

Project Week 2

Ananda Putra Wijaya/G1401221111

2024-08-30

Inisiasi *library*

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(TTR)
```

```
## Warning: package 'TTR' was built under R version 4.3.3
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3
## Registered S3 method overwritten by 'quantmod':
##   method             from
##   as.zoo.data.frame zoo
```

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.3.3
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 4.3.3
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

```
library(orcutt)
```

```
## Warning: package 'orcutt' was built under R version 4.3.3
```

```
library(HoRM)
```

```
## Warning: package 'HoRM' was built under R version 4.3.3
```

```
library(readxl)
```

Inisiasi data

```
DataW2<-as.data.frame(read_xlsx("C:/Tugas/Semester 5/Metode Peramalan Deret Waktu/Minggu 2/Data Project  
DataW2$Tahun <- as.numeric(DataW2$Tahun)
```

```
DataW2
```

##	Tahun	PDB
## 1	1960	84466653
## 2	1961	89249985
## 3	1962	94149984
## 4	1963	96366651
## 5	1964	107566649
## 6	1965	114339050
## 7	1966	134173375
## 8	1967	155102980
## 9	1968	150000000
## 10	1969	164900000
## 11	1970	186300000
## 12	1971	211100000
## 13	1972	235400000
## 14	1973	269500000
## 15	1974	312600000
## 16	1975	345000000
## 17	1976	386300000
## 18	1977	447000000
## 19	1978	475800000
## 20	1979	517200000
## 21	1980	613299968
## 22	1981	739100032
## 23	1982	785500032
## 24	1983	889400000
## 25	1984	985699968
## 26	1985	1039500032
## 27	1986	1173500032
## 28	1987	1296499968
## 29	1988	1415100032
## 30	1989	1501500032
## 31	1990	1592400000
## 32	1991	1634900000
## 33	1992	1679900000
## 34	1993	1820359900
## 35	1994	1867160100
## 36	1995	2030750000
## 37	1996	2695390000
## 38	1997	2932827000
## 39	1998	3130748000

```
## 40 1999 3324433000
## 41 2000 3480219000
## 42 2001 3680483000
## 43 2002 3937228000
## 44 2003 4186525000
## 45 2004 4484703000
## 46 2005 4868136000
## 47 2006 6144000000
## 48 2007 6767000000
## 49 2008 6980000000
## 50 2009 6656000000
## 51 2010 6634526000
## 52 2011 6312691000
## 53 2012 6378188000
## 54 2013 6465756000
## 55 2014 6413988000
## 56 2015 6654541000
## 57 2016 6899911000
## 58 2017 7142316000
## 59 2018 7225977000
## 60 2019 7423465000
## 61 2020 6887147000
## 62 2021 7286607000
## 63 2022 7827980000
```

```
Tahun<-as.numeric(DataW2$Tahun)
PDB<-DataW2$PDB
```

Data yang digunakan adalah data nilai PDB negara Bermuda dalam rentang tahun 1960-2022(full)

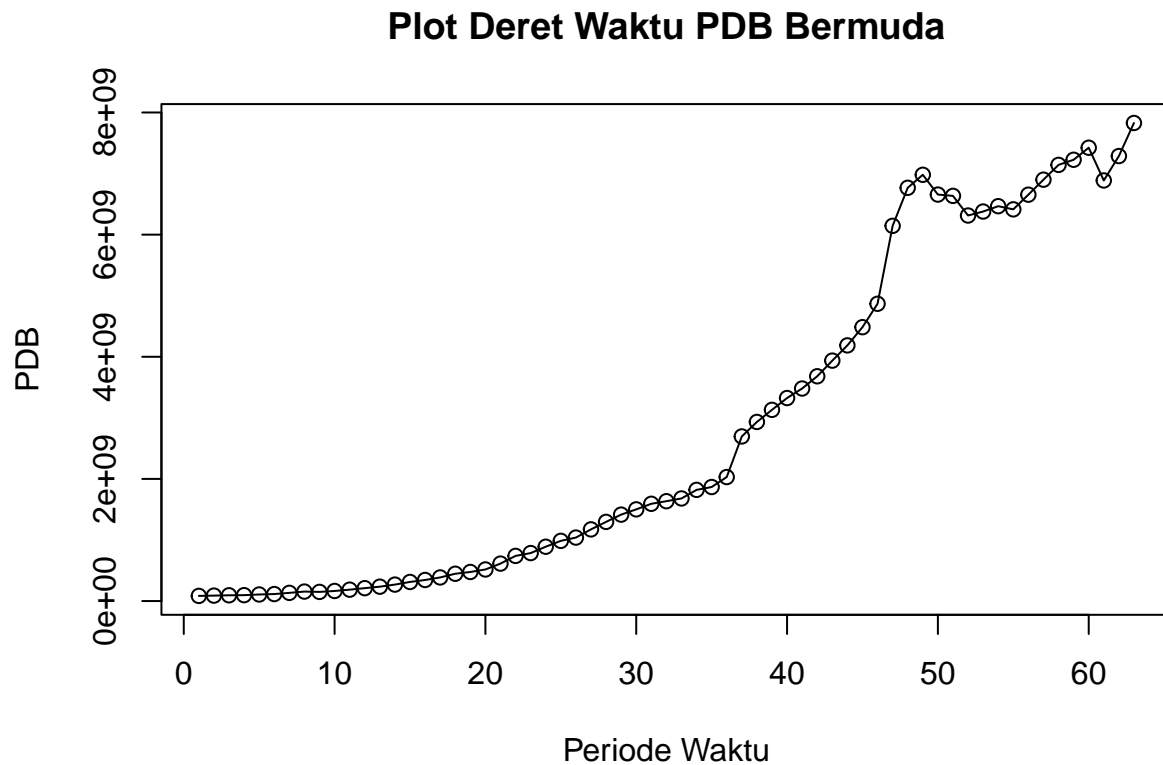
Transformasi data PDB menjadi *time series*

```
DataW2ts<-ts(DataW2$PDB)
DataW2ts
```

```
## Time Series:
## Start = 1
## End = 63
## Frequency = 1
## [1] 84466653 89249985 94149984 96366651 107566649 114339050
## [7] 134173375 155102980 150000000 164900000 186300000 211100000
## [13] 235400000 269500000 312600000 345000000 386300000 447000000
## [19] 475800000 517200000 613299968 739100032 785500032 889400000
## [25] 985699968 1039500032 1173500032 1296499968 1415100032 1501500032
## [31] 1592400000 1634900000 1679900000 1820359900 1867160100 2030750000
## [37] 2695390000 2932827000 3130748000 3324433000 3480219000 3680483000
## [43] 3937228000 4186525000 4484703000 4868136000 6144000000 6767000000
## [49] 6980000000 6656000000 6634526000 6312691000 6378188000 6465756000
## [55] 6413988000 6654541000 6899911000 7142316000 7225977000 7423465000
## [61] 6887147000 7286607000 7827980000
```

Plot awal data PDB

```
ts.plot(DataW2ts, xlab="Periode Waktu", ylab="PDB", main= "Plot Deret Waktu PDB Bermuda")  
points(DataW2ts)
```



Peramalan menggunakan DMA

```
datasma <- SMA(DataW2ts, n=3)  
dma <- SMA(datasma, n = 3)  
At <- 2*datasma - dma  
Bt <- 2/(3-1)*(datasma - dma)  
datadma<- At+Bt  
dmaramal<- c(NA, datadma)  
  
t = 1:5  
f = c()  
  
for (i in t) {  
  f[i] = At[length(At)] + Bt[length(Bt)]*(i)  
}
```

Plot DMA

```

datagab <- cbind(
  aktual = c(DataW2ts, rep(NA, 5)),
  pemulusan1 = c(datasma, rep(NA, 5)),
  pemulusan2 = c(datadma, rep(NA, 5)),
  At = c(At, rep(NA, 5)),
  Bt = c(Bt, rep(NA, 5)),
  ramalan = c(dmaramal, f[-1])
)
datagab

```

##		aktual	pemulusan1	pemulusan2	At	Bt	ramalan
##	[1,]	84466653	NA	NA	NA	NA	NA
##	[2,]	89249985	NA	NA	NA	NA	NA
##	[3,]	94149984	89288874	NA	NA	NA	NA
##	[4,]	96366651	93255540	NA	NA	NA	NA
##	[5,]	107566649	99361095	110146278	104753686	5392592	NA
##	[6,]	114339050	106090783	119134071	112612427	6521644	110146278
##	[7,]	134173375	118693025	139982473	129337749	10644724	119134071
##	[8,]	155102980	134538469	164067221	149302845	14764376	139982473
##	[9,]	150000000	146425452	172838392	159631922	13206470	164067221
##	[10,]	164900000	156667660	178248593	167458127	10790467	172838392
##	[11,]	186300000	167066667	187760148	177413407	10346740	178248593
##	[12,]	211100000	187433333	221521560	204477447	17044113	187760148
##	[13,]	235400000	210933333	255844444	233388889	22455556	221521560
##	[14,]	269500000	238666667	291311111	264988889	26322222	255844444
##	[15,]	312600000	272500000	336100000	304300000	31800000	291311111
##	[16,]	345000000	309033333	380300000	344666667	35633333	336100000
##	[17,]	386300000	347966667	424233333	386100000	38133333	380300000
##	[18,]	447000000	392766667	478455556	435611111	42844444	424233333
##	[19,]	475800000	436366667	524366667	480366667	44000000	478455556
##	[20,]	517200000	480000000	567244444	523622222	43622222	524366667
##	[21,]	613299968	535433323	638433308	586933316	51499993	567244444
##	[22,]	739100032	623200000	777177785	700188892	76988892	638433308
##	[23,]	785500032	712633344	890388921	801511132	88877788	777177785
##	[24,]	889400000	804666688	987000043	895833365	91166677	890388921
##	[25,]	985699968	886866667	1057822201	972344434	85477767	987000043
##	[26,]	1039500032	971533333	1139222208	1055377771	83844437	1057822201
##	[27,]	1173500032	1066233344	1248944469	1157588907	91355563	1139222208
##	[28,]	1296499968	1169833344	1371100018	1270466681	100633337	1248944469
##	[29,]	1415100032	1295033344	1531033344	1413033344	118000000	1371100018
##	[30,]	1501500032	1404366677	1633611122	1518988900	114622222	1531033344
##	[31,]	1592400000	1503000021	1707400036	1605200028	102200007	1633611122
##	[32,]	1634900000	1576266677	1739711115	1657988896	81722219	1707400036
##	[33,]	1679900000	1635733333	1763866645	1699799989	64066656	1739711115
##	[34,]	1820359900	1711719967	1852679915	1782199941	70479974	1763866645
##	[35,]	1867160100	1789140000	1943024467	1866082233	76942233	1852679915
##	[36,]	2030750000	1906090000	2113636689	2009863344	103773344	1943024467
##	[37,]	2695390000	2197766700	2664635633	2431201167	233434467	2113636689
##	[38,]	2932827000	2552989000	3221069867	2887029433	334040433	2664635633
##	[39,]	3130748000	2919655000	3645357867	3282506433	362851433	3221069867
##	[40,]	3324433000	3129336000	3653354667	3391345333	262009333	3645357867
##	[41,]	3480219000	3311800000	3694872667	3503336333	191536333	3653354667
##	[42,]	3680483000	3495045000	3861014333	3678029667	182984667	3694872667

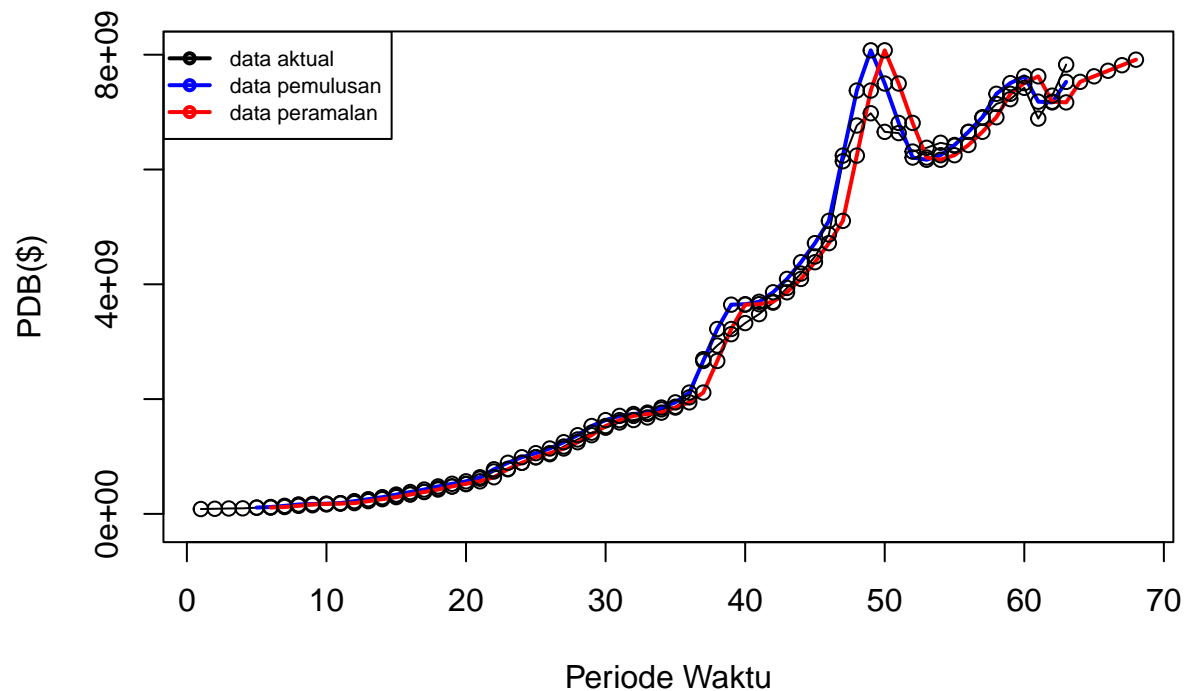
```
## [43,] 3937228000 3699310000 4093826667 3896568333 197258333 3861014333
## [44,] 4186525000 3934745333 4384835778 4159790556 225045222 4093826667
## [45,] 4484703000 4202818667 4717206667 4460012667 257194000 4384835778
## [46,] 4868136000 4513121333 5105573778 4809347556 296226222 4717206667
## [47,] 6144000000 5165613000 6242470333 5704041667 538428667 5105573778
## [48,] 6767000000 5926378667 7375727333 6651053000 724674333 6242470333
## [49,] 6980000000 6630333333 8076116667 7353225000 722891667 7375727333
## [50,] 6656000000 6801000000 7497858667 7149429333 348429333 8076116667
## [51,] 6634526000 6756842000 6811742444 6784292222 27450222 7497858667
## [52,] 6312691000 6534405667 6208385222 6371395444 -163010222 6811742444
## [53,] 6378188000 6441801667 6170038778 6305920222 -135881444 6208385222
## [54,] 6465756000 6385545000 6248800111 6317172556 -68372444 6170038778
## [55,] 6413988000 6419310667 6426827111 6423068889 3758222 6248800111
## [56,] 6654541000 6511428333 6656762333 6584095333 72667000 6426827111
## [57,] 6899911000 6656146667 6910516222 6783331444 127184778 6656762333
## [58,] 7142316000 6898922667 7319102889 7109012778 210090111 6910516222
## [59,] 7225977000 7089401333 7505223556 7297312444 207911111 7319102889
## [60,] 7423465000 7263919333 7623595778 7443757556 179838222 7505223556
## [61,] 6887147000 7178863000 7181799889 7180331444 1468444 7623595778
## [62,] 7286607000 7199073000 7169315444 7184194222 -14878778 7181799889
## [63,] 7827980000 7333911333 7527169111 7430540222 96628889 7169315444
## [64,] NA NA NA NA NA 7527169111
## [65,] NA NA NA NA NA 7623798000
## [66,] NA NA NA NA NA 7720426889
## [67,] NA NA NA NA NA 7817055778
## [68,] NA NA NA NA NA 7913684667
```

```
ts.plot(datagab[, 1], xlab = "Periode Waktu", ylab = "PDB($)",
        main = "DMA n=3 Data PDB", ylim = c(min(datagab, na.rm = TRUE), max(datagab, na.rm = TRUE)))

lines(datagab[, 3], col = "blue", lwd = 2)
lines(datagab[, 6], col = "red", lwd = 2)
points(datagab[, 1], col = "black", pch = 21, bg = NA)
points(datagab[, 3], col = "black", pch = 21, bg = NA)
points(datagab[, 6], col = "black", pch = 21, bg = NA)

legend("topleft", c("data aktual", "data pemulusan", "data peramalan"),
      col = c("black", "blue", "red"), lty = c(1, 1, 1), lwd = 2, pch = 21, pt.bg = NA, cex = 0.7)
```

DMA n=3 Data PDB



Menghitung akurasi DMA

```
error.dma = DataW2ts-dmaramal[1:length(DataW2ts)]
SSE.dma = sum(error.dma[6:length(DataW2ts)]^2)
MSE.dma = mean(error.dma[6:length(DataW2ts)]^2)
MAPE.dma = mean(abs((error.dma[6:length(DataW2ts)]/DataW2ts[6:length(DataW2ts)]))*100))

akurasi.dma <- matrix(c(SSE.dma, MSE.dma, MAPE.dma))
row.names(akurasi.dma) <- c("SSE", "MSE", "MAPE")
colnames(akurasi.dma) <- c("Akurasi m = 3")
akurasi.dma
```

```
##      Akurasi m = 3
## SSE   6.474255e+18
## MSE   1.116251e+17
## MAPE   5.369022e+00
```

Peramalan menggunakan DES

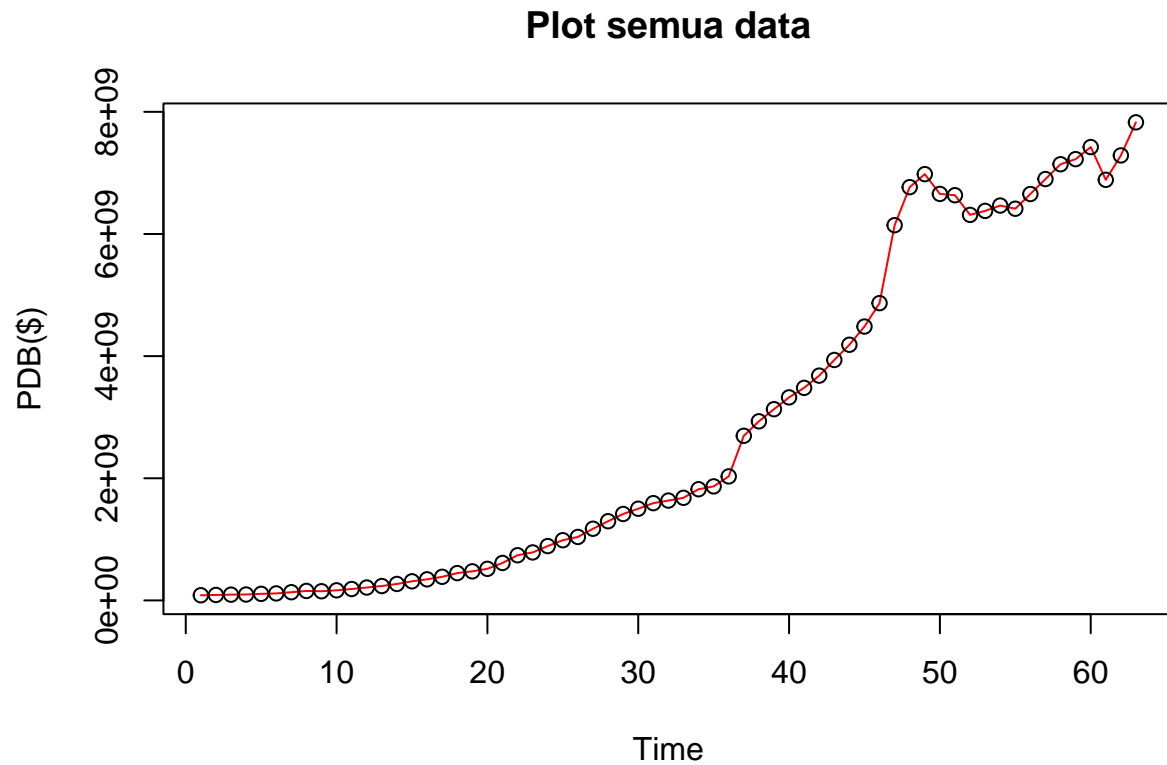
Pembagian data latih dan uji untuk metode DES

```
DataW2ts.lat <- unlist(DataW2[1:50,2])
DataW2ts.uji <- unlist(DataW2[51:63,2])

latw2ts <- ts(DataW2ts.lat)
```

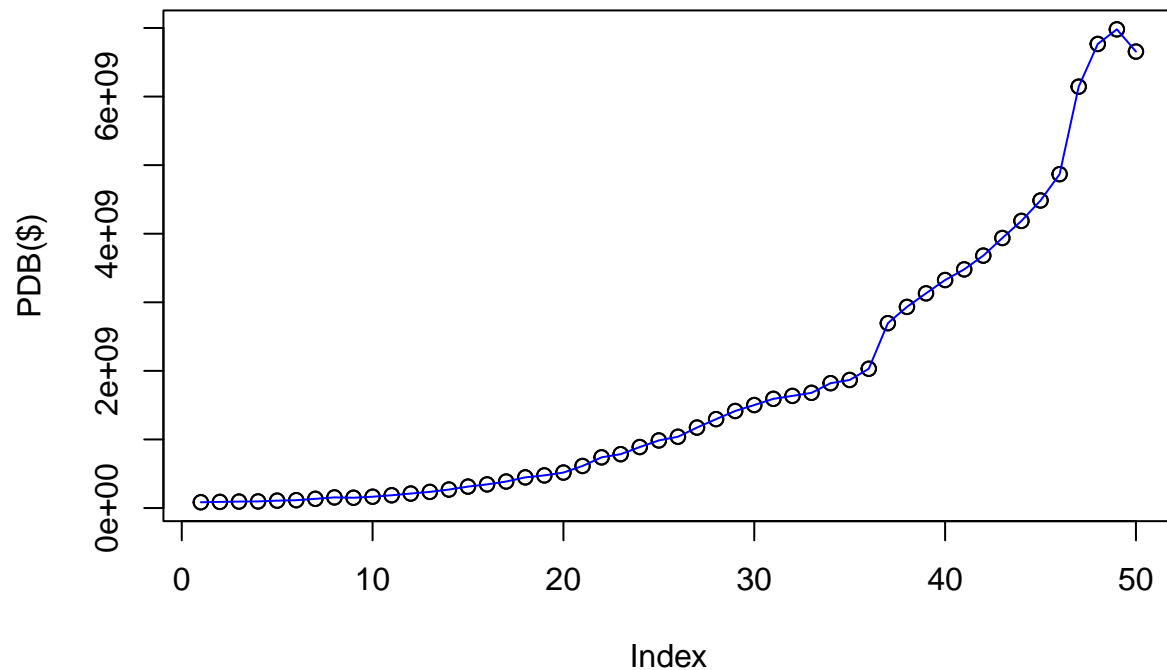
```
uji2w2ts<-ts(DataW2ts.uji,start=51)

plot(DataW2ts,ylab="PDB($)", col="red",main="Plot semua data")
points(DataW2ts)
```



```
plot(DataW2ts.lat,ylab="PDB($)", main = "Plot data latihan", type = "p")
points(DataW2ts.lat, col="black")
lines(DataW2ts.lat,col="blue")
```


Plot data latih



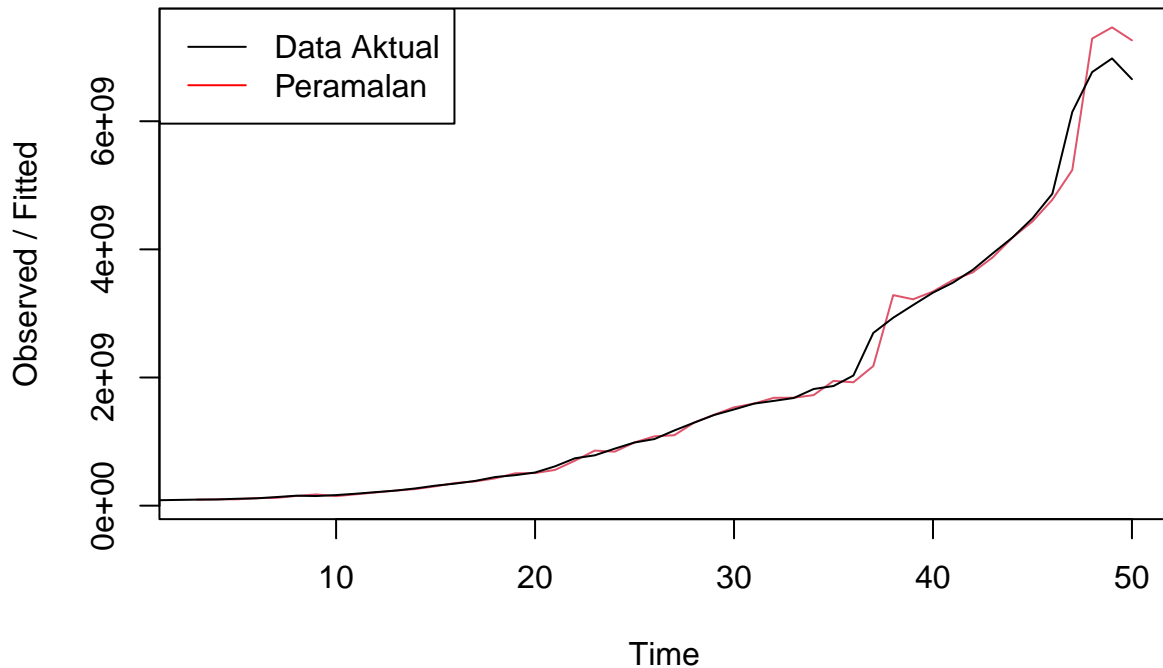
Pemulusan DES

```
desopt<- HoltWinters(DataW2ts.lat, gamma = FALSE)
desopt
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = DataW2ts.lat, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0.8562485
##  gamma: FALSE
##
## Coefficients:
##      [,1]
## a 6656000000
## b -236780389
```

```
plot(desopt)
legend("topleft", c("Data Aktual", "Peramalan"), col = c("black", "red"),
      lty = c(1,1))
```

Holt-Winters filtering



```
ramaldesopt<- forecast(desopt, h=13)
ramaldesopt
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 51	6419219611	6144157979	6694281243	5998549146	6839890076
## 52	6182439222	5602479191	6762399254	5295466895	7069411549
## 53	5945658833	5000659047	6890658620	4500406378	7390911289
## 54	5708878444	4346302882	7071454007	3624998950	7792757938
## 55	5472098055	3645063101	7299133009	2677889219	8266306892
## 56	5235317667	2901158575	7569476758	1665529391	8805105942
## 57	4998537278	2117864028	7879210528	592927724	9404146832
## 58	4761756889	1297808818	8225704959	-535894525	10059408302
## 59	4524976500	443161629	8606791370	-1717620640	10767573640
## 60	4288196111	-444250187	9020642409	-2949455930	11525848152
## 61	4051415722	-1362860094	9465691538	-4229004580	12331836024
## 62	3814635333	-2311305978	9940576645	-5554183421	13183454087
## 63	3577854944	-3288389399	10444099287	-6923159594	14078869483

Akurasi DES data latih

```
ssedes.train<-desopt$SSE
msedes.train<-ssedes.train/length(latw2ts)
sisaandes<-ramaldesopt$residuals
head(sisaandes)
```

```
## Time Series:
```

```
## Start = 1
## End = 6
## Frequency = 1
## [1]          NA          NA    116666.6 -2666561.9  8600009.6 -3191332.3

mapedes.train <- sum(abs(sisaandes[3:length(latw2ts)]/latw2ts[3:length(latw2ts)])*100)/length(latw2ts)

akurasides.opt <- matrix(c(ssedes.train,msedes.train,mapedes.train))
row.names(akurasides.opt)<- c("SSE", "MSE", "MAPE")
colnames(akurasides.opt) <- c("Akurasi lamda dan gamma optimum")
akurasides.opt

##      Akurasi lamda dan gamma optimum
## SSE                2.166797e+18
## MSE                4.333594e+16
## MAPE                4.452996e+00
```

Akurasi DES data uji

```
selisihdesopt<-ramalidesopt$mean-uji2w2ts

SSEtestingdesopt<-sum(selisihdesopt^2)
SSEtestingdesopt<-SSEtestingdesopt/length(uji2w2ts)
MAPEtestingdesopt<-sum(abs(selisihdesopt/uji2w2ts)*100)/length(uji2w2ts)

akurasiDesTesting <- matrix(c(SSEtestingdesopt,SSEtestingdesopt,MAPEtestingdesopt))
row.names(akurasiDesTesting)<- c("SSE", "MSE", "MAPE")
colnames(akurasiDesTesting) <- c("Akurasi lamda dan gamma optimum")
akurasiDesTesting

##      Akurasi lamda dan gamma optimum
## SSE                5.253211e+18
## MSE                5.253211e+18
## MAPE                2.641734e+01
```

Perbandingan akurasi DMA dengan DES

```
cbind(akurasi.dma, akurasides.opt)

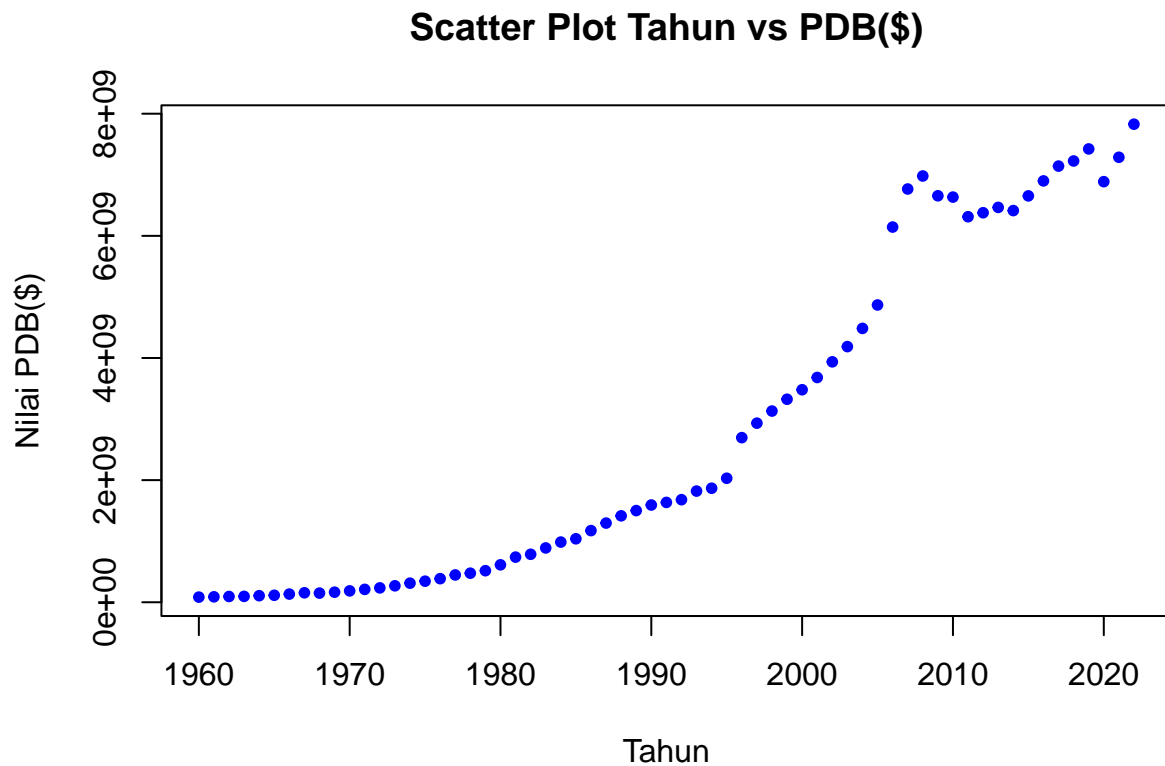
##      Akurasi m = 3 Akurasi lamda dan gamma optimum
## SSE    6.474255e+18                2.166797e+18
## MSE    1.116251e+17                4.333594e+16
## MAPE   5.369022e+00                4.452996e+00
```

Pemulusan dilakukan pada data PDB negara Bermuda dari rentang tahun 1960-2022(full). Berdasarkan nilai akurasi SSE, MSE, dan MAPE, metode DES memiliki nilai-nilai akurasi yang lebih kecil daripada metode DMA. Hal ini membuat metode DES lebih baik daripada metode DMA.

Eksplorasi Data

```
plot(DataW2, pch = 20, col = "blue",
     main = "Scatter Plot Tahun vs PDB($)",
```

```
xlab = "Tahun",
ylab = "Nilai PDB($)")
```



```
#Menampilkan Nilai Korelasi
cor(Tahun, DataW2$PDB)
```

```
## [1] 0.9474186
```

Berdasarkan plot antara tahun dengan nilai pdb, dapat kita lihat bahwa terdapat hubungan positif yang sangat erat. Hal ini juga diperkuat melalui perhitungan korelasi antara peubah tahun dengan nilai pdb, dengan nilai korelasi sebesar *0.9474186*. Nilai ini menunjukkan hubungan positif yang sangat erat.

Model regresi

```
modelreg<- lm(PDB~Tahun, data = DataW2)
summary(modelreg)
```

```
##
## Call:
## lm(formula = PDB ~ Tahun, data = DataW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.389e+09 -7.179e+08 -3.650e+07  6.315e+08  1.749e+09
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.780e+11  1.215e+10  -22.89  <2e-16 ***
## Tahun      1.411e+08   6.100e+06   23.12  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 880500000 on 61 degrees of freedom
## Multiple R-squared:  0.8976, Adjusted R-squared:  0.8959
## F-statistic: 534.7 on 1 and 61 DF,  p-value: < 2.2e-16
```

Model yang dihasilkan adalah

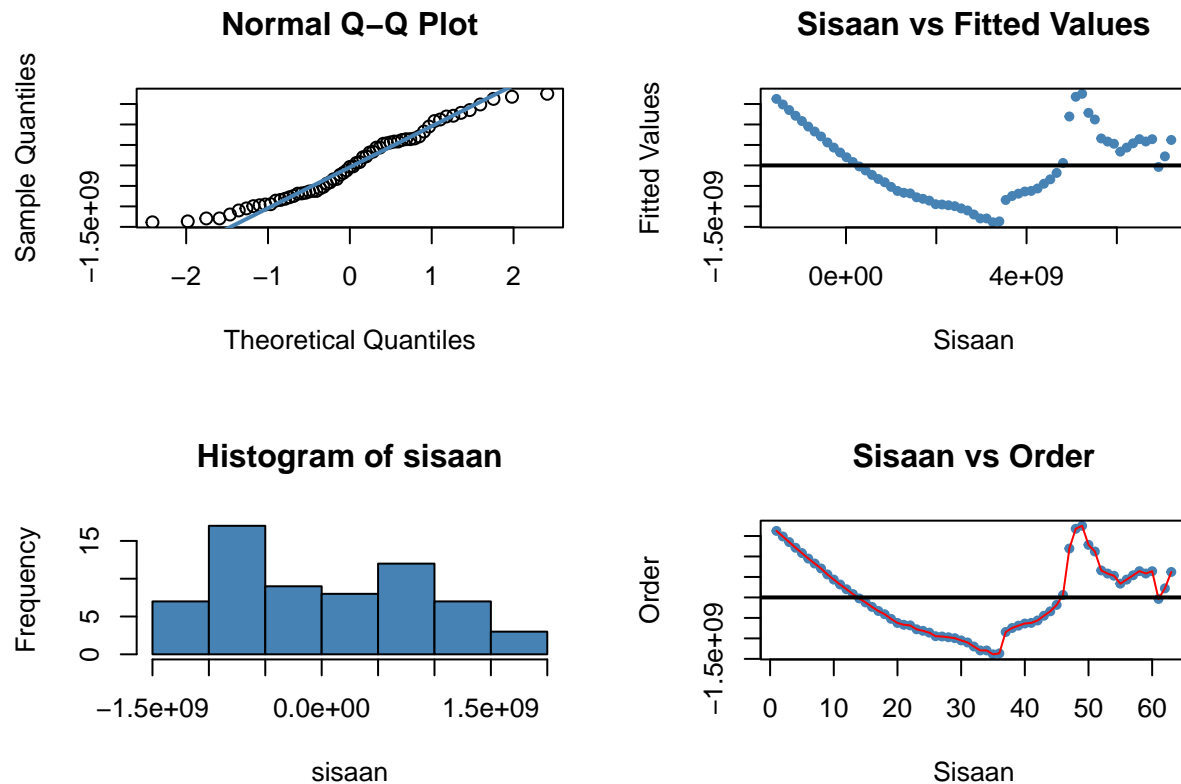
$$y_i = -278030155234 + 141066240x_t$$

. Dari hasil ringkasan model, uji F model memiliki $p\text{-value} < \alpha$ (5%), artinya, terdapat minimal satu variabel yang berpengaruh signifikan terhadap model. Hasil uji-t parsial kedua parameter regresi, yaitu intersep dan koefisien regresi juga menunjukkan hal yang sama, yaitu memiliki $p\text{-value} < \alpha$ (5%) sehingga nyata dalam taraf 5%. Selanjutnya dapat dilihat juga nilai $R^2 = 0.8976$. Artinya, sebesar 89.76% keragaman nilai PDB dapat dijelaskan oleh peubah tahun. Selanjutnya akan dilakukan pengujian terhadap sisaan.

Pengujian sisaan model

```
sisaan<- residuals(modelreg)
fitValue<- predict(modelreg)

par(mfrow = c(2,2))
qqnorm(sisaan)
qqline(sisaan, col = "steelblue", lwd = 2)
plot(fitValue, sisaan, col = "steelblue", pch = 20, xlab = "Sisaan", ylab = "Fitted Values", main = "Sisaan v Fitted Values")
abline(a = 0, b = 0, lwd = 2)
hist(sisaan, col = "steelblue")
plot(seq(1,63,1), sisaan, col = "steelblue", pch = 20, xlab = "Sisaan", ylab = "Order", main = "Sisaan v Order")
lines(seq(1,63,1), sisaan, col = "red")
abline(a = 0, b = 0, lwd = 2)
```



Normal QQ plot di atas menunjukkan bahwa sisaan cenderung menyebar normal, begitu juga dengan histogram sisaan. Plot Sisaan vs *Fitted Value* dan Plot Sisaan vs *Order* menunjukkan adanya pola pada sisaan. Selanjutnya akan dilakukan pengujian formal untuk melihat normalitas sisaan dan plot ACF serta PACF untuk melihat apakah terdapat autokorelasi atau tidak.

Uji formal

Uji normalitas

```
shapiro.test(sisaan)

##
##  Shapiro-Wilk normality test
##
## data:  sisaan
## W = 0.95621, p-value = 0.02505

ks.test(sisaan, "pnorm", mean=mean(sisaan), sd=sd(sisaan))

##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  sisaan
## D = 0.11107, p-value = 0.3901
## alternative hypothesis: two-sided
```

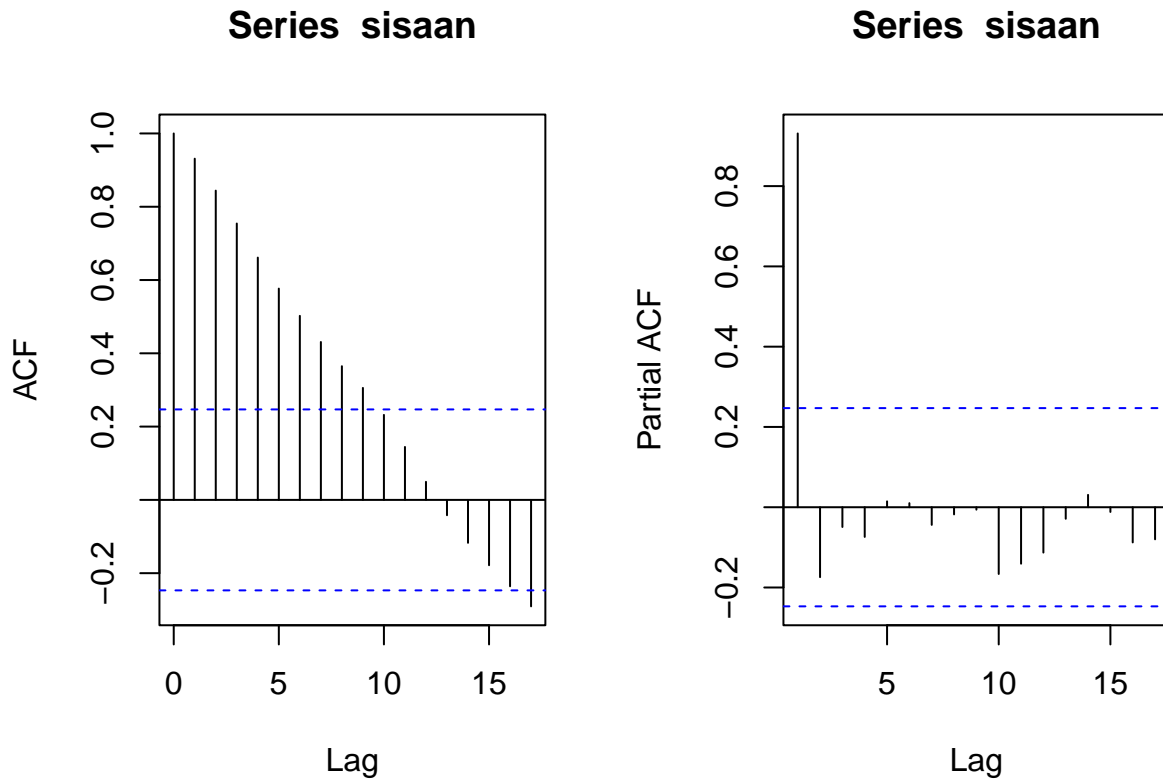
H0: sisaan mengikuti sebaran normal, H1: sisaan tidak mengikuti sebaran normal.

Berdasarkan uji normalitas *Kolmogorov Smirnov*, diperoleh nilai p uji *Kolmogorov Smirnov* sebesar 0.3901,

yang mana lebih besar daripada $\alpha = 5\%$, sehingga kesimpulan yang didapat adalah tak tolak H_0 , artinya sisaan mengikuti sebaran normal

Uji ACF dan PACF

```
par(mfrow = c(1,2))
acf(sisaan)
pacf(sisaan)
```



Pada plot ACF, terdapat lag yang melebihi batas, yakni lag ke 1 sampai 9. Pada plot PACF tidak terdapat lag yang signifikan. Namun, untuk lebih memastikan akan dilakukan uji formal dengan uji Durbin Watson.

Uji Durbin Watson

```
dwtest(modelreg)
```

```
##
## Durbin-Watson test
##
## data: modelreg
## DW = 0.073605, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

H_0 : tidak ada autokorelasi, H_1 : ada autokorelasi.

Berdasarkan uji Durbin Watson yang telah dilakukan, diperoleh nilai p sebesar $2.2e-16 < \alpha = 5\%$. Kesimpulan yang dapat diambil adalah pada taraf nyata 5%, tolak H_0 , artinya terdapat autokorelasi, sehingga diperlukan

penanganan untuk autokorelasi tersebut. Penanganan autokorelasi yang akan dilakukan adalah metode Cochrane-Orcutt dan Hildret-Lu.

Penanganan Autokorelasi

Metode Cochrane-Orcutt

```
modelCO<-cochrane.orcutt(modelreg)
modelCO

## Cochrane-orcutt estimation for first order autocorrelation
##
## Call:
## lm(formula = PDB ~ Tahun, data = DataW2)
##
## number of interaction: 44
## rho 0.940121
##
## Durbin-Watson statistic
## (original): 0.07361 , p-value: 2.389e-40
## (transformed): 1.27215 , p-value: 9.071e-04
##
## coefficients:
## (Intercept)      Tahun
## -395823454080    199612066

rho<-modelCO$rho
```

Model yang didapat dari metode Cochrane-Orcutt adalah

$$y_i = -395823454080 + 199612066x_t$$

. Metode Cocharane-Orcutt juga menunjukkan bahwa nilai DW dan p-value meningkat menjadi 1.27215 dan 0.0009071. Namun nilai DW tidak berada pada rentang $DU < DW < 4-DU$, melainkan masih berada dalam rentang $DW < DL$ atau $1.27215 < 1.5599$. Nilai p juga masih lebih kecil daripada taraf nyata $\alpha = 5\%$. Artinya, masih tolak H_0 , yang mana masih terdapat autokorelasi pada taraf nyata 5%.

Transformasi data

```
PDB

## [1] 84466653 89249985 94149984 96366651 107566649 114339050
## [7] 134173375 155102980 150000000 164900000 186300000 211100000
## [13] 235400000 269500000 312600000 345000000 386300000 447000000
## [19] 475800000 517200000 613299968 739100032 785500032 889400000
## [25] 985699968 1039500032 1173500032 1296499968 1415100032 1501500032
## [31] 1592400000 1634900000 1679900000 1820359900 1867160100 2030750000
## [37] 2695390000 2932827000 3130748000 3324433000 3480219000 3680483000
## [43] 3937228000 4186525000 4484703000 4868136000 6144000000 6767000000
## [49] 6980000000 6656000000 6634526000 6312691000 6378188000 6465756000
## [55] 6413988000 6654541000 6899911000 7142316000 7225977000 7423465000
## [61] 6887147000 7286607000 7827980000

PDB[-1]
```

```
## [1] 89249985 94149984 96366651 107566649 114339050 134173375
```



```
## [7] 155102980 150000000 164900000 186300000 211100000 235400000
## [13] 269500000 312600000 345000000 386300000 447000000 475800000
## [19] 517200000 613299968 739100032 785500032 889400000 985699968
## [25] 1039500032 1173500032 1296499968 1415100032 1501500032 1592400000
## [31] 1634900000 1679900000 1820359900 1867160100 2030750000 2695390000
## [37] 2932827000 3130748000 3324433000 3480219000 3680483000 3937228000
## [43] 4186525000 4484703000 4868136000 6144000000 6767000000 6980000000
## [49] 6656000000 6634526000 6312691000 6378188000 6465756000 6413988000
## [55] 6654541000 6899911000 7142316000 7225977000 7423465000 6887147000
## [61] 7286607000 7827980000
```

```
PDBtrans<- PDB[-1]-PDB[-63]*rho
Tahuntrans<- Tahun[-1]-Tahun[-63]*rho
modelCManual<- lm(PDBtrans~Tahuntrans)
summary(modelCManual)
```

```
##
## Call:
## lm(formula = PDBtrans ~ Tahuntrans)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -722148438 -72173368 -20025238  55490084 1104358984
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.370e+10  3.209e+09  -7.387 5.47e-10 ***
## Tahuntrans   1.996e+08  2.670e+07   7.477 3.83e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225300000 on 60 degrees of freedom
## Multiple R-squared:  0.4823, Adjusted R-squared:  0.4737
## F-statistic: 55.91 on 1 and 60 DF,  p-value: 3.832e-10
```

Hasil transformasi ini bukanlah merupakan model sesungguhnya. Koefisien regresi masih perlu dicari kembali mengikuti $\beta_0^* = \beta_0 + \rho\beta_1$ dan $\beta_1^* = \beta_1$.

```
b0bintang <- modelCManual$coefficients[-2]
b0 <- b0bintang/(1-rho)
b1 <- modelCManual$coefficients[-1]
b0
```

```
##      (Intercept)
## -395823454080
b1
```

```
## Tahuntrans
## 199612066
```

Hasil β_0 dan β_1 sudah sama dengan jika menggunakan fungsi `cochrane.orcutt(modelreg)`.

Metode Hildreth-Lu

```
HiLu.func<- function(r, modelreg){
  x <- model.matrix(modelreg)[,-1]
```

```

y <- model.response(model.frame(modelreg))
n <- length(y)
t <- 2:n
y <- y[t]-r*y[t-1]
x <- x[t]-r*x[t-1]

return(lm(y~x))
}

r <- c(seq(0.1,1, by= 0.1))
tab <- data.frame("rho" = r, "SSE" = sapply(r, function(i){deviance(HiLu.func(i, modelreg))}))
round(tab, 4)

```

```

##      rho      SSE
## 1  0.1 3.613161e+19
## 2  0.2 2.872360e+19
## 3  0.3 2.225317e+19
## 4  0.4 1.672033e+19
## 5  0.5 1.212505e+19
## 6  0.6 8.467362e+18
## 7  0.7 5.747248e+18
## 8  0.8 3.964713e+18
## 9  0.9 3.119757e+18
## 10 1.0 3.464694e+18

```

Pada $\rho = 0.9$, didapatkan nilai SSE yang paling minimum. Selanjutnya akan dilakukan pencarian nilai ρ dengan jarak yang lebih kecil yang dapat lebih meminimumkan nilai SSE. Pencarian nilai ρ yang diduga lebih meminimumkan SSE akan dilakukan dalam selang 0.9 sampai 1 dengan jarak 0.001

```

rOpt <- seq(0.9,1.5, by= 0.001)
tabOpt <- data.frame("rho" = rOpt, "SSE" = sapply(rOpt, function(i){deviance(HiLu.func(i, modelreg))}))
head(tabOpt[order(tabOpt$SSE),])

```

```

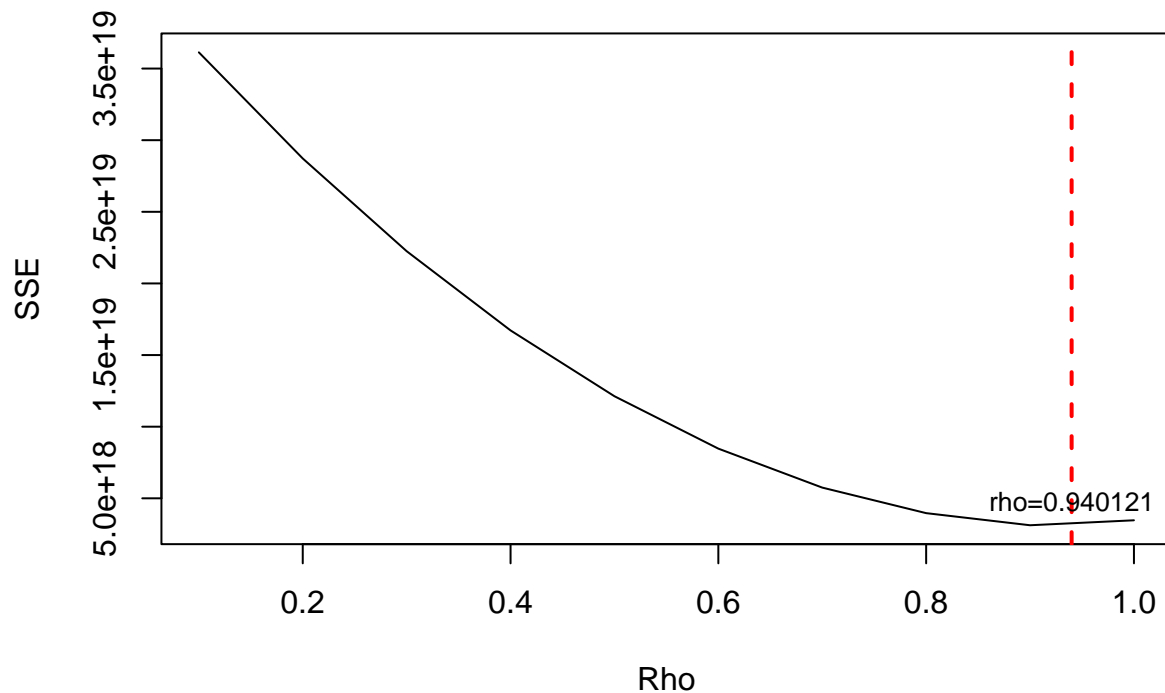
##      rho      SSE
## 41 0.940 3.044296e+18
## 42 0.941 3.044332e+18
## 40 0.939 3.044355e+18
## 43 0.942 3.044461e+18
## 39 0.938 3.044507e+18
## 44 0.943 3.044684e+18

```

```

#Grafik SSE optimum
par(mfrow = c(1,1))
plot(tab$SSE ~ tab$rho , type = "l", xlab = "Rho", ylab = "SSE")
abline(v = tabOpt[tabOpt$SSE==min(tabOpt$SSE),"rho"], lty = 2, col="red",lwd=2)
text(x=0.940121, y=480000000000000000, labels = "rho=0.940121", cex = 0.8)

```



Perhitungan yang dilakukan aplikasi R menunjukkan bahwa nilai ρ optimum, yaitu saat SSE terkecil terdapat pada nilai $\rho = 0.9401$.

```
modelHiLu <- HiLu.func(0.9401, modelreg)
summary(modelHiLu)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -722134105 -72182695 -20022439  55510975 1104360775
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.371e+10  3.209e+09  -7.388 5.43e-10 ***
## x            1.996e+08  2.669e+07   7.479 3.80e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225300000 on 60 degrees of freedom
## Multiple R-squared:  0.4825, Adjusted R-squared:  0.4738
## F-statistic: 55.93 on 1 and 60 DF, p-value: 3.804e-10
```

```
#Transformasi Balik
```

```
cat("y = ", coef(modelHiLu)[1]/(1-0.9401), "+", coef(modelHiLu)[2], "x", sep = "")
```

```
## y = -395780908856+199591088x
```

Setelah melakukan transformasi balik, didapatkan model dengan metode Hildreth-Lu :

$$y_i = -395780908856 + 199591088x_t$$

Uji Durbin-Watson model Hildreth-Lu

```
dwtest(modelHiLu)
```

```
##
## Durbin-Watson test
##
## data: modelHiLu
## DW = 1.2721, p-value = 0.0009068
## alternative hypothesis: true autocorrelation is greater than 0
```

Hasil uji Durbin-Watson juga menunjukkan bahwa nilai DW sebesar 1.2721 berada pada selang daerah yang ada autokorelasi, yaitu pada rentang $DW < DL$ atau $1.27215 < 1.5599$. Hal tersebut juga didukung oleh *p-value* sebesar 0.0009068, di mana $p\text{-value} < \alpha = 5\%$. Artinya tolak H_0 , terdapat cukup bukti untuk menyatakan bahwa ada autokorelasi dalam data nilai PDB dengan metode Hildreth-Lu pada taraf nyata 5%.

Perbandingan SSE model awal, model Cochrane-Orcutt, dan model Hildreth-Lu

```
#Perbandingan
sseModelawal <- anova(modelreg)$`Sum Sq`[-1]
sseModelCO <- anova(modelCOmanual)$`Sum Sq`[-1]
sseModelHL <- anova(modelHiLu)$`Sum Sq`[-1]
mseModelawal <- sseModelawal/length(PDB)
mseModelCO <- sseModelCO/length(PDB)
mseModelHL <- sseModelHL/length(PDB)
akurasi <- matrix(c(sseModelawal,sseModelCO,sseModelHL,
                    mseModelawal,mseModelCO,mseModelHL),nrow=2,ncol=3,byrow = T)
colnames(akurasi) <- c("Model Awal", "Model Cochrane-Orcutt", "Model Hildreth-Lu")
row.names(akurasi) <- c("SSE","MSE")
akurasi
```

```
##      Model Awal Model Cochrane-Orcutt Model Hildreth-Lu
## SSE 4.729165e+19      3.044296e+18      3.044296e+18
## MSE 7.506611e+17      4.832215e+16      4.832215e+16
```

Dari akurasi SSE pada semua model, model Cochrane-Orcutt dan model Hildreth-Lu memiliki nilai SSE dan MSE yang sama persis, dan sama-sama lebih kecil daripada model awal. Hal ini menunjukkan bahwa model Cochrane-Orcutt dan model Hildreth-Lu lebih baik daripada model awal.

Kesimpulan

Penanganan autokorelasi pada data awal (rentang tahun 1960-2022). Penanganan dilakukan dengan metode Cochrane-Orcutt dan Hildreth-Lu. Model Cochrane-Orcutt dan model Hildreth-Lu memiliki nilai SSE dan MSE yang lebih kecil daripada model awal. Akan tetapi kedua metode masih belum bisa menangani autokorelasi pada data PDB negara Bermuda. Perlu dilakukan metode lain untuk menangani autokorelasi pada data PDB negara Bermuda.