TAKAGI-SUGENO FUZZY MODELING

TUGAS AKHIR

MODEL LOGIKA FUZZY

Oleh :

1. **Rauzan Sumara (135090501111014)**
2. **M. Yuyud Anizar (135090507111006)**

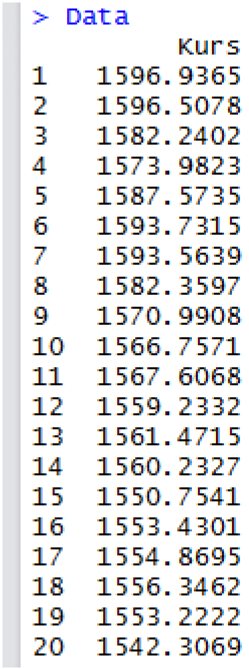
Fuzzy Logic

PROGRAM STUDI
STATISTIKA
FAKULTAS
MATEMATIKA DAN
ILMU PENGETAHUAN
ALAM UNIVERSITAS
BRAWIJAYA

Berikut ini akan dijelaskan tahap-tahap dalam memprediksi data deret waktu menggunakan metode *Fuzzy* Takagi-Sugeno. Data deret waktu yang digunakan yaitu nilai kurs Rupiah terhadap Swedia (IDR/SEK) mulai tangal 01 Januari 2015 sampai 10 April 2015 ($n = 100$).

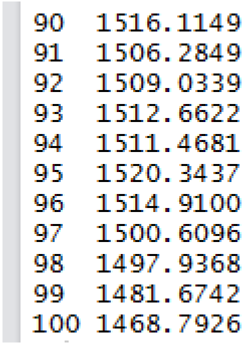
Tanggal	*Kurs*	*Tanggal*	*Kurs*	*Tanggal*	*Kurs*
01/01/2015	1596.937	*04/02/2015*	1523.025	*10/03/2015*	1536.256
02/01/2015	1596.508	*05/02/2015*	1531.7279	*11/03/2015*	1529.235
03/01/2015	1582.24	*06/02/2015*	1509.3388	*12/03/2015*	1535.194
04/01/2015	1573.982	*07/02/2015*	1510.1184	*13/03/2015*	1518.793
05/01/2015	1587.574	*08/02/2015*	1502.9263	*14/03/2015*	1519.115
06/01/2015	1593.732	*09/02/2015*	1515.7618	*15/03/2015*	1517.368
07/01/2015	1593.564	*10/02/2015*	1523.2385	*16/03/2015*	1526.411
08/01/2015	1582.36	*11/02/2015*	1523.7132	*17/03/2015*	1517.956
09/01/2015	1570.991	*12/02/2015*	1515.7651	*18/03/2015*	1521.811
10/01/2015	1566.757	*13/02/2015*	1516.3187	*19/03/2015*	1507.649
11/01/2015	1567.607	*14/02/2015*	1515.0659	*20/03/2015*	1511.857
12/01/2015	1559.233	*15/02/2015*	1517.1538	*21/03/2015*	1513.807
13/01/2015	1561.472	*16/02/2015*	1514.992	*22/03/2015*	1517.124
14/01/2015	1560.233	*17/02/2015*	1532.9727	*23/03/2015*	1525.841
15/01/2015	1550.754	*18/02/2015*	1535.5013	*24/03/2015*	1520.22
16/01/2015	1553.43	*19/02/2015*	1527.7996	*25/03/2015*	1526.922
17/01/2015	1554.87	*20/02/2015*	1532.3782	*26/03/2015*	1514.969
18/01/2015	1556.346	*21/02/2015*	1532.1574	*27/03/2015*	1522.907
19/01/2015	1553.222	*22/02/2015*	1531.3465	*28/03/2015*	1522.192
20/01/2015	1542.307	*23/02/2015*	1528.8916	*29/03/2015*	1521.246
21/01/2015	1533.218	*24/02/2015*	1531.6619	*30/03/2015*	1520.86
22/01/2015	1519.267	*25/02/2015*	1549.1531	*31/03/2015*	1516.115
23/01/2015	1501.65	*26/02/2015*	1531.059	*01/04/2015*	1506.285
24/01/2015	1497.999	*27/02/2015*	1550.6478	*02/04/2015*	1509.034
25/01/2015	1491.3	*28/02/2015*	1552.4525	*03/04/2015*	1512.662
26/01/2015	1501.042	*01/03/2015*	1553.1833	*04/04/2015*	1511.468
27/01/2015	1519.962	*02/03/2015*	1557.4921	*05/04/2015*	1520.344
28/01/2015	1506.984	*03/03/2015*	1566.0803	*06/04/2015*	1514.91
29/01/2015	1522.084	*04/03/2015*	1560.6493	*07/04/2015*	1500.61
30/01/2015	1529.524	*05/03/2015*	1561.4128	*08/04/2015*	1497.937
31/01/2015	1533.852	*06/03/2015*	1542.6846	*09/04/2015*	1481.674
01/02/2015	1540.433	*07/03/2015*	1540.8291	*10/04/2015*	1468.793
02/02/2015	1529.22	*08/03/2015*	1540.5509		
03/02/2015	1535.441	*09/03/2015*	1542.5527		

Berikut merupakan hasil *input* data ke dalam *software* R :



```
> Data
         Kurs
1   1596.9365
2   1596.5078
3   1582.2402
4   1573.9823
5   1587.5735
6   1593.7315
7   1593.5639
8   1582.3597
9   1570.9908
10  1566.7571
11  1567.6068
12  1559.2332
13  1561.4715
14  1560.2327
15  1550.7541
16  1553.4301
17  1554.8695
18  1556.3462
19  1553.2222
20  1542.3069
```

....



```
90  1516.1149
91  1506.2849
92  1509.0339
93  1512.6622
94  1511.4681
95  1520.3437
96  1514.9100
97  1500.6096
98  1497.9368
99  1481.6742
100 1468.7926
```

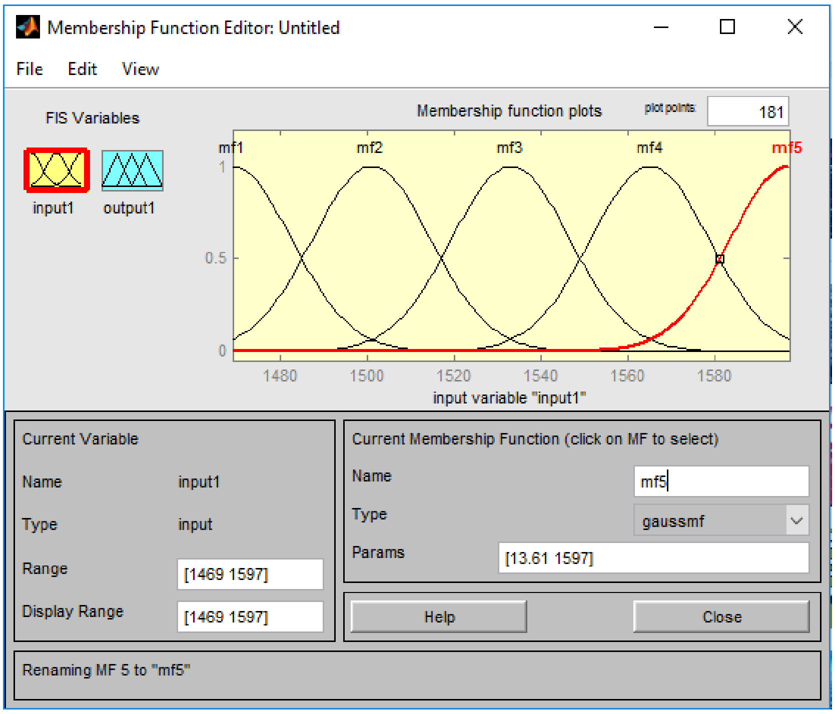
✓ Pertama, mencari nilai maksimum dan minimum dari data yang bertujuan untuk memudahkan dalam menentukan parameter dari setiap himpunan *fuzzy* :

Minimum	**Maksimum**
1468.7926	1596.9365

✓ Kedua yaitu menentukan nilai parameter dari setiap himpunan *fuzzy*. Pada kasus ini digunakan fungsi keanggotaan gaussmf (Gaussian) dengan 5 himpunan *fuzzy* (*fuzzy set*), yakni :

$$\mu_{A_k^l}(y_{t,k}) = \exp\left(-\frac{1}{2}\left(\frac{y_{t,k} - c_k^l}{\sigma_k^l}\right)^2\right)$$

dimana A_k^l adalah himpunan *fuzzy* dari variabel input ke-k untuk rule ke-l yang direpresentasikan oleh suatu fungsi keanggotaan $\mu_{A_k^l}(y_{t,k})$ sedangkan c_k^l dan σ_k^l berturut-turut adalah parameter pusat (*mean*) dan parameter *spread* gaussian (zigma). Berikut adalah nilai hasil parameter dari setiap himpunan *fuzzy* melalui *software* MATLAB :



Current Variable

Name input1

Type input

Range [1469 1597]

Display Range [1469 1597]

Current Membership Function (click on MF to select)

Name mf5

Type gaussmf

Params [13.61 1597]

Renaming MF 5 to "mf5"

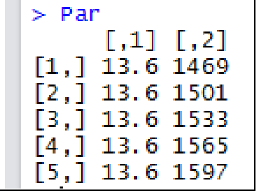
Berikut *script* pada *software* R untuk mencari parameter Gaussian :

```
-----------------------------------------------------------------
#Fungsi Mencari Parameter Gaussian
#Keterangan :
#Data -> Data time series berupa suatu vektor
#k -> Banyaknya himpunan fuzzy berupa suatu skalar
-----------------------------------------------------------------
Par.gauss <- function(Data,k){
Maxx <- ceiling(max(Data))
Minn <- ceiling(min(Data))
r <- (Maxx-Minn)
s <- r/(k-1)
mu <- matrix(0,k,2)
zikma <- (s/2)/1.1774
for (i in 1:k){
     mu[i,1] <- zikma
     mu[i,2] <- Minn + (i-1)*s
  }
Par.gauss <- mu
}
```

Nilai parameter yang diperoleh di *export* ke dalam *software* R sebagai berikut :

Fuzzy Set	*σ*	*μ*
Fuzzy Set-1	13.61	1469
Fuzzy Set-2	13.61	1501
Fuzzy Set-3	13.61	1533
Fuzzy Set-4	13.61	1565
Fuzzy Set-5	13.61	1597

Software R

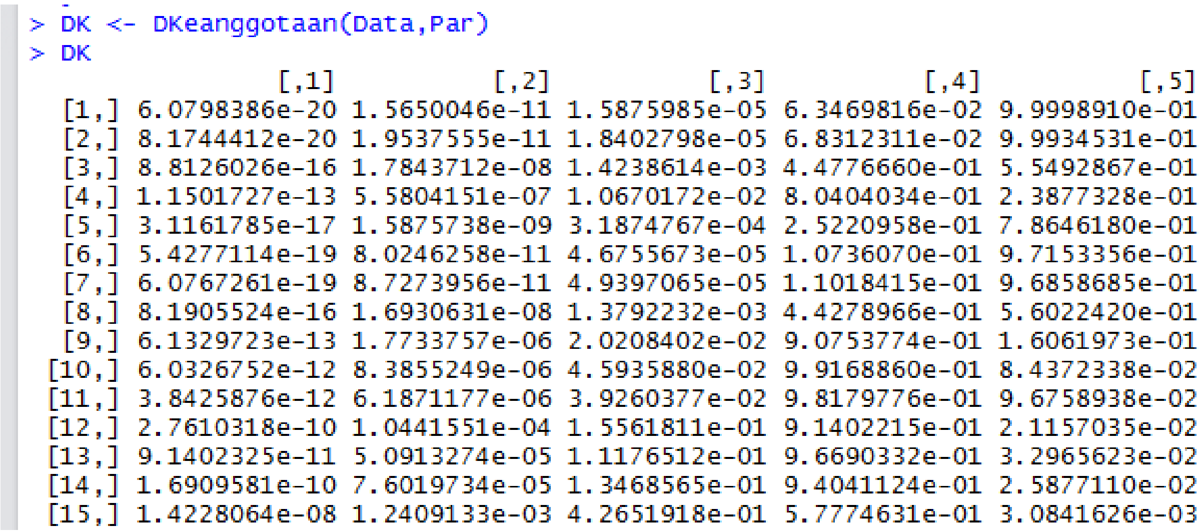


```
> Par
     [,1] [,2]
[1,] 13.6 1469
[2,] 13.6 1501
[3,] 13.6 1533
[4,] 13.6 1565
[5,] 13.6 1597
```

✓ Setelah itu, dari nilai parameter yang diperoleh digunakan untuk mencari derajat keanggotaan dari data pada setiap himpunan *fuzzy*. Berikut *script* pada *software* R :

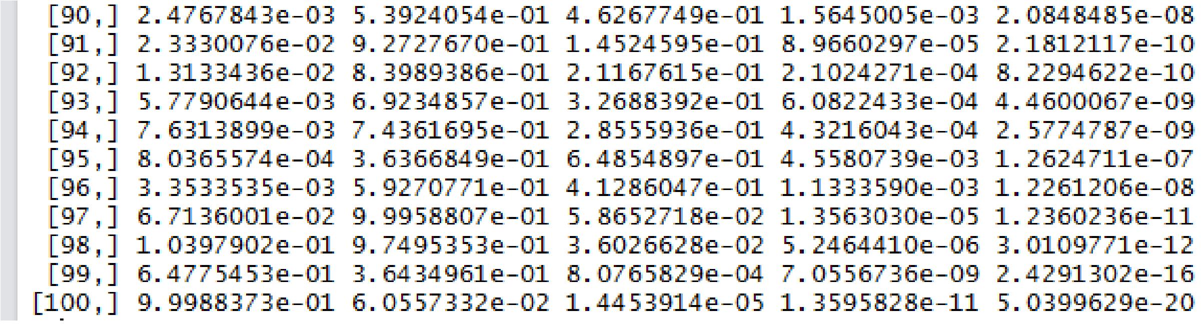
```
-----------------------------------------------------------------
#Fungsi Mencari Derajat Keanggotaan
#Keterangan :
#Ser -> Data time series berupa suatu vektor
#Par -> Parameter dari fungsi keanggotaan gaussian sigma, mean,
and tinggi berupa suatu matriks
-----------------------------------------------------------------
DKeanggotaan <- function(Ser,Par){
m <- nrow(Par)
myMF <- NULL
for(i in 1:m) {
     myDM <- gaussMF("myDM",Ser,c(Par[i,],1))$mfVals
     myMF <- cbind(myMF,myDM)
     }
colnames(myMF) <- NULL
DKeanggotaan <- myMF
}
```

Sehingga diperoleh derajat keanggotaan dari data pada setiap himpunan *fuzzy* sebagai berikut:



```
> DK <- DKeanggotaan(Data,Par)
> DK
               [,1]         [,2]         [,3]         [,4]         [,5]
  [1,] 6.0798386e-20 1.5650046e-11 1.5875985e-05 6.3469816e-02 9.9998910e-01
  [2,] 8.1744412e-20 1.9537555e-11 1.8402798e-05 6.8312311e-02 9.9934531e-01
  [3,] 8.8126026e-16 1.7843712e-08 1.4238614e-03 4.4776660e-01 5.5492867e-01
  [4,] 1.1501727e-13 5.5804151e-07 1.0670172e-02 8.0404034e-01 2.3877328e-01
  [5,] 3.1161785e-17 1.5875738e-09 3.1874767e-04 2.5220958e-01 7.8646180e-01
  [6,] 5.4277114e-19 8.0246258e-11 4.6755673e-05 1.0736070e-01 9.7153356e-01
  [7,] 6.0767261e-19 8.7273956e-11 4.9397065e-05 1.1018415e-01 9.6858685e-01
  [8,] 8.1905524e-16 1.6930631e-08 1.3792232e-03 4.4278966e-01 5.6022420e-01
  [9,] 6.1329723e-13 1.7733757e-06 2.0208402e-02 9.0753774e-01 1.6061973e-01
 [10,] 6.0326752e-12 8.3855249e-06 4.5935880e-02 9.9168860e-01 8.4372338e-02
 [11,] 3.8425876e-12 6.1871177e-06 3.9260377e-02 9.8179776e-01 9.6758938e-02
 [12,] 2.7610318e-10 1.0441551e-04 1.5561811e-01 9.1402215e-01 2.1157035e-02
 [13,] 9.1402325e-11 5.0913274e-05 1.1176512e-01 9.6690332e-01 3.2965623e-02
 [14,] 1.6909581e-10 7.6019734e-05 1.3468565e-01 9.4041124e-01 2.5877110e-02
 [15,] 1.4228064e-08 1.2409133e-03 4.2651918e-01 5.7774631e-01 3.0841626e-03
```

....



```
 [90,] 2.4767843e-03 5.3924054e-01 4.6267749e-01 1.5645005e-03 2.0848485e-08
 [91,] 2.3330076e-02 9.2727670e-01 1.4524595e-01 8.9660297e-05 2.1812117e-10
 [92,] 1.3133436e-02 8.3989386e-01 2.1167615e-01 2.1024271e-04 8.2294622e-10
 [93,] 5.7790644e-03 6.9234857e-01 3.2688392e-01 6.0822433e-04 4.4600067e-09
 [94,] 7.6313899e-03 7.4361695e-01 2.8555936e-01 4.3216043e-04 2.5774787e-09
 [95,] 8.0365574e-04 3.6366849e-01 6.4854897e-01 4.5580739e-03 1.2624711e-07
 [96,] 3.3533535e-03 5.9270771e-01 4.1286047e-01 1.1333590e-03 1.2261206e-08
 [97,] 6.7136001e-02 9.9958807e-01 5.8652718e-02 1.3563030e-05 1.2360236e-11
 [98,] 1.0397902e-01 9.7495353e-01 3.6026628e-02 5.2464410e-06 3.0109771e-12
 [99,] 6.4775453e-01 3.6434961e-01 8.0765829e-04 7.0556736e-09 2.4291302e-16
[100,] 9.9988373e-01 6.0557332e-02 1.4453914e-05 1.3595828e-11 5.0399629e-20
```

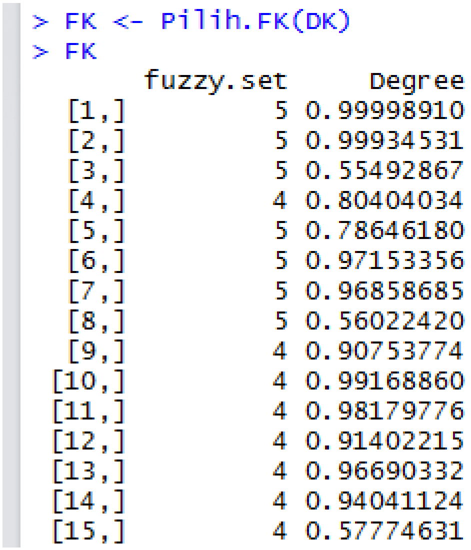
Keterangan :

- Kolom1 = derajat keanggotaan setiap data pada himpunan *fuzzy* pertama,
- Kolom2 = derajat keanggotaan setiap data pada himpunan *fuzzy* kedua,
- Kolom3 = derajat keanggotaan setiap data pada himpunan *fuzzy* ketiga,
- Kolom4 = derajat keanggotaan setiap data pada himpunan *fuzzy* keempat,
- Kolom5 = derajat keanggotaan setiap data pada himpunan *fuzzy* kelima.

✓ Kemudian kita memilih nilai derajat keanggotaan dan himpunan *fuzzy* yang mewakili setiap data. Berikut ini merupakan *script* pada *software* R :

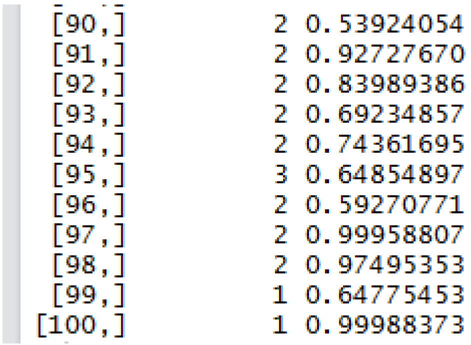
```
-----------------------------------------------------------------
#Mencari Letak Himpunan Fuzzy Untuk Setiap Data
#Keterangan :
#DK -> Derajat keanggotaan data yang berupa suatu matriks
-----------------------------------------------------------------
Pilih.FK <- function(DK) {
n <- nrow(DK)
c <- NULL
s <- NULL
for(i in 1:n) {
     c[i] <- max(DK[i,])
     s[i] <- which.max(DK[i,])
     }
Pilih.FK <- cbind(fuzzy.set=s,Degree=c)
}
```

Sehingga diperoleh hasil sebagai berikut :



```
> FK <- Pilih.FK(DK)
> FK
      fuzzy.set     Degree
 [1,]         5 0.99998910
 [2,]         5 0.99934531
 [3,]         5 0.55492867
 [4,]         4 0.80404034
 [5,]         5 0.78646180
 [6,]         5 0.97153356
 [7,]         5 0.96858685
 [8,]         5 0.56022420
 [9,]         4 0.90753774
[10,]         4 0.99168860
[11,]         4 0.98179776
[12,]         4 0.91402215
[13,]         4 0.96690332
[14,]         4 0.94041124
[15,]         4 0.57774631
```

……………………………………………………

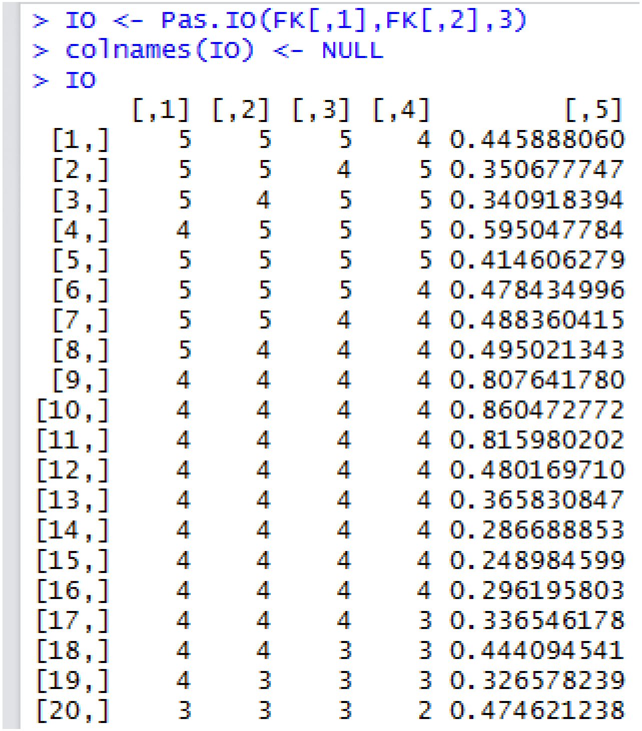


```
 [90,]         2 0.53924054
 [91,]         2 0.92727670
 [92,]         2 0.83989386
 [93,]         2 0.69234857
 [94,]         2 0.74361695
 [95,]         3 0.64854897
 [96,]         2 0.59270771
 [97,]         2 0.99958807
 [98,]         2 0.97495353
 [99,]         1 0.64775453
[100,]         1 0.99988373
```

- ✓ Setelah itu kita menentukan pasangan *Input-Output*. Pada kasus ini digunakan *input* sebanyak 3 lag. Berikut *script* pada *software* R :

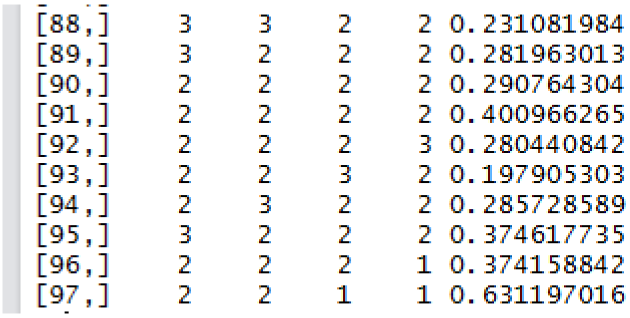
```
-----------------------------------------------------------------
#Mencari Rule dari Pasangan Input Output
#Keterangan :
#FK -> Himpunan fuzzy yang mewakili suatu data yang berupa suatu
vektor
#DK -> Derajat keanggotaan dari data yang berupa suatu vektor
#m -> Banyaknya input yang berpengaruh yang berupa suatu sekalar
-----------------------------------------------------------------
Pas.IO <- function(FK,DK,m) {
n <- length(FK)
Pasangan <- matrix(0,(n-m),(m+1))
Degree <- NULL
for(i in 1:(n-m)) {
     s <- i
     for(j in 1:(m+1)) {
          Pasangan[i,j] = FK[s]
          s <- s+1
          }
     Degree[i] <- prod(DK[i:(m+i)])
   }
Pas.IO <- cbind(Pasangan,Degree)
}
```

Sehingga diperoleh hasil sebagai berikut :



```
> IO <- Pas.IO(FK[,1],FK[,2],3)
> colnames(IO) <- NULL
> IO
      [,1] [,2] [,3] [,4]        [,5]
 [1,]    5    5    5    4 0.445888060
 [2,]    5    5    4    5 0.350677747
 [3,]    5    4    5    5 0.340918394
 [4,]    4    5    5    5 0.595047784
 [5,]    5    5    5    5 0.414606279
 [6,]    5    5    5    4 0.478434996
 [7,]    5    5    4    4 0.488360415
 [8,]    5    4    4    4 0.495021343
 [9,]    4    4    4    4 0.807641780
[10,]    4    4    4    4 0.860472772
[11,]    4    4    4    4 0.815980202
[12,]    4    4    4    4 0.480169710
[13,]    4    4    4    4 0.365830847
[14,]    4    4    4    4 0.286688853
[15,]    4    4    4    4 0.248984599
[16,]    4    4    4    4 0.296195803
[17,]    4    4    4    3 0.336546178
[18,]    4    4    3    3 0.444094541
[19,]    4    3    3    3 0.326578239
[20,]    3    3    3    2 0.474621238
```

....



```
[88,]    3    3    2    2 0.231081984
[89,]    3    2    2    2 0.281963013
[90,]    2    2    2    2 0.290764304
[91,]    2    2    2    2 0.400966265
[92,]    2    2    2    3 0.280440842
[93,]    2    2    3    2 0.197905303
[94,]    2    3    2    2 0.285728589
[95,]    3    2    2    2 0.374617735
[96,]    2    2    2    1 0.374158842
[97,]    2    2    1    1 0.631197016
```

Y_{t-3}	Y_{t-2}	Y_{t-1}	Y_t
-	-	-	5
-	-	5	5
-	5	5	5
5	5	5	4
5	5	4	5
5	4	5	5
4	5	5	5
5	5	5	5
5	5	5	4
5	5	4	4
5	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	3
4	4	3	3
....			
2	2	2	3
2	2	3	2
2	3	2	2
3	2	2	2
2	2	2	1
2	2	1	1
2	1	1	
1	1		
1			

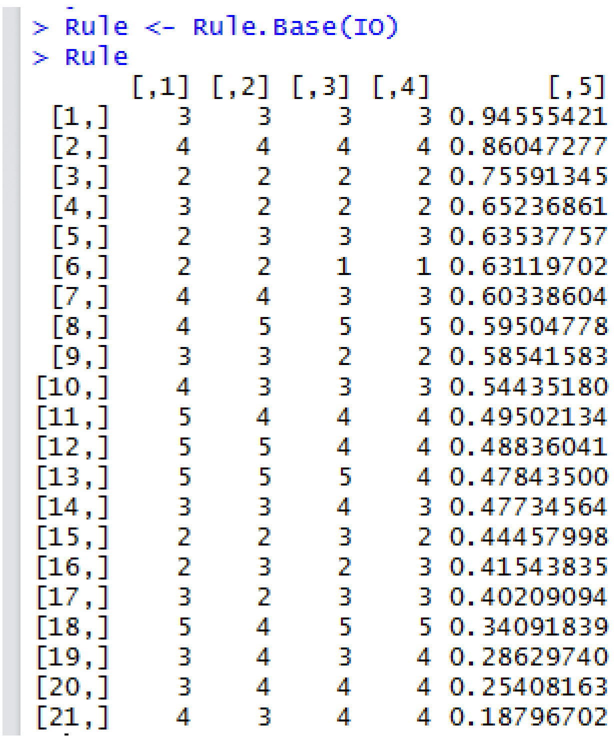
Keterangan :

Kolom1 -> Y_{t-3}
Kolom2 -> Y_{t-2}
Kolom3 -> Y_{t-1}
Kolom4 -> Y_t
Kolom5 -> *Degree*

✓ Kemudian mencari basis kaidah (*Rule Base*) yang bersesuaian dan tidak konflik. Berikut ini merupakan *script* pada *software* R :

```
-----------------------------------------------------------------
#Mencari Rule Base
#Keterangan :
#Pasangan  ->  Pasangan  input  ouput  dengan  tambahan  derajat
keanggotaan yang berupa suatu matriks
-----------------------------------------------------------------
Rule.Base <- function(Pasangan) {
n <- nrow(Pasangan)
m <- ncol(Pasangan)
KR <- Pasangan[order(Pasangan[,m],decreasing=T),]
KRt <- KR[,1:(m-2)]
for(i in 1:(n-1)) {
     for(j in (i+1):n) {
          in.cond <- all(KRt[i,]==KRt[j,])
          if(in.cond==TRUE) {
               KR[j,] <- 0
          }
     }
}
rule <- KR[which(KR[,1]!=0),]
Rule.Base=rule
}
```

Sehingga diperoleh basis kaidah sebanyak 21. Hasil dari *software* R sebagai berikut :

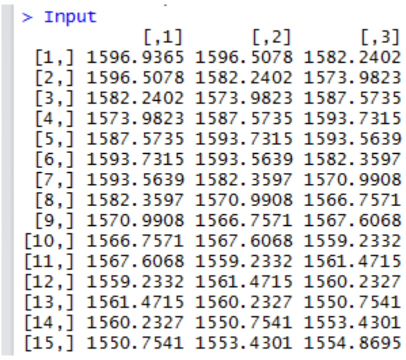


```
> Rule <- Rule.Base(IO)
> Rule
      [,1] [,2] [,3] [,4]       [,5]
 [1,]    3    3    3    3 0.94555421
 [2,]    4    4    4    4 0.86047277
 [3,]    2    2    2    2 0.75591345
 [4,]    3    2    2    2 0.65236861
 [5,]    2    3    3    3 0.63537757
 [6,]    2    2    1    1 0.63119702
 [7,]    4    4    3    3 0.60338604
 [8,]    4    5    5    5 0.59504778
 [9,]    3    3    2    2 0.58541583
[10,]    4    3    3    3 0.54435180
[11,]    5    4    4    4 0.49502134
[12,]    5    5    4    4 0.48836041
[13,]    5    5    5    4 0.47843500
[14,]    3    3    4    3 0.47734564
[15,]    2    2    3    2 0.44457998
[16,]    2    3    2    3 0.41543835
[17,]    3    2    3    3 0.40209094
[18,]    5    4    5    5 0.34091839
[19,]    3    4    3    4 0.28629740
[20,]    3    4    4    4 0.25408163
[21,]    4    3    4    4 0.18796702
```

- ✓ Kemudian memasukan nilai variabel *input* ke dalam masing-masing *rule base* untuk memperoleh derajat keanggotaan masing-masing nilai pada variabel *input*. Berikut variabel *Input* dan *Output* :

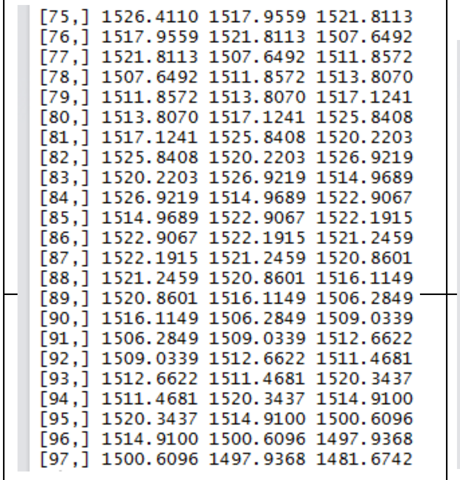
Variabel Input			*Output*
Y_{t-3}	Y_{t-2}	Y_{t-1}	Y_t
-	-	-	1596.9365
-	-	1596.9365	1596.5078
-	1596.9365	1596.5078	1582.2402
1596.9365	1596.5078	1582.2402	1573.9823
1596.5078	1582.2402	1573.9823	1587.5735
1582.2402	1573.9823	1587.5735	1593.7315
1573.9823	1587.5735	1593.7315	1593.5639
1587.5735	1593.7315	1593.5639	1582.3597
1593.7315	1593.5639	1582.3597	1570.9908
1593.5639	1582.3597	1570.9908	1566.7571
1582.3597	1570.9908	1566.7571	1567.6068
1570.9908	1566.7571	1567.6068	1559.2332
1566.7571	1567.6068	1559.2332	1561.4715
1567.6068	1559.2332	1561.4715	1560.2327
1559.2332	1561.4715	1560.2327	1550.7541
1561.4715	1560.2327	1550.7541	1553.4301
1560.2327	1550.7541	1553.4301	1554.8695
....	....	....	....
1506.2849	1509.0339	1512.6622	1511.4681
1509.0339	1512.6622	1511.4681	1520.3437
1512.6622	1511.4681	1520.3437	1514.91
1511.4681	1520.3437	1514.91	1500.6096
1520.3437	1514.91	1500.6096	1497.9368
1514.91	1500.6096	1497.9368	1481.6742
1500.6096	1497.9368	1481.6742	1468.7926
1497.9368	1481.6742	1468.7926	
1481.6742	1468.7926		
1468.7926			

Variabel Input	*Output*

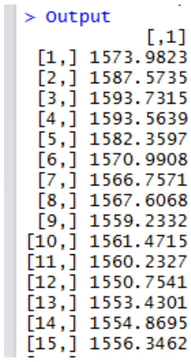


```
> Input
            [,1]      [,2]      [,3]
 [1,] 1596.9365 1596.5078 1582.2402
 [2,] 1596.5078 1582.2402 1573.9823
 [3,] 1582.2402 1573.9823 1587.5735
 [4,] 1573.9823 1587.5735 1593.7315
 [5,] 1587.5735 1593.7315 1593.5639
 [6,] 1593.7315 1593.5639 1582.3597
 [7,] 1593.5639 1582.3597 1570.9908
 [8,] 1582.3597 1570.9908 1566.7571
 [9,] 1570.9908 1566.7571 1567.6068
[10,] 1566.7571 1567.6068 1559.2332
[11,] 1567.6068 1559.2332 1561.4715
[12,] 1559.2332 1561.4715 1560.2327
[13,] 1561.4715 1560.2327 1550.7541
[14,] 1560.2327 1550.7541 1553.4301
[15,] 1550.7541 1553.4301 1554.8695
```

....

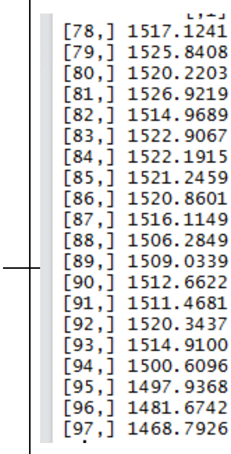


```
[75,] 1526.4110 1517.9559 1521.8113
[76,] 1517.9559 1521.8113 1507.6492
[77,] 1521.8113 1507.6492 1511.8572
[78,] 1507.6492 1511.8572 1513.8070
[79,] 1511.8572 1513.8070 1517.1241
[80,] 1513.8070 1517.1241 1525.8408
[81,] 1517.1241 1525.8408 1520.2203
[82,] 1525.8408 1520.2203 1526.9219
[83,] 1520.2203 1526.9219 1514.9689
[84,] 1526.9219 1514.9689 1522.9067
[85,] 1514.9689 1522.9067 1522.1915
[86,] 1522.9067 1522.1915 1521.2459
[87,] 1522.1915 1521.2459 1520.8601
[88,] 1521.2459 1520.8601 1516.1149
[89,] 1520.8601 1516.1149 1506.2849
[90,] 1516.1149 1506.2849 1509.0339
[91,] 1506.2849 1509.0339 1512.6622
[92,] 1509.0339 1512.6622 1511.4681
[93,] 1512.6622 1511.4681 1520.3437
[94,] 1511.4681 1520.3437 1514.9100
[95,] 1520.3437 1514.9100 1500.6096
[96,] 1514.9100 1500.6096 1497.9368
[97,] 1500.6096 1497.9368 1481.6742
```



```
> Output
           [,1]
 [1,] 1573.9823
 [2,] 1587.5735
 [3,] 1593.7315
 [4,] 1593.5639
 [5,] 1582.3597
 [6,] 1570.9908
 [7,] 1566.7571
 [8,] 1567.6068
 [9,] 1559.2332
[10,] 1561.4715
[11,] 1560.2327
[12,] 1550.7541
[13,] 1553.4301
[14,] 1554.8695
[15,] 1556.3462
```

....

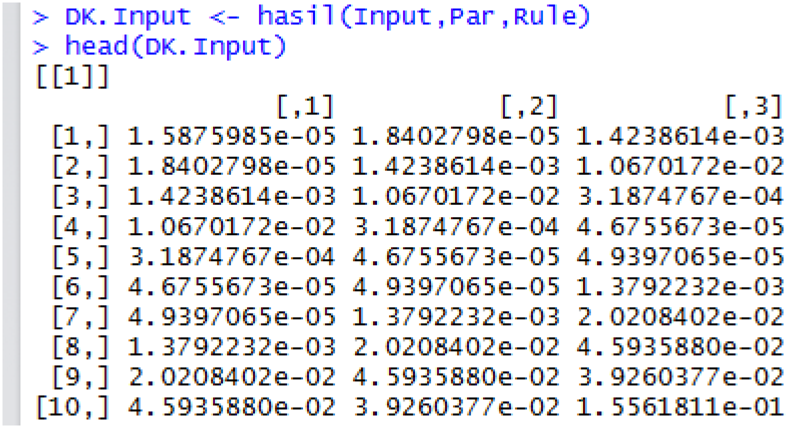


```
[78,] 1517.1241
[79,] 1525.8408
[80,] 1520.2203
[81,] 1526.9219
[82,] 1514.9689
[83,] 1522.9067
[84,] 1522.1915
[85,] 1521.2459
[86,] 1520.8601
[87,] 1516.1149
[88,] 1506.2849
[89,] 1509.0339
[90,] 1512.6622
[91,] 1511.4681
[92,] 1520.3437
[93,] 1514.9100
[94,] 1500.6096
[95,] 1497.9368
[96,] 1481.6742
[97,] 1468.7926
```

Berikut ini adalah *script* pada *software* R untuk memperoleh derajat keanggotaan masing-masing nilai pada variabel *input* yang bersesuaian dengan *rule base* :

```
----------------------------------------------------------------
#memasukan data pada rule base
#keterangan :
#Data -> Data input yang berupa suatu matriks
#Par -> Parameter fungsi keanggotaan yang berupa suatu matriks
#Rule -> Rule base atau basis kaidah yang berupa suatu matriks
----------------------------------------------------------------
hasil <- function(Data,Par,Rule) {
k <- ncol(Data)
l <- nrow(Rule)
n <- nrow(Par)
envi <- list()
for(i in 1:l) {
     myMF <- NULL
     for(j in 1:k) {
          s <- Rule[i,j]
          myDM <- gaussMF("",Data[,j],c(Par[s,],1))$mfVals
          myMF <- cbind(myMF,myDM)
          }
colnames(myMF) <- NULL
envi[[i]] <- myMF
     }
hasil <- envi
}
```

Sehingga diperoleh suatu tipe *data list* dengan *length list* sebanyak *rule base* yaitu 21. hasil pada *software* R sebagai berikut :



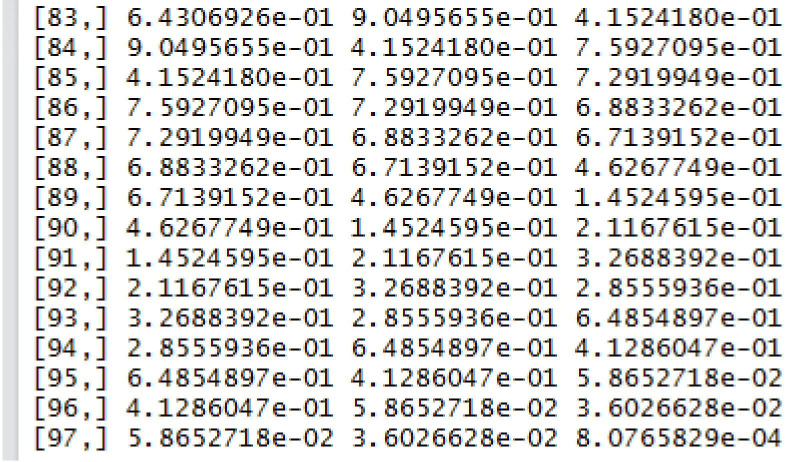
```
> DK.Input <- hasil(Input,Par,Rule)
> head(DK.Input)
[[1]]
              [,1]         [,2]         [,3]
 [1,] 1.5875985e-05 1.8402798e-05 1.4238614e-03
 [2,] 1.8402798e-05 1.4238614e-03 1.0670172e-02
 [3,] 1.4238614e-03 1.0670172e-02 3.1874767e-04
 [4,] 1.0670172e-02 3.1874767e-04 4.6755673e-05
 [5,] 3.1874767e-04 4.6755673e-05 4.9397065e-05
 [6,] 4.6755673e-05 4.9397065e-05 1.3792232e-03
 [7,] 4.9397065e-05 1.3792232e-03 2.0208402e-02
 [8,] 1.3792232e-03 2.0208402e-02 4.5935880e-02
 [9,] 2.0208402e-02 4.5935880e-02 3.9260377e-02
[10,] 4.5935880e-02 3.9260377e-02 1.5561811e-01
```

Keterangan :

List pertama, menyatakan nilai derajat keanggotaan setiap variabel *input* pada *rule base* ke-1.

List kedua, menyatakan nilai derajat keanggotaan setiap variabel *input* pada *rule base* ke-2.

....

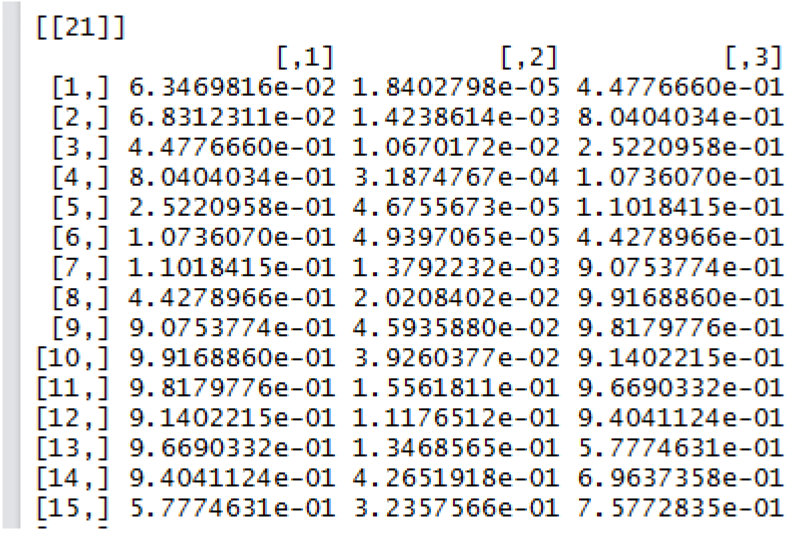


```
[83,] 6.4306926e-01 9.0495655e-01 4.1524180e-01
[84,] 9.0495655e-01 4.1524180e-01 7.5927095e-01
[85,] 4.1524180e-01 7.5927095e-01 7.2919949e-01
[86,] 7.5927095e-01 7.2919949e-01 6.8833262e-01
[87,] 7.2919949e-01 6.8833262e-01 6.7139152e-01
[88,] 6.8833262e-01 6.7139152e-01 4.6267749e-01
[89,] 6.7139152e-01 4.6267749e-01 1.4524595e-01
[90,] 4.6267749e-01 1.4524595e-01 2.1167615e-01
[91,] 1.4524595e-01 2.1167615e-01 3.2688392e-01
[92,] 2.1167615e-01 3.2688392e-01 2.8555936e-01
[93,] 3.2688392e-01 2.8555936e-01 6.4854897e-01
[94,] 2.8555936e-01 6.4854897e-01 4.1286047e-01
[95,] 6.4854897e-01 4.1286047e-01 5.8652718e-02
[96,] 4.1286047e-01 5.8652718e-02 3.6026628e-02
[97,] 5.8652718e-02 3.6026628e-02 8.0765829e-04
```

dan seterusnya, . . .

....

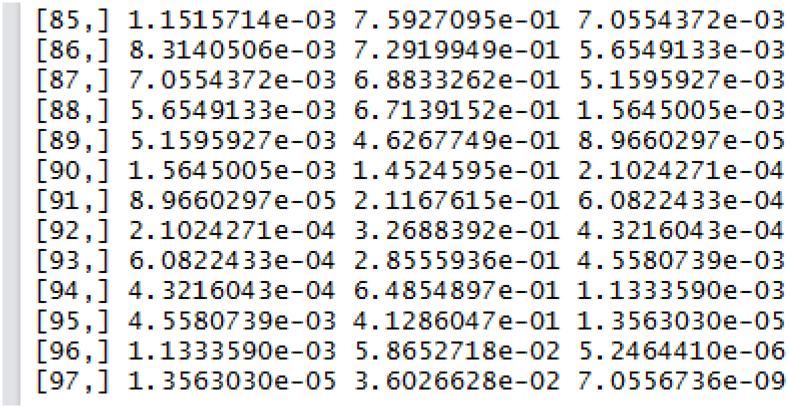
....



```
[[21]]
               [,1]         [,2]         [,3]
 [1,] 6.3469816e-02 1.8402798e-05 4.4776660e-01
 [2,] 6.8312311e-02 1.4238614e-03 8.0404034e-01
 [3,] 4.4776660e-01 1.0670172e-02 2.5220958e-01
 [4,] 8.0404034e-01 3.1874767e-04 1.0736070e-01
 [5,] 2.5220958e-01 4.6755673e-05 1.1018415e-01
 [6,] 1.0736070e-01 4.9397065e-05 4.4278966e-01
 [7,] 1.1018415e-01 1.3792232e-03 9.0753774e-01
 [8,] 4.4278966e-01 2.0208402e-02 9.9168860e-01
 [9,] 9.0753774e-01 4.5935880e-02 9.8179776e-01
[10,] 9.9168860e-01 3.9260377e-02 9.1402215e-01
[11,] 9.8179776e-01 1.5561811e-01 9.6690332e-01
[12,] 9.1402215e-01 1.1176512e-01 9.4041124e-01
[13,] 9.6690332e-01 1.3468565e-01 5.7774631e-01
[14,] 9.4041124e-01 4.2651918e-01 6.9637358e-01
[15,] 5.7774631e-01 3.2357566e-01 7.5772835e-01
```

List kedua puluh satu, menyatakan nilai derajat keanggotaan setiap variabel *input* pada *rule base* ke-21.

....



```
[85,] 1.1515714e-03 7.5927095e-01 7.0554372e-03
[86,] 8.3140506e-03 7.2919949e-01 5.6549133e-03
[87,] 7.0554372e-03 6.8833262e-01 5.1595927e-03
[88,] 5.6549133e-03 6.7139152e-01 1.5645005e-03
[89,] 5.1595927e-03 4.6267749e-01 8.9660297e-05
[90,] 1.5645005e-03 1.4524595e-01 2.1024271e-04
[91,] 8.9660297e-05 2.1167615e-01 6.0822433e-04
[92,] 2.1024271e-04 3.2688392e-01 4.3216043e-04
[93,] 6.0822433e-04 2.8555936e-01 4.5580739e-03
[94,] 4.3216043e-04 6.4854897e-01 1.1333590e-03
[95,] 4.5580739e-03 4.1286047e-01 1.3563030e-05
[96,] 1.1333590e-03 5.8652718e-02 5.2464410e-06
[97,] 1.3563030e-05 3.6026628e-02 7.0556736e-09
```

- ✓ Kemudian dari hasil sebelumnya maka dicari *fire strength rule (Alfa Cut)*. *Fire strength rule* dihitung dengan operator perkalian, yaitu :

$$\alpha_t^l = \prod_{k=1}^{p} \mu_{A_k^l}(y_{t,k})$$

dimana :

l : Banyaknya *rule*

t : Banyaknya amatan

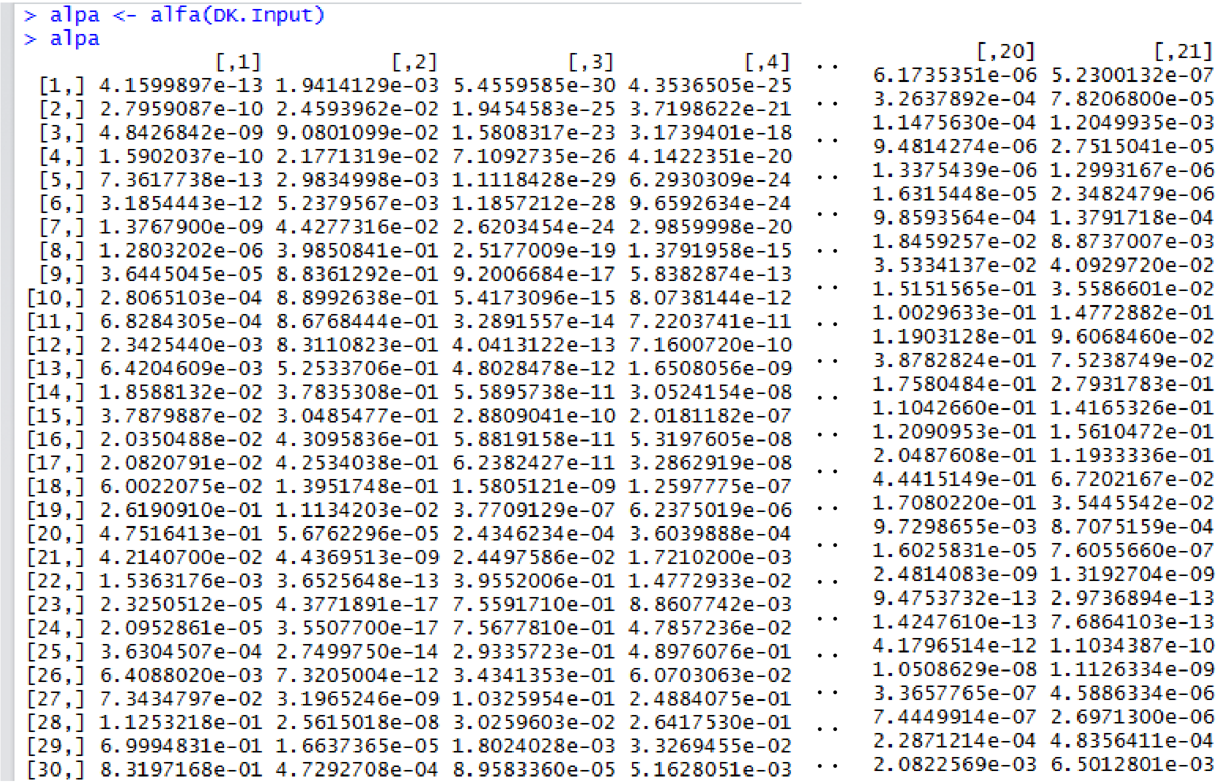
p : Banyaknya variabel *input*

Berikut ini adalah *script* pada *software* R untuk memperoleh *fire strength rule (Alfa Cut)* :

```
-----------------------------------------------------------------
#mencari alfa cut atau fire strength
#Keterangan :
#ipt -> nilai derajat keanggotaan setiap rule base dari data input
yang berupa suatu list
-----------------------------------------------------------------
```

```
alfa <- function(ipt) {
n <- nrow(ipt[[1]])
m <- length(ipt)
out <- matrix(0,n,m)
for(i in 1:m) {
     for(j in 1:n) {
          out[j,i] <- prod(ipt[[i]][j,])
          }
     }
alfa <- out
}
```

Sehingga diperoleh *Alfa Cut* yang berupa suatu matriks berukuran $97x21$ (banyaknya data dikali banyaknya *rule base*). Hasil dari *software* R sebagai berikut :

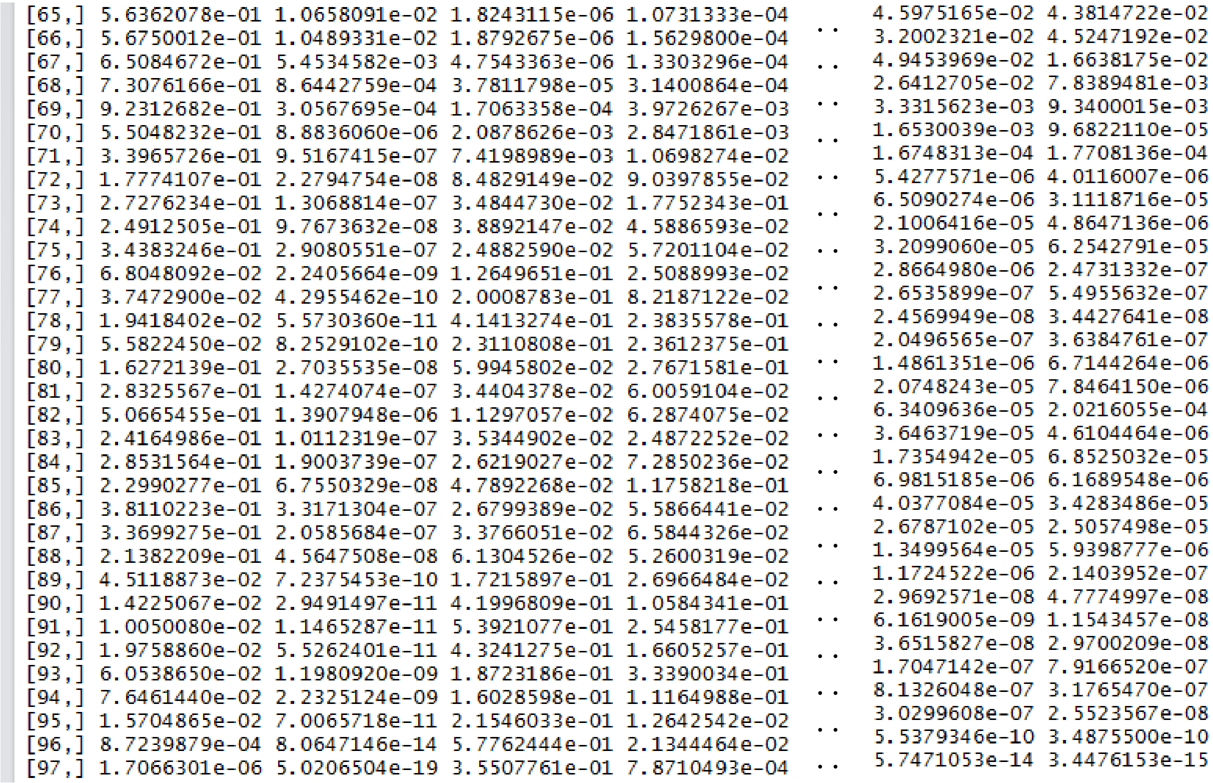


```
> alpa <- alfa(DK.Input)
> alpa
               [,1]          [,2]          [,3]          [,4]   ..           [,20]          [,21]
 [1,] 4.1599897e-13 1.9414129e-03 5.4559585e-30 4.3536505e-25      6.1735351e-06 5.2300132e-07
 [2,] 2.7959087e-10 2.4593962e-02 1.9454583e-25 3.7198622e-21   ..  3.2637892e-04 7.8206800e-05
 [3,] 4.8426842e-09 9.0801099e-02 1.5808317e-23 3.1739401e-18   ..  1.1475630e-04 1.2049935e-03
 [4,] 1.5902037e-10 2.1771319e-02 7.1092735e-26 4.1422351e-20      9.4814274e-06 2.7515041e-05
 [5,] 7.3617738e-13 2.9834998e-03 1.1118428e-29 6.2930309e-24   ..  1.3375439e-06 1.2993167e-06
 [6,] 3.1854443e-12 5.2379567e-03 1.1857212e-28 9.6592634e-24   ..  1.6315448e-05 2.3482479e-06
 [7,] 1.3767900e-09 4.4277316e-02 2.6203454e-24 2.9859998e-20      9.8593564e-04 1.3791718e-04
 [8,] 1.2803202e-06 3.9850841e-01 2.5177009e-19 1.3791958e-15   ..  1.8459257e-02 8.8737007e-03
 [9,] 3.6445045e-05 8.8361292e-01 9.2006684e-17 5.8382874e-13      3.5334137e-02 4.0929720e-02
[10,] 2.8065103e-04 8.8992638e-01 5.4173096e-15 8.0738144e-12   ..  1.5151565e-01 3.5586601e-02
[11,] 6.8284305e-04 8.6768444e-01 3.2891557e-14 7.2203741e-11   ..  1.0029633e-01 1.4772882e-01
[12,] 2.3425440e-03 8.3110823e-01 4.0413122e-13 7.1600720e-10      1.1903128e-01 9.6068460e-02
[13,] 6.4204609e-03 5.2533706e-01 4.8028478e-12 1.6508056e-09   ..  3.8782824e-01 7.5238749e-02
[14,] 1.8588132e-02 3.7835308e-01 5.5895738e-11 3.0524154e-08   ..  1.7580484e-01 2.7931783e-01
[15,] 3.7879887e-02 3.0485477e-01 2.8809041e-10 2.0181182e-07      1.1042660e-01 1.4165326e-01
[16,] 2.0350488e-02 4.3095836e-01 5.8819158e-11 5.3197605e-08   ..  1.2090953e-01 1.5610472e-01
[17,] 2.0820791e-02 4.2534038e-01 6.2382427e-11 3.2862919e-08   ..  2.0487608e-01 1.1933336e-01
[18,] 6.0022075e-02 1.3951748e-01 1.5805121e-09 1.2597775e-07      4.4415149e-01 6.7202167e-02
[19,] 2.6190910e-01 1.1134203e-02 3.7709129e-07 6.2375019e-06   ..  1.7080220e-01 3.5445542e-02
[20,] 4.7516413e-01 5.6762296e-05 2.4346234e-04 3.6039888e-04   ..  9.7298655e-03 8.7075159e-04
[21,] 4.2140700e-02 4.4369513e-09 2.4497586e-02 1.7210200e-03      1.6025831e-05 7.6055660e-07
[22,] 1.5363176e-03 3.6525648e-13 3.9552006e-01 1.4772933e-02   ..  2.4814083e-09 1.3192704e-09
[23,] 2.3250512e-05 4.3771891e-17 7.5591710e-01 8.8607742e-03      9.4753732e-13 2.9736894e-13
[24,] 2.0952861e-05 3.5507700e-17 7.5677810e-01 4.7857236e-02   ..  1.4247610e-13 7.6864103e-13
[25,] 3.6304507e-04 2.7499750e-14 2.9335723e-01 4.8976076e-01   ..  4.1796514e-12 1.1034387e-10
[26,] 6.4088020e-03 7.3205004e-12 3.4341353e-01 6.0703063e-02      1.0508629e-08 1.1126334e-09
[27,] 7.3434797e-02 3.1965246e-09 1.0325954e-01 2.4884075e-01   ..  3.3657765e-07 4.5886334e-06
[28,] 1.1253218e-01 2.5615018e-08 3.0259603e-02 2.6417530e-01   ..  7.4449914e-07 2.6971300e-06
[29,] 6.9994831e-01 1.6637365e-05 1.8024028e-03 3.3269455e-02      2.2871214e-04 4.8356411e-04
[30,] 8.3197168e-01 4.7292708e-04 8.9583360e-05 5.1628051e-03   ..  2.0822569e-03 6.5012801e-03
```

....

....

....



```
[65,] 5.6362078e-01 1.0658091e-02 1.8243115e-06 1.0731333e-04       4.5975165e-02 4.3814722e-02
[66,] 5.6750012e-01 1.0489331e-02 1.8792675e-06 1.5629800e-04  ..   3.2002321e-02 4.5247192e-02
[67,] 6.5084672e-01 5.4534582e-03 4.7543363e-06 1.3303296e-04  ..   4.9453969e-02 1.6638175e-02
[68,] 7.3076166e-01 8.6442759e-04 3.7811798e-05 3.1400864e-04       2.6412705e-02 7.8389481e-03
[69,] 9.2312682e-01 3.0567695e-04 1.7063358e-04 3.9726267e-03  ..   3.3315623e-03 9.3400015e-03
[70,] 5.5048232e-01 8.8836060e-06 2.0878626e-03 2.8471861e-03  ..   1.6530039e-03 9.6822110e-05
[71,] 3.3965726e-01 9.5167415e-07 7.4198989e-03 1.0698274e-02       1.6748313e-04 1.7708136e-04
[72,] 1.7774107e-01 2.2794754e-08 8.4829149e-02 9.0397855e-02  ..   5.4277571e-06 4.0116007e-06
[73,] 2.7276234e-01 1.3068814e-07 3.4844730e-02 1.7752343e-01       6.5090274e-06 3.1118716e-05
[74,] 2.4912505e-01 9.7673632e-08 3.8892147e-02 4.5886593e-02  ..   2.1006416e-05 4.8647136e-06
[75,] 3.4383246e-01 2.9080551e-07 2.4882590e-02 5.7201104e-02  ..   3.2099060e-05 6.2542791e-05
[76,] 6.8048092e-02 2.2405664e-09 1.2649651e-01 2.5088993e-02       2.8664980e-06 2.4731332e-07
[77,] 3.7472900e-02 4.2955462e-10 2.0008783e-01 8.2187122e-02  ..   2.6535899e-07 5.4955632e-07
[78,] 1.9418402e-02 5.5730360e-11 4.1413274e-01 2.3835578e-01  ..   2.4569949e-08 3.4427641e-08
[79,] 5.5822450e-02 8.2529102e-10 2.3110808e-01 2.3612375e-01       2.0496565e-07 3.6384761e-07
[80,] 1.6272139e-01 2.7035535e-08 5.9945802e-02 2.7671581e-01  ..   1.4861351e-06 6.7144264e-06
[81,] 2.8325567e-01 1.4274074e-07 3.4404378e-02 6.0059104e-02  ..   2.0748243e-05 7.8464150e-06
[82,] 5.0665455e-01 1.3907948e-06 1.1297057e-02 6.2874075e-02       6.3409636e-05 2.0216055e-04
[83,] 2.4164986e-01 1.0112319e-07 3.5344902e-02 2.4872252e-02  ..   3.6463719e-05 4.6104464e-06
[84,] 2.8531564e-01 1.9003739e-07 2.6219027e-02 7.2850236e-02  ..   1.7354942e-05 6.8525032e-05
[85,] 2.2990277e-01 6.7550329e-08 4.7892268e-02 1.1758218e-01       6.9815185e-06 6.1689548e-06
[86,] 3.8110223e-01 3.3171304e-07 2.6799389e-02 5.5866441e-02  ..   4.0377084e-05 3.4283486e-05
[87,] 3.3699275e-01 2.0585684e-07 3.3766051e-02 6.5844326e-02       2.6787102e-05 2.5057498e-05
[88,] 2.1382209e-01 4.5647508e-08 6.1304526e-02 5.2600319e-02  ..   1.3499564e-05 5.9398777e-06
[89,] 4.5118873e-02 7.2375453e-10 1.7215897e-01 2.6966484e-02  ..   1.1724522e-06 2.1403952e-07
[90,] 1.4225067e-02 2.9491497e-11 4.1996809e-01 1.0584341e-01       2.9692571e-08 4.7774997e-08
[91,] 1.0050080e-02 1.1465287e-11 5.3921077e-01 2.5458177e-01  ..   6.1619005e-09 1.1543457e-08
[92,] 1.9758860e-02 5.5262401e-11 4.3241275e-01 1.6605257e-01  ..   3.6515827e-08 2.9700209e-08
[93,] 6.0538650e-02 1.1980920e-09 1.8723186e-01 3.3390034e-01       1.7047142e-07 7.9166520e-07
[94,] 7.6461440e-02 2.2325124e-09 1.6028598e-01 1.1164988e-01  ..   8.1326048e-07 3.1765470e-07
[95,] 1.5704865e-02 7.0065718e-11 2.1546033e-01 1.2642542e-02  ..   3.0299608e-07 2.5523567e-08
[96,] 8.7239879e-04 8.0647146e-14 5.7762444e-01 2.1344464e-02       5.5379346e-10 3.4875500e-10
[97,] 1.7066301e-06 5.0206504e-19 3.5507761e-01 7.8710493e-04  ..   5.7471053e-14 3.4476153e-15
```

✓ Selanjutnya sebelum menghitung *output* nilai tegas, *fire strength* terlebih dahulu dinormalisasi, yakni:

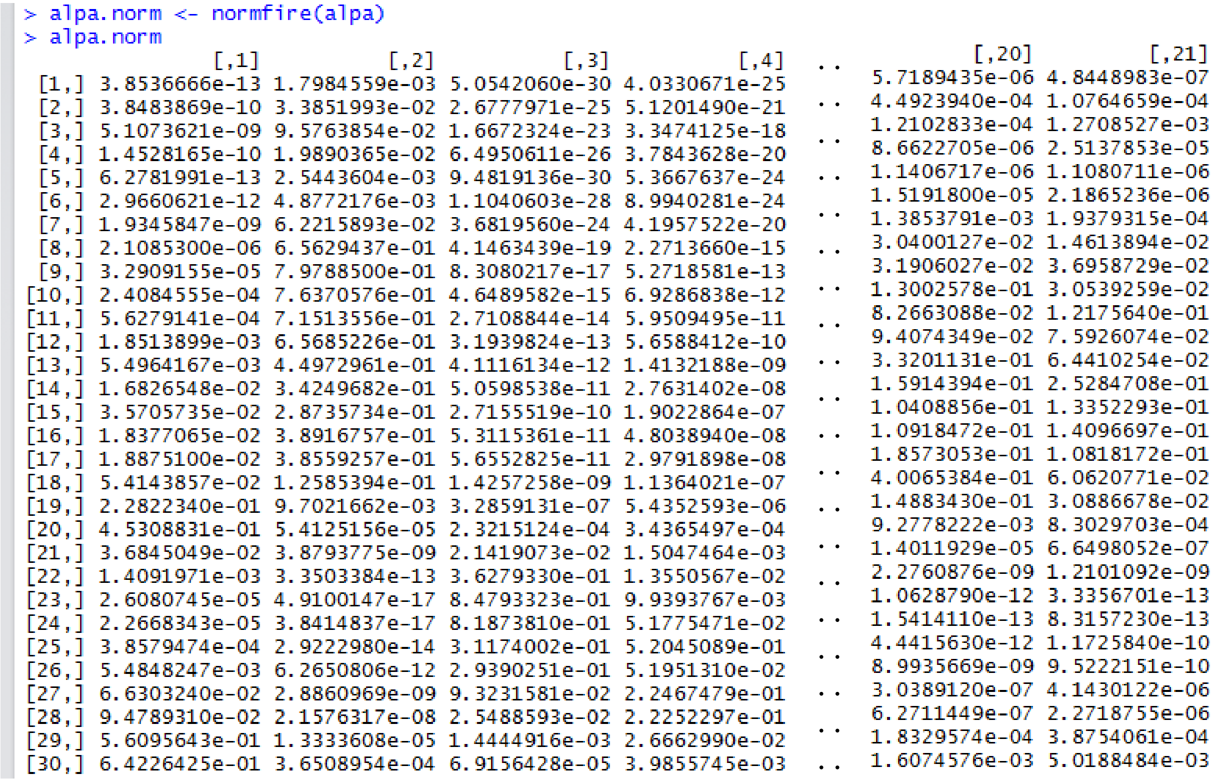
$$\bar{\alpha}_t^l = \frac{\alpha_t^l}{\sum_{l=1}^{L} \alpha_t^l} = \frac{\prod_{k=1}^{p} \mu_{A_k^l}(y_{t,k})}{\sum_{l=1}^{L} \prod_{k=1}^{p} \mu_{A_k^l}(y_{t,k})}$$

Berikut ini adalah *script* pada *software* R untuk menormalisasi *fire strength* atau *alfa cut* :

```
---------------------------------------------------------------
#Normalisasi fire strength
#Keterangan :
#alfa -> nilai alfa cut atau fire strength yang berupa suatu
matriks
---------------------------------------------------------------
normfire <- function(alfa) {
n <- nrow(alfa)
m <- ncol(alfa)
out <- matrix(0,n,m)
for(i in 1:n) {
   jum <- sum(alfa[i,])
   for(j in 1:m) {
        out[i,j] <- alfa[i,j]/jum
        }
```

```
    }
normfire <- out
}
```

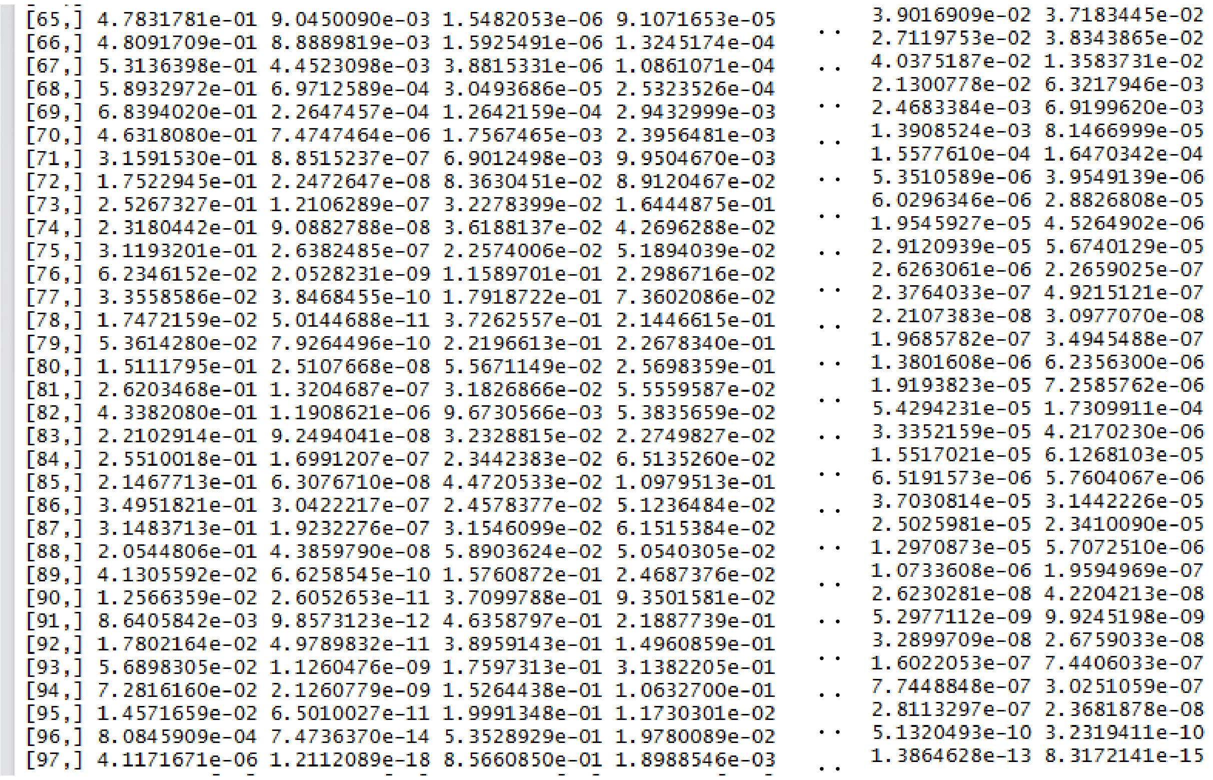
Hasil dari *software* R sebagai berikut :



```
> alpa.norm <- normfire(alpa)
> alpa.norm
               [,1]         [,2]         [,3]         [,4]   ..         [,20]         [,21]
 [1,] 3.8536666e-13 1.7984559e-03 5.0542060e-30 4.0330671e-25 .. 5.7189435e-06 4.8448983e-07
 [2,] 3.8483869e-10 3.3851993e-02 2.6777971e-25 5.1201490e-21 .. 4.4923940e-04 1.0764659e-04
 [3,] 5.1073621e-09 9.5763854e-02 1.6672324e-23 3.3474125e-18 .. 1.2102833e-04 1.2708527e-03
 [4,] 1.4528165e-10 1.9890365e-02 6.4950611e-26 3.7843628e-20 .. 8.6622705e-06 2.5137853e-05
 [5,] 6.2781991e-13 2.5443604e-03 9.4819136e-30 5.3667637e-24 .. 1.1406717e-06 1.1080711e-06
 [6,] 2.9660621e-12 4.8772176e-03 1.1040603e-28 8.9940281e-24 .. 1.5191800e-05 2.1865236e-06
 [7,] 1.9345847e-09 6.2215893e-02 3.6819560e-24 4.1957522e-20 .. 1.3853791e-03 1.9379315e-04
 [8,] 2.1085300e-06 6.5629437e-01 4.1463439e-19 2.2713660e-15 .. 3.0400127e-02 1.4613894e-02
 [9,] 3.2909155e-05 7.9788500e-01 8.3080217e-17 5.2718581e-13 .. 3.1906027e-02 3.6958729e-02
[10,] 2.4084555e-04 7.6370576e-01 4.6489582e-15 6.9286838e-12 .. 1.3002578e-01 3.0539259e-02
[11,] 5.6279141e-04 7.1513556e-01 2.7108844e-14 5.9509495e-11 .. 8.2663088e-02 1.2175640e-01
[12,] 1.8513899e-03 6.5685226e-01 3.1939824e-13 5.6588412e-10 .. 9.4074349e-02 7.5926074e-02
[13,] 5.4964167e-03 4.4972961e-01 4.1116134e-12 1.4132188e-09 .. 3.3201131e-01 6.4410254e-02
[14,] 1.6826548e-02 3.4249682e-01 5.0598538e-11 2.7631402e-08 .. 1.5914394e-01 2.5284708e-01
[15,] 3.5705735e-02 2.8735734e-01 2.7155519e-10 1.9022864e-07 .. 1.0408856e-01 1.3352293e-01
[16,] 1.8377065e-02 3.8916757e-01 5.3115361e-11 4.8038940e-08 .. 1.0918472e-01 1.4096697e-01
[17,] 1.8875100e-02 3.8559257e-01 5.6552825e-11 2.9791898e-08 .. 1.8573053e-01 1.0818172e-01
[18,] 5.4143857e-02 1.2585394e-01 1.4257258e-09 1.1364021e-07 .. 4.0065384e-01 6.0620771e-02
[19,] 2.2822340e-01 9.7021662e-03 3.2859131e-07 5.4352593e-06 .. 1.4883430e-01 3.0886678e-02
[20,] 4.5308831e-01 5.4125156e-05 2.3215124e-04 3.4365497e-04 .. 9.2778222e-03 8.3029703e-04
[21,] 3.6845049e-02 3.8793775e-09 2.1419073e-02 1.5047464e-03 .. 1.4011929e-05 6.6498052e-07
[22,] 1.4091971e-03 3.3503384e-13 3.6279330e-01 1.3550567e-02 .. 2.2760876e-09 1.2101092e-09
[23,] 2.6080745e-05 4.9100147e-17 8.4793323e-01 9.9393767e-03 .. 1.0628790e-12 3.3356701e-13
[24,] 2.2668343e-05 3.8414837e-17 8.1873810e-01 5.1775471e-02 .. 1.5414110e-13 8.3157230e-13
[25,] 3.8579474e-04 2.9222980e-14 3.1174002e-01 5.2045089e-01 .. 4.4415630e-12 1.1725840e-10
[26,] 5.4848247e-03 6.2650806e-12 2.9390251e-01 5.1951310e-02 .. 8.9935669e-09 9.5222151e-10
[27,] 6.6303240e-02 2.8860969e-09 9.3231581e-02 2.2467479e-01 .. 3.0389120e-07 4.1430122e-06
[28,] 9.4789310e-02 2.1576317e-08 2.5488593e-02 2.2252297e-01 .. 6.2711449e-07 2.2718755e-06
[29,] 5.6095643e-01 1.3333608e-05 1.4444916e-03 2.6662990e-02 .. 1.8329574e-04 3.8754061e-04
[30,] 6.4226425e-01 3.6508954e-04 6.9156428e-05 3.9855745e-03 .. 1.6074576e-03 5.0188484e-03
```

....

....

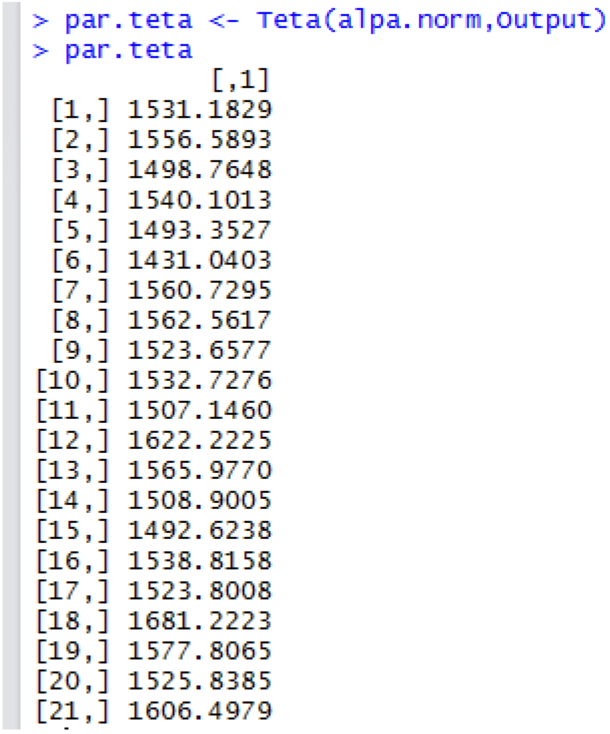


```
[65,] 4.7831781e-01 9.0450090e-03 1.5482053e-06 9.1071653e-05 .. 3.9016909e-02 3.7183445e-02
[66,] 4.8091709e-01 8.8889819e-03 1.5925491e-06 1.3245174e-04 .. 2.7119753e-02 3.8343865e-02
[67,] 5.3136398e-01 4.4523098e-03 3.8815331e-06 1.0861071e-04 .. 4.0375187e-02 1.3583731e-02
[68,] 5.8932972e-01 6.9712589e-04 3.0493686e-05 2.5323526e-04 .. 2.1300778e-02 6.3217946e-03
[69,] 6.8394020e-01 2.2647457e-04 1.2642159e-04 2.9432999e-03 .. 2.4683384e-03 6.9199620e-03
[70,] 4.6318080e-01 7.4747464e-06 1.7567465e-03 2.3956481e-03 .. 1.3908524e-03 8.1466999e-05
[71,] 3.1591530e-01 8.8515237e-07 6.9012498e-03 9.9504670e-03 .. 1.5577610e-04 1.6470342e-04
[72,] 1.7522945e-01 2.2472647e-08 8.3630451e-02 8.9120467e-02 .. 5.3510589e-06 3.9549139e-06
[73,] 2.5267327e-01 1.2106289e-07 3.2278399e-02 1.6444875e-01 .. 6.0296346e-06 2.8826808e-05
[74,] 2.3180442e-01 9.0882788e-08 3.6188137e-02 4.2696288e-02 .. 1.9545927e-05 4.5264902e-06
[75,] 3.1193201e-01 2.6382485e-07 2.2574006e-02 5.1894039e-02 .. 2.9120939e-05 5.6740129e-05
[76,] 6.2346152e-02 2.0528231e-09 1.1589701e-01 2.2986716e-02 .. 2.6263061e-06 2.2659025e-07
[77,] 3.3558586e-02 3.8468455e-10 1.7918722e-01 7.3602086e-02 .. 2.3764033e-07 4.9215121e-07
[78,] 1.7472159e-02 5.0144688e-11 3.7262557e-01 2.1446615e-01 .. 2.2107383e-08 3.0977070e-08
[79,] 5.3614280e-02 7.9264496e-10 2.2196613e-01 2.2678340e-01 .. 1.9685782e-07 3.4945488e-07
[80,] 1.5111795e-01 2.5107668e-08 5.5671149e-02 2.5698359e-01 .. 1.3801608e-06 6.2356300e-06
[81,] 2.6203468e-01 1.3204687e-07 3.1826866e-02 5.5559587e-02 .. 1.9193823e-05 7.2585762e-06
[82,] 4.3382080e-01 1.1908621e-06 9.6730566e-03 5.3835659e-02 .. 5.4294231e-05 1.7309911e-04
[83,] 2.2102914e-01 9.2494041e-08 3.2328815e-02 2.2749827e-02 .. 3.3352159e-05 4.2170230e-06
[84,] 2.5510018e-01 1.6991207e-07 2.3442383e-02 6.5135260e-02 .. 1.5517021e-05 6.1268103e-05
[85,] 2.1467713e-01 6.3076710e-08 4.4720533e-02 1.0979513e-01 .. 6.5191573e-06 5.7604067e-06
[86,] 3.4951821e-01 3.0422217e-07 2.4578377e-02 5.1236484e-02 .. 3.7030814e-05 3.1442226e-05
[87,] 3.1483713e-01 1.9232276e-07 3.1546099e-02 6.1515384e-02 .. 2.5025981e-05 2.3410090e-05
[88,] 2.0544806e-01 4.3859790e-08 5.8903624e-02 5.0540305e-02 .. 1.2970873e-05 5.7072510e-06
[89,] 4.1305592e-02 6.6258545e-10 1.5760872e-01 2.4687376e-02 .. 1.0733608e-06 1.9594969e-07
[90,] 1.2566359e-02 2.6052653e-11 3.7099788e-01 9.3501581e-02 .. 2.6230281e-08 4.2204213e-08
[91,] 8.6405842e-03 9.8573123e-12 4.6358797e-01 2.1887739e-01 .. 5.2977112e-09 9.9245198e-09
[92,] 1.7802164e-02 4.9789832e-11 3.8959143e-01 1.4960859e-01 .. 3.2899709e-08 2.6759033e-08
[93,] 5.6898305e-02 1.1260476e-09 1.7597313e-01 3.1382205e-01 .. 1.6022053e-07 7.4406033e-07
[94,] 7.2816160e-02 2.1260779e-09 1.5264438e-01 1.0632700e-01 .. 7.7448848e-07 3.0251059e-07
[95,] 1.4571659e-02 6.5010027e-11 1.9991348e-01 1.1730301e-02 .. 2.8113297e-07 2.3681878e-08
[96,] 8.0845909e-04 7.4736370e-14 5.3528929e-01 1.9780089e-02 .. 5.1320493e-10 3.2319411e-10
[97,] 4.1171671e-06 1.2112089e-18 8.5660850e-01 1.8988546e-03 .. 1.3864628e-13 8.3172141e-15
```

- ✓ Selanjutnya menduga parameter konsekuen. Proses pendugaan parameter konsekuen diestimasi menggunakan metode kuadrat terkecil (MKT). Berikut ini adalah *script* pada *software* R untuk memperoleh parameter konsekuen :

```
----------------------------------------------------------------
#Mencari Teta
#Keterangan :
#normalfa -> nilai fire strength yang telah dinormalisasi yang
berupa suatu matriks
#Data -> data output yang berupa suatu matriks
----------------------------------------------------------------
Teta <- function(normalfa,Data) {
n <- length(Data)
X <- normalfa
Teta <- solve(crossprod(X,X))%*%t(X)%*%Data
}
```

Hasil dari *software* R sebagai berikut :



```
> par.teta <- Teta(alpa.norm,Output)
> par.teta
           [,1]
 [1,] 1531.1829
 [2,] 1556.5893
 [3,] 1498.7648
 [4,] 1540.1013
 [5,] 1493.3527
 [6,] 1431.0403
 [7,] 1560.7295
 [8,] 1562.5617
 [9,] 1523.6577
[10,] 1532.7276
[11,] 1507.1460
[12,] 1622.2225
[13,] 1565.9770
[14,] 1508.9005
[15,] 1492.6238
[16,] 1538.8158
[17,] 1523.8008
[18,] 1681.2223
[19,] 1577.8065
[20,] 1525.8385
[21,] 1606.4979
```

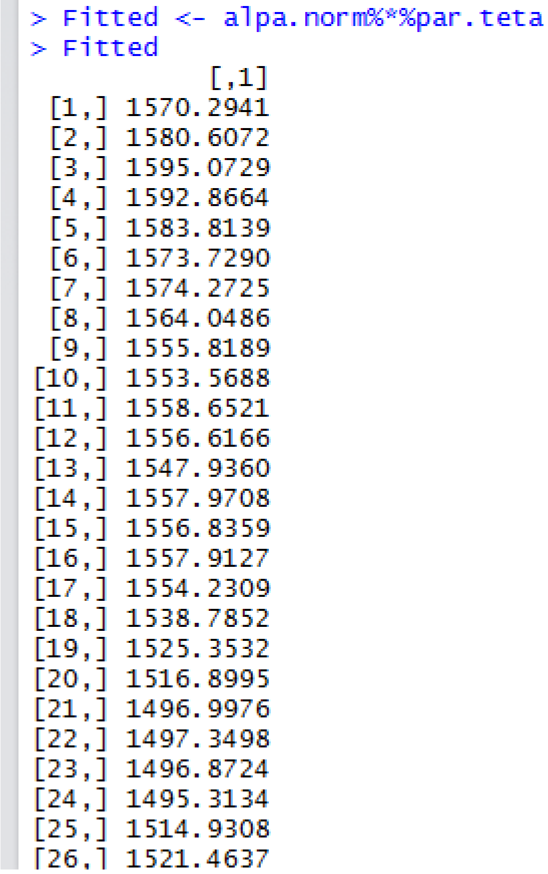
- ✓ Selanjutnya menduga *output* nilai tegas $\hat{y}_t$ yang merupakan nilai prediksi pada pengamatan ke-t dihitung dengan :

$$\hat{y}_t = \sum_{l=1}^{L} \bar{\alpha}_t^l \, \hat{y}_t = \sum_{l=1}^{L} \bar{\alpha}_t^l \, (\theta_0^l + \theta_1^l y_{t,1} + \cdots + \theta_p^l y_{t,p})$$

Karena pada kasus ini digunakan *fuzzy* takagi-sugeno orde nol, maka :

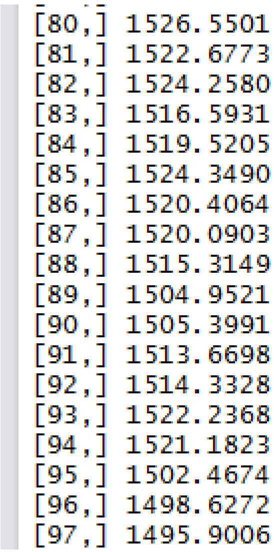
$$\hat{y}_t = \sum_{l=1}^{L} \bar{\alpha}_t^l \, \theta_0^l$$

Berikut hasil dari *software* R :



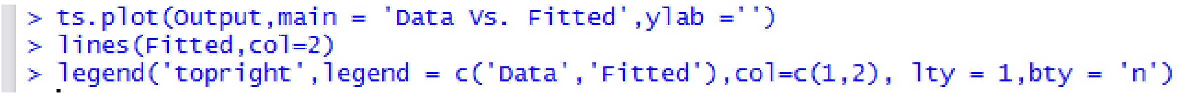
```
> Fitted <- alpa.norm%*%par.teta
> Fitted
          [,1]
 [1,] 1570.2941
 [2,] 1580.6072
 [3,] 1595.0729
 [4,] 1592.8664
 [5,] 1583.8139
 [6,] 1573.7290
 [7,] 1574.2725
 [8,] 1564.0486
 [9,] 1555.8189
[10,] 1553.5688
[11,] 1558.6521
[12,] 1556.6166
[13,] 1547.9360
[14,] 1557.9708
[15,] 1556.8359
[16,] 1557.9127
[17,] 1554.2309
[18,] 1538.7852
[19,] 1525.3532
[20,] 1516.8995
[21,] 1496.9976
[22,] 1497.3498
[23,] 1496.8724
[24,] 1495.3134
[25,] 1514.9308
[26,] 1521.4637
```

………………………………

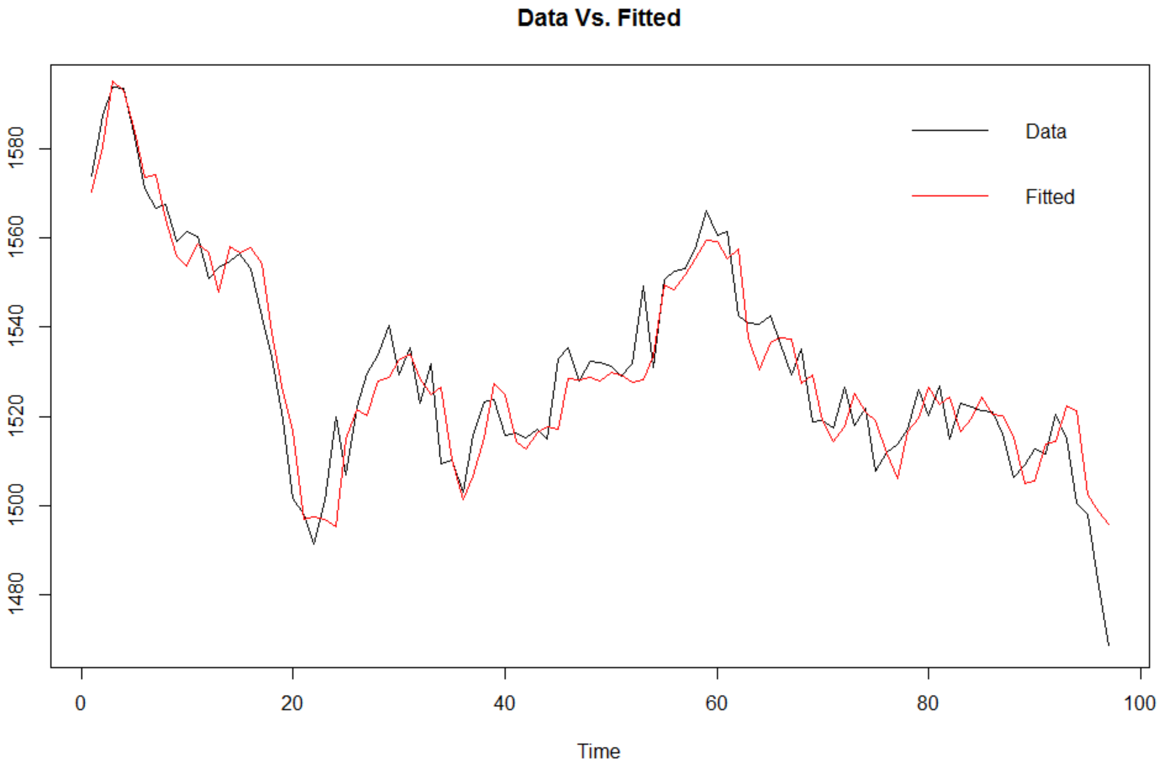


```
[80,] 1526.5501
[81,] 1522.6773
[82,] 1524.2580
[83,] 1516.5931
[84,] 1519.5205
[85,] 1524.3490
[86,] 1520.4064
[87,] 1520.0903
[88,] 1515.3149
[89,] 1504.9521
[90,] 1505.3991
[91,] 1513.6698
[92,] 1514.3328
[93,] 1522.2368
[94,] 1521.1823
[95,] 1502.4674
[96,] 1498.6272
[97,] 1495.9006
```

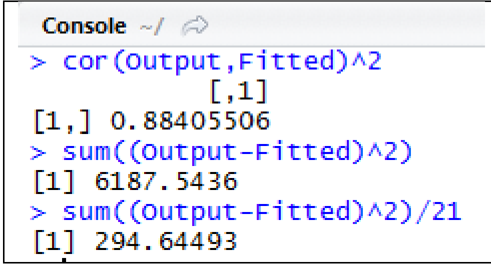
✓ Berikut plot data aktual versus nilai *fitted* :



```
> ts.plot(Output,main = 'Data Vs. Fitted',ylab ='')
> lines(Fitted,col=2)
> legend('topright',legend = c('Data','Fitted'),col=c(1,2), lty = 1,bty = 'n')
```



✓ Untuk melihat kabaikan prediksi dengan metode *fuzzy* takagi-sugeno, maka kita mencari nilai R^2, SSE (*Sum Square Error*), dan MSE (*Mean Square Error*). Berikut hasil pada *software* R :

 Console ~/	***R^2***	***SSE***	***MSE***
> cor(Output,Fitted)^2 [,1] [1,] 0.88405506 > sum((Output-Fitted)^2) [1] 6187.5436 > sum((Output-Fitted)^2)/21 [1] 294.64493	88.4055%	6187.5436	294.6449

- Metode *fuzzy* takagi-sugeno dengan orde nol didapatkan nilai R^2 sebesar 88.4055%. Ini menandakan metode yang digunakan sangat baik dalam menjelaskan keragaman data yang ada. Penggunaan orde yang lebih tinggi mungkin akan memberikan hasil yang lebih baik.
- MSE (*Mean Square Error*) yang diperoleh yaitu sebesar 294.6449. Nilai ini sangat kecil apabila dibandingkan dengan besaran data aktual yang berkisar antara 1468.7926 sampai 1596.9365 dengan banyaknya amatan adalah 100 amatan.

Lampiran 1. Data Aktual & *Fitted*

Tanggal	***Kurs***	***Fitted***	***Tanggal***	***Kurs***	***Fitted***
01/01/2015	1596.9365	-	*20/02/2015*	1532.3782	1528.8400
02/01/2015	1596.5078	-	*21/02/2015*	1532.1574	1527.8723
03/01/2015	1582.2402	-	*22/02/2015*	1531.3465	1529.7293
04/01/2015	1573.9823	1570.2941	*23/02/2015*	1528.8916	1529.2167
05/01/2015	1587.5735	1580.6072	*24/02/2015*	1531.6619	1527.6524
06/01/2015	1593.7315	1595.0729	*25/02/2015*	1549.1531	1528.3198
07/01/2015	1593.5639	1592.8664	*26/02/2015*	1531.059	1533.4763
08/01/2015	1582.3597	1583.8139	*27/02/2015*	1550.6478	1549.6053
09/01/2015	1570.9908	1573.7290	*28/02/2015*	1552.4525	1548.3575
10/01/2015	1566.7571	1574.2725	*01/03/2015*	1553.1833	1551.7529
11/01/2015	1567.6068	1564.0486	*02/03/2015*	1557.4921	1555.4153
12/01/2015	1559.2332	1555.8189	*03/03/2015*	1566.0803	1559.3997
13/01/2015	1561.4715	1553.5688	*04/03/2015*	1560.6493	1559.3224
14/01/2015	1560.2327	1558.6521	*05/03/2015*	1561.4128	1555.3913
15/01/2015	1550.7541	1556.6166	*06/03/2015*	1542.6846	1557.4686
16/01/2015	1553.4301	1547.9360	*07/03/2015*	1540.8291	1537.7234
17/01/2015	1554.8695	1557.9708	*08/03/2015*	1540.5509	1530.3135
18/01/2015	1556.3462	1556.8359	*09/03/2015*	1542.5527	1536.4128
19/01/2015	1553.2222	1557.9127	*10/03/2015*	1536.2563	1537.7262
20/01/2015	1542.3069	1554.2309	*11/03/2015*	1529.2351	1537.2976
21/01/2015	1533.2178	1538.7852	*12/03/2015*	1535.1935	1527.3979
22/01/2015	1519.2672	1525.3532	*13/03/2015*	1518.7929	1529.2079
23/01/2015	1501.6503	1516.8995	*14/03/2015*	1519.115	1519.1343
24/01/2015	1497.9988	1496.9976	*15/03/2015*	1517.3675	1514.2711
25/01/2015	1491.3004	1497.3498	*16/03/2015*	1526.411	1517.7075
26/01/2015	1501.0423	1496.8724	*17/03/2015*	1517.9559	1525.1451
27/01/2015	1519.9624	1495.3134	*18/03/2015*	1521.8113	1521.0899
28/01/2015	1506.9836	1514.9308	*19/03/2015*	1507.6492	1519.0721
29/01/2015	1522.0839	1521.4637	*20/03/2015*	1511.8572	1511.8834
30/01/2015	1529.5241	1520.1513	*21/03/2015*	1513.807	1505.9325
31/01/2015	1533.8517	1527.8895	*22/03/2015*	1517.1241	1516.4686
01/02/2015	1540.4325	1528.8742	*23/03/2015*	1525.8408	1519.6051
02/02/2015	1529.2204	1532.5740	*24/03/2015*	1520.2203	1526.5501
03/02/2015	1535.4412	1533.8814	*25/03/2015*	1526.9219	1522.6773
04/02/2015	1523.025	1528.3709	*26/03/2015*	1514.9689	1524.2580
05/02/2015	1531.7279	1524.7965	*27/03/2015*	1522.9067	1516.5931
06/02/2015	1509.3388	1526.4405	*28/03/2015*	1522.1915	1519.5205
07/02/2015	1510.1184	1510.0912	*29/03/2015*	1521.2459	1524.3490
08/02/2015	1502.9263	1501.3854	*30/03/2015*	1520.8601	1520.4064
09/02/2015	1515.7618	1506.6110	*31/03/2015*	1516.1149	1520.0903
10/02/2015	1523.2385	1515.2972	*01/04/2015*	1506.2849	1515.3149
11/02/2015	1523.7132	1527.2951	*02/04/2015*	1509.0339	1504.9521
12/02/2015	1515.7651	1524.9059	*03/04/2015*	1512.6622	1505.3991
13/02/2015	1516.3187	1514.4262	*04/04/2015*	1511.4681	1513.6698

14/02/2015	1515.0659	1512.6768	*05/04/2015*	1520.3437	1514.3328
15/02/2015	1517.1538	1516.3482	*06/04/2015*	1514.91	1522.2368
16/02/2015	1514.992	1517.6861	*07/04/2015*	1500.6096	1521.1823
17/02/2015	1532.9727	1517.1627	*08/04/2015*	1497.9368	1502.4674
18/02/2015	1535.5013	1528.5486	*09/04/2015*	1481.6742	1498.6272
19/02/2015	1527.7996	1528.2116	*10/04/2015*	1468.7926	1495.9006