PROGRAM STUDI STATISTIKA
SK BAN-PT No. 1765/SK/BAN-PT/AK-PPJ/S/III/2022
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PGRI ADI BUANA SURABAYA

FORM F.SK05

BUKTI BIMBINGAN SKRIPSI

Nama Mahasiswa : Eka Mauliana Sari

NIM : 192409020

Judul Skripsi : Analisis Peramalan untuk Memprediksi Harga Cabai Rawit di Provinsi Jawa Timur

Dosen Pembimbing : Alfisyahrina Hapsery, S.Si., M.Si

Materi Pembimbingan Proposal	Tanda Tangan Dosen Pembimbing
1. Menentukan komoditi yang akan dilakukan analisis dan metode yang mungkin	
2. Manfaat penulisan dan perbedaan dengan penelitian sebelumnya	
3. Agar ditambahkan deteksi outlier pada bab 2	
4. Agar ditambahkan definisi cabai rawit	
5. Langkah penelitian agar dicek kembali kesesuaian	
6. Diagram alir agar direvisi	
7. Running data sementara	
8. Pengujian asumsi sementara	

Catatan: *) Coret yang tidak sesuai

Lembar ini digunakan untuk mendaftar Seminar dan Ujian Skripsi (bimbingan skripsi minimal 8 kali)

FORM F.SK08
PERBAIKAN/REVISI SEMINAR DAN UJIAN SKRIPSI

Nama Mahasiswa : Eka Mauliana Sari
NIM :192409020
Judul Skripsi : Analisis Peramalan untuk Memprediksi Harga Cabai Rawit di Provinsi Jawa Timur
Dosen Pembimbing : Alfisyahrina Hapsery

Materi Revisi Seminar dan Ujian Skripsi	Tanda Tangan Dosen Penguji
1. Statistik deskriptif untuk harga cabai rawit tertinggi dan terendah per tahun	
2. Konsistensi penulisan titik dan koma pada nominal di dalam tabel	
3. Pemilihan model terbaik menggunaan nilai MSE dan MAPE	

Surabaya, Juli 2023
Dosen Pembimbing,

Alfisyahrina Hapsery
NIP/NPP : 1804856/PY

*Catatan: *) Coret yang tidak sesuai*

PROGRAM STUDI STATISTIKA
SK BAN-PT No. 1765/SK/BAN-PT/AK-PPJ/S/III/2022
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PGRI ADI BUANA SURABAYA

FORM F.SK08

PERBAIKAN/REVISI SEMINAR DAN UJIAN SKRIPSI

Nama Mahasiswa : Eka Mauliana Sari

NIM :192409020

Judul Skripsi : Analisis Peramalan untuk Memprediksi Harga Cabai Rawit di Provinsi Jawa Timur

Dosen Pembimbing : Alfisyahrina Hapsery

Materi Revisi Seminar dan Ujian Skripsi	Tanda Tangan Dosen Penguji
1. Pemilihan model terbaik ditambahkan nilai MAPE	
2. Plot data aktual dan hasil peramalan agar diperjelas	

Surabaya, Juli 2023
Dosen Pembimbing,

Alfisyahrina Hapsery
NIP/NPP : 1804856/DY

*Catatan: *) Coret yang tidak sesuai*

LAMPIRAN

Lampiran A. Data Harga Rata-Rata Cabai Rawit Provinsi Jawa Timur

Periode	Harga (Rupiah)
Jan-15	55516,47
Feb-15	21398,21
Mar-15	27141,66
Apr-15	18577
May-15	18045,84
Jun-15	16473,53
Jul-15	33405,92
Aug-15	57322,63
Sep-15	42651,29
Oct-15	14743,76
Nov-15	13321,79
Dec-15	25264,47
Jan-16	28266,29
Feb-16	16181,08
Mar-16	39387,03
Apr-16	21429,26
May-16	18342
Jun-16	16754,62
Jul-16	31487,11
Aug-16	37859,82
Sep-16	26557,37
Oct-16	27177,05
Nov-16	44647,05
Dec-16	49640,66
Jan-17	89780,24

Feb-17	121580
Mar-17	99871,21
Apr-17	52610,61
May-17	51919,79
Jun-17	33385,76
Jul-17	35115,21
Aug-17	22622,24
Sep-17	13912,97
Oct-17	13251,05
Nov-17	16553,24
Dec-17	24537,05
Jan-18	33784,97
Feb-18	37883,05
Mar-18	47681,18
Apr-18	30947,42
May-18	23114,5
Jun-18	31353,5
Jul-18	46739,34
Aug-18	27346,29
Sep-18	16168,66
Oct-18	21162,74
Nov-18	20611,03
Dec-18	26037,89
Jan-19	25374,47
Feb-19	14838,45
Mar-19	18197,21
Apr-19	17527,34
May-19	16878,71
Jun-19	17293,29
Jul-19	57857,42

Aug-19	73625,26
Sep-19	50286,84
Oct-19	38511,76
Nov-19	30680,24
Dec-19	30957,11
Jan-20	59870,26
Feb-20	46217,37
Mar-20	35024,89
Apr-20	32568,03
May-20	17685,97
Jun-20	14845,74
Jul-20	19190,32
Aug-20	17289,76
Sep-20	15177,47
Oct-20	15999,03
Nov-20	25536,13
Dec-20	42662,39
Jan-21	66147,87
Feb-21	74674
Mar-21	103414,3
Apr-21	55393,16
May-21	42719,37
Jun-21	33977
Jul-21	52213,21
Aug-21	22808
Sep-21	16760,18
Oct-21	19090,53
Nov-21	19058,42
Dec-21	69751,47
Jan-22	39728,55

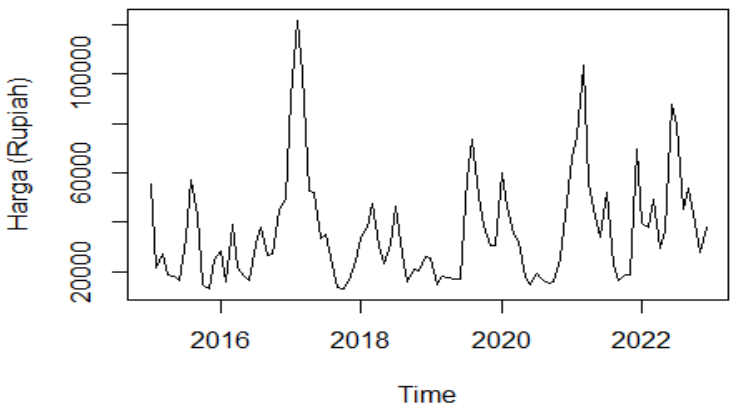
Feb-22	38067,66
Mar-22	49499,21
Apr-22	29383,42
May-22	36296,11
Jun-22	87563,45
Jul-22	79093,03
Aug-22	45164,58
Sep-22	53989,34
Oct-22	39658,5
Nov-22	27789,08
Dec-22	37892,29

Lampiran B. Output Software RStudio

```
library(tseries)

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

library(forecast)
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

library(lattice)
library(leaps)
library(ltsa)
library(bestglm)
library(FitAR)

##
## Attaching package: 'FitAR'

## The following object is masked from 'package:fo
recast':
##
##     BoxCox

library(stats)
library(EnvStats)

##
## Attaching package: 'EnvStats'
```

```
## The following objects are masked from 'package:
stats':
##
##     predict, predict.lm

## The following object is masked from 'package:ba
se':
##
##     print.default

#mengimpor data
library(readxl)
harga=read_excel("F:/cabai.xlsx")
View(harga)

#membentuk deret waktu
dataharga=ts(harga[,2], start=2015, frequency=12)
dataharga

##            Jan       Feb       Mar       Apr
May       Jun       Jul
## 2015  55516.47  21398.21  27141.66  18577.00  1
8045.84  16473.53  33405.92
## 2016  28266.29  16181.08  39387.03  21429.26  1
8342.00  16754.62  31487.11
## 2017  89780.24 121579.97  99871.21  52610.61  5
1919.79  33385.76  35115.21
## 2018  33784.97  37883.05  47681.18  30947.42  2
3114.50  31353.50  46739.34
## 2019  25374.47  14838.45  18197.21  17527.34  1
6878.71  17293.29  57857.42
## 2020  59870.26  46217.37  35024.89  32568.03  1
7685.97  14845.74  19190.32
## 2021  66147.87  74674.00 103414.34  55393.16  4
2719.37  33977.00  52213.21
## 2022  39728.55  38067.66  49499.21  29383.42  3
6296.11  87563.45  79093.03
##            Aug       Sep       Oct       Nov
Dec
## 2015  57322.63  42651.29  14743.76  13321.79  2
5264.47
```

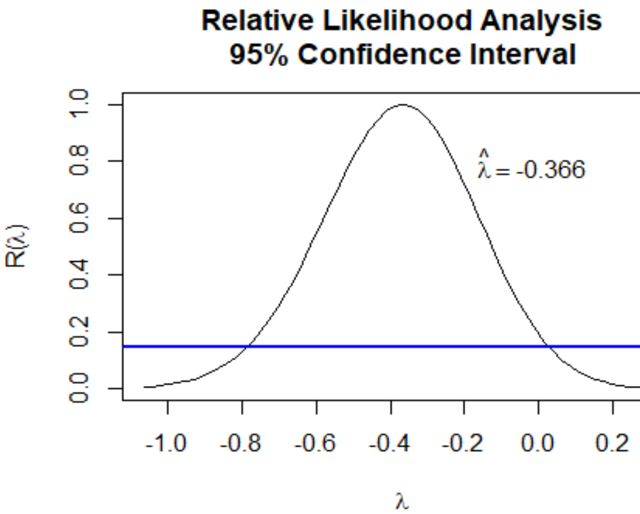
```
## 2016   37859.82   26557.37   27177.05   44647.05   4
9640.66
## 2017   22622.24   13912.97   13251.05   16553.24   2
4537.05
## 2018   27346.29   16168.66   21162.74   20611.03   2
6037.89
## 2019   73625.26   50286.84   38511.76   30680.24   3
0957.11
## 2020   17289.76   15177.47   15999.03   25536.13   4
2662.39
## 2021   22808.00   16760.18   19090.53   19058.42   6
9751.47
## 2022   45164.58   53989.34   39658.50   27789.08   3
7892.29

#membentuk plot deret waktu
ts.plot(dataharga, ylab="Harga (Rupiah)")
```




```
#menampilkan statistik deskriptif
summary(dataharga)

##       harga
##   Min.    : 13251
##   1st Qu.: 19083
##   Median : 31155
```

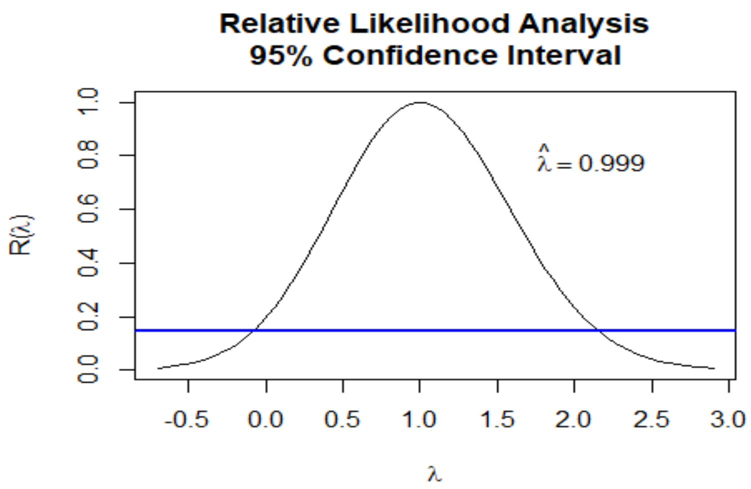
```
##   Mean    : 36602
##   3rd Qu.: 46348
##   Max.    :121580

#membagi data menjadi data training dan data testi
ng
datatraining = dataharga[1:84,]
datatraining

##   [1]  55516.47  21398.21  27141.66  18577.00  1
8045.84  16473.53  33405.92
##   [8]  57322.63  42651.29  14743.76  13321.79  2
5264.47  28266.29  16181.08
##  [15]  39387.03  21429.26  18342.00  16754.62  3
1487.11  37859.82  26557.37
##  [22]  27177.05  44647.05  49640.66  89780.24 12
1579.97  99871.21  52610.61
##  [29]  51919.79  33385.76  35115.21  22622.24  1
3912.97  13251.05  16553.24
##  [36]  24537.05  33784.97  37883.05  47681.18  3
0947.42  23114.50  31353.50
##  [43]  46739.34  27346.29  16168.66  21162.74  2
0611.03  26037.89  25374.47
##  [50]  14838.45  18197.21  17527.34  16878.71  1
7293.29  57857.42  73625.26
##  [57]  50286.84  38511.76  30680.24  30957.11  5
9870.26  46217.37  35024.89
##  [64]  32568.03  17685.97  14845.74  19190.32  1
7289.76  15177.47  15999.03
##  [71]  25536.13  42662.39  66147.87  74674.00 10
3414.34  55393.16  42719.37
##  [78]  33977.00  52213.21  22808.00  16760.18  1
9090.53  19058.42  69751.47

datatesting = dataharga[85:96,]
datatesting

##   [1] 39728.55 38067.66 49499.21 29383.42 36296.
11 87563.45 79093.03 45164.58
##   [9] 53989.34 39658.50 27789.08 37892.29
```

```
#memeriksa kestasioneran data terhadap ragam
BoxCox.ts(datatraining)
```




```
#mentransformasi data
pakai=datatraining^(-0.366)
pakai

##  [1] 0.01834601 0.02600650 0.02383906 0.0273876
6 0.02767999 0.02861911
##  [7] 0.02209432 0.01813229 0.02020435 0.0298050
2 0.03093215 0.02447266
## [13] 0.02348744 0.02880735 0.02080179 0.0259927
0 0.02751557 0.02844243
## [19] 0.02257789 0.02110506 0.02402970 0.0238276
9 0.01986899 0.01911276
## [25] 0.01538636 0.01377021 0.01479806 0.0187105
8 0.01880131 0.02209920
## [31] 0.02169445 0.02548238 0.03044446 0.0309924
8 0.02856859 0.02473575
## [37] 0.02200327 0.02110032 0.01939657 0.0227212
1 0.02528240 0.02261306
## [43] 0.01953872 0.02377361 0.02881544 0.0261120
```

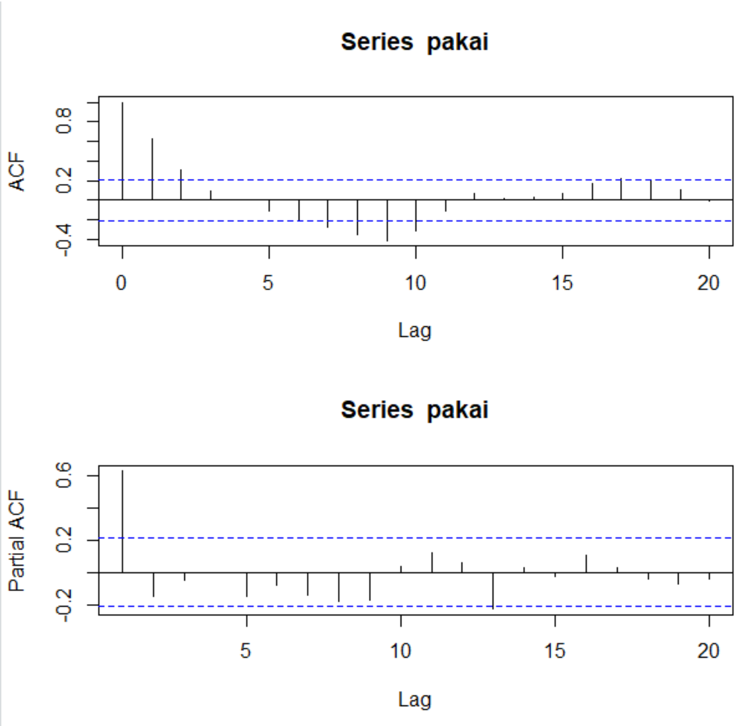
```
3 0.02636571 0.02420406
## [49] 0.02443378 0.02973527 0.02759550 0.0279769
2 0.02836572 0.02811492
## [55] 0.01807077 0.01654505 0.01902250 0.0209735
9 0.02279343 0.02271861
## [61] 0.01784599 0.01961920 0.02171491 0.0223006
9 0.02788482 0.02972992
## [67] 0.02706400 0.02811701 0.02949042 0.0289268
9 0.02437705 0.02020242
## [73] 0.01720645 0.01645963 0.01461044 0.0183609
5 0.02019256 0.02195767
## [79] 0.01876257 0.02540622 0.02843898 0.0271156
9 0.02713240 0.01687561

#memeriksa ulang kestasioneran data terhadap ragam
BoxCox(pakai)
```




```
#memeriksa kestasioneran data terhadap rata-rata
adf.test(pakai)

##
##  Augmented Dickey-Fuller Test
##
```

```
## data:  pakai
## Dickey-Fuller = -4.0279, Lag order = 4, p-value
= 0.01229
## alternative hypothesis: stationary
```

```
#menampilkan plot acf dan pacf
par(mfrow=c(2,1))
acf(pakai, lag=20)
pacf(pakai, lag=20)
```




```
#membentuk model tentatif
model102=arima(ts(pakai), order=c(1,0,2), method="
ML")
model101=arima(ts(pakai), order=c(1,0,1), method="
ML")
model100=arima(ts(pakai), order=c(1,0,0), method="
ML")
```

```
model002=arima(ts(pakai), order=c(0,0,2), method="
ML")
model001=arima(ts(pakai), order=c(0,0,1), method="
ML")

#menguji signifikansi parameter
lmtest::coeftest(model102)

##
## z test of coefficients:
##
##             Estimate Std. Error z value Pr(>|z|
)
## ar1       0.28647597 0.34643242  0.8269   0.408
3
## ma1       0.53786562 0.35971102  1.4953   0.134
8
## ma2       0.19307798 0.23926475  0.8070   0.419
7
## intercept 0.02318492 0.00086691 26.7442   <2e-1
6 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.0
5 '.' 0.1 ' ' 1

lmtest::coeftest(model101)

##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z
|)
## ar1       0.50253729 0.13048280  3.8514 0.00011
75 ***
## ma1       0.29986631 0.14017292  2.1393 0.03241
46 *
## intercept 0.02316414 0.00093366 24.8101 < 2.2e-
16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.0
5 '.' 0.1 ' ' 1

lmtest::coeftest(model100)

##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|
)
## ar1       0.6533063  0.0836392   7.811 5.674e-1
5 ***
## intercept 0.0231632  0.0010457  22.151 < 2.2e-1
6 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.0
5 '.' 0.1 ' ' 1

lmtest::coeftest(model002)

##
## z test of coefficients:
##
##              Estimate Std. Error z value  Pr(>|z
|)
## ma1       0.81235664 0.11297852  7.1904 6.462e-
13 ***
## ma2       0.35708837 0.10079797  3.5426 0.00039
62 ***
## intercept 0.02321957 0.00077929 29.7957 < 2.2e-
16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.0
5 '.' 0.1 ' ' 1

lmtest::coeftest(model001)

##
## z test of coefficients:
##
##              Estimate Std. Error z value  Pr(>|z
```

```
|)
## ma1       0.60227866 0.06939232  8.6793 < 2.2e-
16 ***
## intercept 0.02330505 0.00062025 37.5738 < 2.2e-
16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.0
5 '.' 0.1 ' ' 1

#diagnostik sisaan
sisaan101=model101$residuals
sisaan100=model100$residuals
sisaan002=model002$residuals
sisaan001=model001$residuals

#uji normalitas sisaan
ks.test(sisaan101,"pnorm",mean=mean(sisaan101),sd=
sd(sisaan101))

##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  sisaan101
## D = 0.053425, p-value = 0.9598
## alternative hypothesis: two-sided

ks.test(sisaan100,"pnorm",mean=mean(sisaan100),sd=
sd(sisaan100))

##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  sisaan100
## D = 0.063325, p-value = 0.8678
## alternative hypothesis: two-sided

ks.test(sisaan002,"pnorm",mean=mean(sisaan002),sd=
sd(sisaan002))

##
##  Exact one-sample Kolmogorov-Smirnov test
```

```
##
## data:  sisaan002
## D = 0.061586, p-value = 0.888
## alternative hypothesis: two-sided

ks.test(sisaan001,"pnorm",mean=mean(sisaan001),sd=
sd(sisaan001))

##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  sisaan001
## D = 0.064879, p-value = 0.8485
## alternative hypothesis: two-sided

#uji autokorelasi sisaan
Box.test(sisaan101,type="Box-Pierce")

##
##  Box-Pierce test
##
## data:  sisaan101
## X-squared = 0.00048636, df = 1, p-value = 0.982
4

Box.test(sisaan100,type="Box-Pierce")

##
##  Box-Pierce test
##
## data:  sisaan100
## X-squared = 1.4476, df = 1, p-value = 0.2289

Box.test(sisaan002,type="Box-Pierce")

##
##  Box-Pierce test
##
## data:  sisaan002
## X-squared = 0.00062657, df = 1, p-value = 0.98
```

```
Box.test(sisaan001,type="Box-Pierce")

##
##  Box-Pierce test
##
## data:  sisaan001
## X-squared = 3.1952, df = 1, p-value = 0.07386

#memilih model terbaik menggunakan nilai MSE
r1=forecast::forecast(ts(pakai),model=model101, h=
12)
r1

##    Point Forecast      Lo 80      Hi 80      Lo
95      Hi 95
## 85      0.01735715 0.01314291 0.02157140 0.01091
202 0.02380228
## 86      0.02024591 0.01484272 0.02564911 0.01198
243 0.02850940
## 87      0.02169762 0.01603350 0.02736175 0.01303
509 0.03036016
## 88      0.02242716 0.01669902 0.02815531 0.01366
673 0.03118760
## 89      0.02279378 0.01704959 0.02853798 0.01400
879 0.03157877
## 90      0.02297802 0.01722978 0.02872627 0.01418
684 0.03176920
## 91      0.02307061 0.01732135 0.02881988 0.01427
787 0.03186335
## 92      0.02311714 0.01736762 0.02886666 0.01432
400 0.03191028
## 93      0.02314052 0.01739093 0.02889011 0.01434
729 0.03193376
## 94      0.02315227 0.01740267 0.02890188 0.01435
901 0.03194554
## 95      0.02315818 0.01740857 0.02890779 0.01436
491 0.03195145
## 96      0.02316115 0.01741154 0.02891076 0.01436
788 0.03195442
```

```
r2=forecast::forecast(ts(pakai),model=model100, h=
12)
r2

##    Point Forecast      Lo 80      Hi 80      Lo
95      Hi 95
## 85      0.01905548 0.01474239 0.02336858 0.01245
917 0.02565179
## 86      0.02047961 0.01532765 0.02563156 0.01260
037 0.02835884
## 87      0.02141000 0.01593904 0.02688096 0.01304
288 0.02977711
## 88      0.02201783 0.01641624 0.02761942 0.01345
094 0.03058472
## 89      0.02241493 0.01675850 0.02807135 0.01376
417 0.03106568
## 90      0.02267435 0.01699469 0.02835402 0.01398
806 0.03136065
## 91      0.02284384 0.01715428 0.02853340 0.01414
241 0.03154527
## 92      0.02295457 0.01726079 0.02864834 0.01424
669 0.03166244
## 93      0.02302690 0.01733133 0.02872247 0.01431
628 0.03173753
## 94      0.02307416 0.01737782 0.02877050 0.01436
236 0.03178596
## 95      0.02310504 0.01740837 0.02880170 0.01439
274 0.03181734
## 96      0.02312521 0.01742840 0.02882201 0.01441
269 0.03183772

r3=forecast::forecast(ts(pakai),model=model002, h=
12)
r3

##    Point Forecast      Lo 80      Hi 80      Lo
95      Hi 95
## 85      0.01711622 0.01290867 0.02132377 0.01068
133 0.02355111
## 86      0.02006964 0.01464872 0.02549056 0.01177
906 0.02836022
```

```
## 87      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 88      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 89      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 90      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 91      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 92      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 93      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 94      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 95      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269
## 96      0.02321957 0.01759429 0.02884484 0.01461
644 0.03182269

r4=forecast::forecast(ts(pakai),model=model001, h=
12)
r4

##     Point Forecast      Lo 80      Hi 80      Lo
95      Hi 95
## 85      0.01858741 0.01409302 0.02308181 0.01171
383 0.02546100
## 86      0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 87      0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 88      0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 89      0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 90      0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 91      0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
```

```
## 92     0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 93     0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 94     0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 95     0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903
## 96     0.02330505 0.01805845 0.02855164 0.01528
107 0.03132903

pred1=(r1$mean)^(1/(-0.366))
pred1

## Time Series:
## Start = 85
## End = 96
## Frequency = 1
##  [1] 64590.40 42412.46 35101.19 32068.67 30678.
94 30011.50 29683.57 29520.61
##  [9] 29439.18 29398.38 29377.90 29367.62

pred2=(r2$mean)^(1/(-0.366))
pred2

## Time Series:
## Start = 85
## End = 96
## Frequency = 1
##  [1] 50049.42 41103.17 36404.64 33723.96 32116.
53 31122.47 30495.62 30095.38
##  [9] 29837.77 29671.09 29562.89 29492.49

pred3=(r3$mean)^(1/(-0.366))
pred3

## Time Series:
## Start = 85
## End = 96
## Frequency = 1
##  [1] 67104.95 43438.03 29166.18 29166.18 29166.
```

```
18 29166.18 29166.18 29166.18
##  [9] 29166.18 29166.18 29166.18 29166.18

pred4=(r4$mean)^(1/(-0.366))
pred4

## Time Series:
## Start = 85
## End = 96
## Frequency = 1
##  [1] 53568.55 28874.81 28874.81 28874.81 28874.
81 28874.81 28874.81 28874.81
##  [9] 28874.81 28874.81 28874.81 28874.81

a1=accuracy(pred1,datatesting)
a1

##                 ME     RMSE      MAE      MPE
MAPE
## Test set 12706.23 25272.65 18286.35 17.69907 32
.50719

a2=accuracy(pred2,datatesting)
a2

##                 ME    RMSE      MAE     MPE
MAPE
## Test set 13370.82 23900.2 16615.94 19.5177 28.7
0231

a3=accuracy(pred3,datatesting)
a3

##                 ME     RMSE      MAE      MPE
MAPE
## Test set 13493.37 26149.92 19180.68 19.51586 34
.17779

a4=accuracy(pred4,datatesting)
a4
```

```
##                     ME      RMSE       MAE       MPE
MAPE
## Test set 16077.81 25524.26 18565.43 26.11164 32
.56888

rmse=c(a1[2],a2[2],a3[2],a4[2])
rmse

## [1] 25272.65 23900.20 26149.92 25524.26

mse=rmse^2
mse

## [1] 638707024 571219562 683818523 651487965

#melakukan peramalan menggunakan model terbaik
modelfix=arima(ts(dataharga), order=c(1,0,0), meth
od="ML")
ramalan=forecast::forecast(ts(dataharga),model=mod
elfix, h=12)
ramalan

##     Point Forecast     Lo 80    Hi 80      Lo 9
5    Hi 95
##  97       37581.80 16679.725 58483.88  5614.834
5 69548.77
##  98       37377.74 12365.619 62389.87  -874.999
8 75630.48
##  99       37243.63 10651.980 63835.28 -3424.790
7 77912.05
## 100       37155.49  9909.895 64401.09 -4513.052
5 78824.04
## 101       37097.57  9574.314 64620.82 -4995.615
7 79190.75
## 102       37059.50  9417.178 64701.81 -5215.780
9 79334.77
## 103       37034.48  9340.888 64728.06 -5319.211
7 79388.16
## 104       37018.03  9302.329 64733.74 -5369.478
5 79405.54
## 105       37007.23  9281.975 64732.48 -5394. 5
```

```
79409.34
## 106         37000.12  9270.749 64729.50 -5408.294
1 79408.54
## 107         36995.45  9264.301 64726.61 -5415.685
1 79406.59
## 108         36992.39  9260.464 64724.31 -5419.929
0 79404.70

#membentuk plot ramalan vs aktual
akt=c(50939,56860,68969,31636,30605,34085)
aktu=c(dataharga,akt)
aktual=ts(aktu, frequency=12, start=c(2015,1))
hasil=ts(ramalan$mean, frequency=12, start=c(2023,
1))

win.graph()
plot(aktual, xlab="Waktu", ylab="Harga (Rupiah)",
col='red')
points(hasil, pch=20, col="blue")
```