramalan1$residuals
head(residuals1)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1]      NA  1.0000  0.4000 -1.9800  0.3160  0.1528
```

```
resid1 <- training.ts - fitted1
head(resid1)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1]      NA  1.0000  0.4000 -1.9800  0.3160  0.1528
```

```
SSE.1 <- sum(residuals1[2:length(training.ts)]^2)
SSE.1
```

```
## [1] 52.50274
```

```
MSE.1 <- SSE.1 / length(training.ts)
MSE.1
```

```
## [1] 0.5250274
```

```
MAPE.1 <- sum(abs(residuals1[2:length(training.ts)] / training.ts[2:length(training.ts)]) * 100) / length(trainin
g.ts)
MAPE.1
```

```
## [1] 2.124049
```

```
akurasi.1 <- matrix(c(SSE.1, MSE.1, MAPE.1), nrow = 3)
row.names(akurasi.1) <- c("SSE", "MSE", "MAPE")
colnames(akurasi.1) <- c("Akurasi alpha=0.2")
akurasi.1
```

```
##      Akurasi alpha=0.2
## SSE         52.5027388
## MSE          0.5250274
## MAPE         2.1240486
```

```
# Alpha = 0.7
fitted2 <- ramalan2$fitted
residuals2 <- ramalan2$residuals
head(residuals2)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1]      NA  1.0000 -0.1000 -2.3300  1.2010  0.2603
```

```
resid2 <- training.ts - fitted2
head(resid2)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1]      NA  1.0000 -0.1000 -2.3300  1.2010  0.2603
```

```
SSE.2 <- sum(residuals2[2:length(training.ts)]^2)
SSE.2
```

```
## [1] 62.95584
```

```
MSE.2 <- SSE.2 / length(training.ts)
MSE.2
```

```
## [1] 0.6295584
```

```
MAPE.2 <- sum(abs(residuals2[2:length(training.ts)] / training.ts[2:length(training.ts)]) * 100) / length(trainin
g.ts)
MAPE.2
```

```
## [1] 2.354092
```

```
akurasi.2 <- matrix(c(SSE.2, MSE.2, MAPE.2), nrow = 3)
row.names(akurasi.2) <- c("SSE", "MSE", "MAPE")
colnames(akurasi.2) <- c("Akurasi alpha=0.7")
akurasi.2
```

```
##      Akurasi alpha=0.7
## SSE         62.9558431
## MSE          0.6295584
## MAPE         2.3540922
```

```
# Jumlah observasi testing
n_test <- length(testing.ts)

# Error (forecast - aktual)
e1   <- as.numeric(ramalan1$mean)[1:n_test] - as.numeric(testing.ts)
e2   <- as.numeric(ramalan2$mean)[1:n_test] - as.numeric(testing.ts)
eopt <- as.numeric(ramalanopt$mean)[1:n_test] - as.numeric(testing.ts)

# SSE / MSE / RMSE untuk masing-masing model (abaikan NA)
SSEtesting1  <- sum(e1^2,  na.rm = TRUE)
MSEtesting1  <- mean(e1^2, na.rm = TRUE)
RMSEtesting1 <- sqrt(MSEtesting1)

SSEtesting2  <- sum(e2^2,  na.rm = TRUE)
MSEtesting2  <- mean(e2^2, na.rm = TRUE)
RMSEtesting2 <- sqrt(MSEtesting2)

SSEtestingopt  <- sum(eopt^2,  na.rm = TRUE)
MSEtestingopt  <- mean(eopt^2, na.rm = TRUE)
RMSEtestingopt <- sqrt(MSEtestingopt)

# Tabel ringkas
akurasitesting_SSE <- matrix(c(SSEtesting1, SSEtesting2, SSEtestingopt),
                             nrow = 3,
                             dimnames = list(c("SSE alpha=0.2","SSE alpha=0.7","SSE alpha optimum"), "Nilai"))
akurasitesting_MSE <- matrix(c(MSEtesting1, MSEtesting2, MSEtestingopt),
                             nrow = 3,
                             dimnames = list(c("MSE alpha=0.2","MSE alpha=0.7","MSE alpha optimum"), "Nilai"))
akurasitesting_RMSE <- matrix(c(RMSEtesting1, RMSEtesting2, RMSEtestingopt),
                              nrow = 3,
                              dimnames = list(c("RMSE alpha=0.2","RMSE alpha=0.7","RMSE alpha optimum"), "Nilai")
)

akurasitesting_SSE
```

```
##                      Nilai
## SSE alpha=0.2     14.51461
## SSE alpha=0.7     23.71210
## SSE alpha optimum 12.75138
```

```
akurasitesting_MSE
```

```
##                       Nilai
## MSE alpha=0.2     0.5805843
## MSE alpha=0.7     0.9484841
## MSE alpha optimum 0.5100553
```

```
akurasitesting_RMSE
```

```
##                        Nilai
## RMSE alpha=0.2     0.7619608
## RMSE alpha=0.7     0.9739015
## RMSE alpha optimum 0.7141815
```

```r
library(forecast)

# Misal training.ts = ts(training, frequency = 1)  # harian, start=1
# Maka testing.ts juga dibuat ts
testing_ts <- ts(testing, start = length(training.ts) + 1, frequency = 1)

# Sekarang hitung akurasi
accuracy(ramalan1, testing_ts)
```

```
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.0239252 0.7282381 0.5625518 -0.1622462 2.145504 0.8130311
## Test set      0.5617190 0.7619608 0.6720265  2.0984413 2.541293 0.9712499
##                    ACF1 Theil's U
## Training set 0.11158256        NA
## Test set     0.05521188  1.176161
```

```r
accuracy(ramalan2, testing_ts)
```

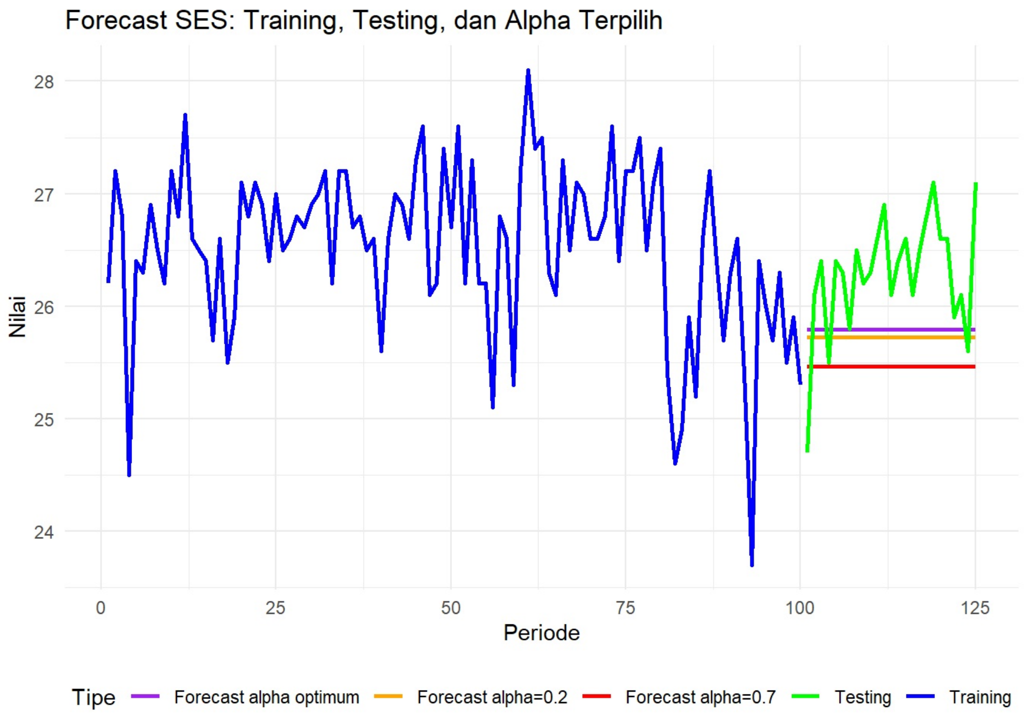
```
##                       ME      RMSE       MAE        MPE     MAPE     MASE
## Training set -0.01065941 0.7974444 0.6250881 -0.1018746 2.377871 0.903412
## Test set      0.82669712 0.9739015 0.8876013  3.1068174 3.353393 1.282811
##                     ACF1 Theil's U
## Training set -0.20822344        NA
## Test set      0.05521188  1.542631
```

```r
accuracy(ramalanopt, testing_ts)
```

```
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.02847827 0.7259373 0.5524138 -0.181378 2.108247 0.7983791
## Test set      0.49497402 0.7141815 0.6213003  1.844443 2.350709 0.8979376
##                    ACF1 Theil's U
## Training set 0.15161657        NA
## Test set     0.05521188  1.089598
```

```r
library(ggplot2)
library(forecast)

# Buat data frame untuk plotting
df_train <- data.frame(Periode = 1:length(training.ts),
                       Nilai = as.numeric(training.ts),
                       Tipe = "Training")
df_test <- data.frame(Periode = (length(training.ts)+1):(length(training.ts)+length(testing.ts)),
                      Nilai = as.numeric(testing.ts),
                      Tipe = "Testing")

# Forecast alpha=0.2
df_forecast02 <- data.frame(Periode = (length(training.ts)+1):(length(training.ts)+length(testing.ts)),
                            Nilai = as.numeric(ramalan1$mean),
                            Tipe = "Forecast alpha=0.2")

# Forecast alpha=0.7
df_forecast07 <- data.frame(Periode = (length(training.ts)+1):(length(training.ts)+length(testing.ts)),
                            Nilai = as.numeric(ramalan2$mean),
                            Tipe = "Forecast alpha=0.7")

# Forecast alpha optimum
df_forecast_opt <- data.frame(Periode = (length(training.ts)+1):(length(training.ts)+length(testing.ts)),
                              Nilai = as.numeric(ramalanopt$mean),
                              Tipe = "Forecast alpha optimum")

# Gabungkan semua
df_all <- rbind(df_train, df_test, df_forecast02, df_forecast07, df_forecast_opt)

# Plot
ggplot(df_all, aes(x = Periode, y = Nilai, color = Tipe)) +
  geom_line(size=1) +
  labs(title = "Forecast SES: Training, Testing, dan Alpha Terpilih",
       x = "Periode",
       y = "Nilai") +
  scale_color_manual(values = c("Training"="blue",
                                "Testing"="green",
                                "Forecast alpha=0.2"="orange",
                                "Forecast alpha=0.7"="red",
                                "Forecast alpha optimum"="purple")) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
cat("Kesimpulan: Hasil pemulusan data menunjukkan bahwa metode single exponential smoothing dengan alpha optimum
memberikan performa paling baik dibandingkan alpha fixed (0.2 maupun 0.7).
Nilai alpha 0.2 masih cukup baik namun cenderung lambat dalam merespon perubahan, sedangkan alpha 0.7 terlalu res
ponsif sehingga menghasilkan error yang lebih besar.
Dengan nilai MAPE di bawah 5%, model pemulusan ini termasuk dalam kategori highly accurate forecast, yang berarti
hasil peramalan dapat dipercaya karena tingkat kesalahannya sangat rendah.")
```

```
## Kesimpulan: Hasil pemulusan data menunjukkan bahwa metode single exponential smoothing dengan alpha optimum me
mberikan performa paling baik dibandingkan alpha fixed (0.2 maupun 0.7).
## Nilai alpha 0.2 masih cukup baik namun cenderung lambat dalam merespon perubahan, sedangkan alpha 0.7 terlalu
responsif sehingga menghasilkan error yang lebih besar.
## Dengan nilai MAPE di bawah 5%, model pemulusan ini termasuk dalam kategori highly accurate forecast, yang bera
rti hasil peramalan dapat dipercaya karena tingkat kesalahannya sangat rendah.
```