DAFTAR PUSTAKA

[1] Stalling, William. 2001. *Dasar-Dasar Komunikasi Data*. Jakarta: Penerbit Salemba Teknika

[2] Ariyus, Dony. 2008. *Komunikasi Data*. Yogyakarta: Penerbit Andi

[3] Gunawan, Arif Hamdani. 2004. *Komunikasi Data Via Cable Network*. Jakarta: Penerbit Salemba Teknika

[4] Mauro, Dounglas R dn Kevin J. Schmidt, 2005, *Essential SNMP*, edisi ke-2, O'Reilly, Sebastopol, CA.

[5] Behrouz A. Forouzan, *Data Communication and Networking Fourth Edition*, Mcgraw Hill, 2007

[6] Drs. Jek Siang, Jong, M. Sc. 2009. *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB.* Yogyakarta : Penerbit Andi

[7] Hermawan, Arief. 2006. *Jaringan Syaraf Tiruan Teori dan Aplikasi.* Yogyakarta : Penerbit Andi

[8] Hery Purnomo, Mauridhi ; Kurniawan, Agus. 2006. *Supervised Neural Networks dan Aplikasinya.* Yogyakarta : Graha Ilmu

[9] Away, Gunaidi Abdia. 2010. *Matlab Programming*. Bandung: Penerbit Informatika

[10] Djarwanto PS., S.E. 1990. *Pokok-Pokok Metode Riset dan Bimbingan Teknis Penulisan Skripsi.* Yogyakarta: Liberty Yogyakarta

[11] JJ Siang. 2005. *Jaringan Saraf Tiruan dan Pemogramannya menggunakan MATLAB*. Yogyakarta: Andi Yogyakarta

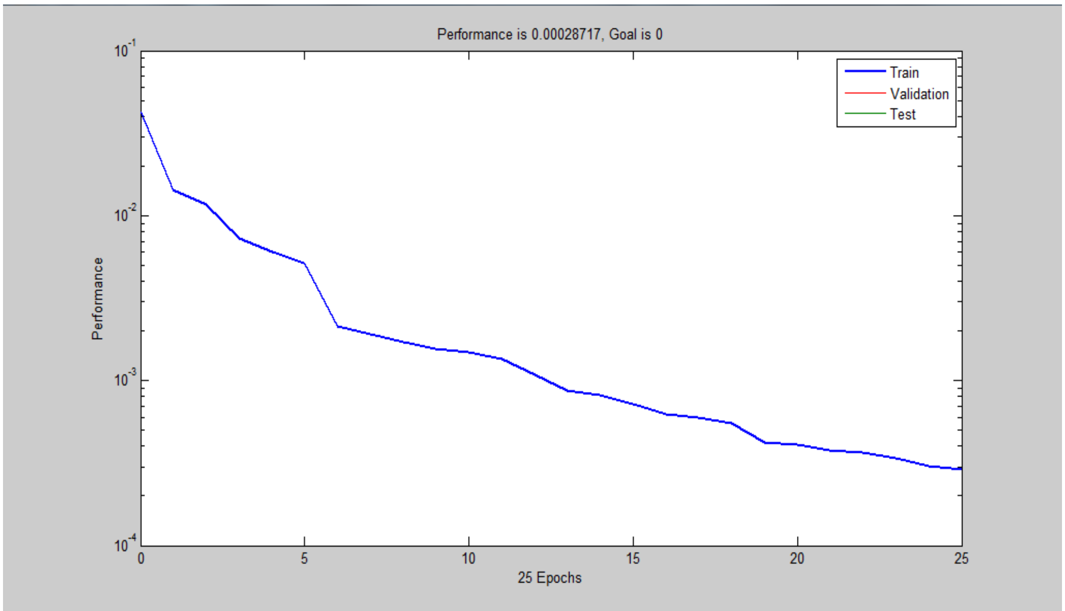
LAMPIRAN

OUTBOUND

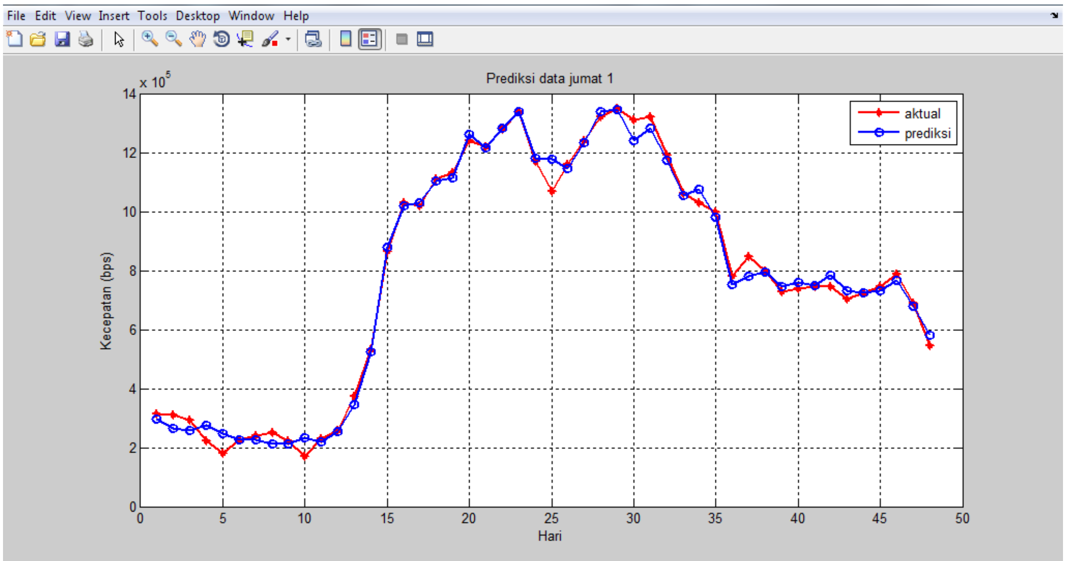
Jum’at

```
clc;
clear;
%load target
   load jumat_target.mat
    j=jumat_target'


%load data 1
    load jumat_data01.mat
    P1=jumat_data01'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load jumat_data02.mat
    P2=jumat_data02';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];


%bentuk jaringan RBF
    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,0.3,25,1)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);

%data uji

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
     yqq=abs(yqq);
     yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF


    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi data jumat
1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
    grid;



%menampilkan RMSE
    rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
    disp(['Root Meant Square Error = ' num2str(rmse)])
```

maka didapatkan nilai epoch



Dan grafiknya :



```
NEWRB, neurons = 0, MSE = 0.0426808

NEWRB, neurons = 2, MSE = 0.0117328

NEWRB, neurons = 3, MSE = 0.00724135

NEWRB, neurons = 4, MSE = 0.00606054

NEWRB, neurons = 5, MSE = 0.00518527

NEWRB, neurons = 6, MSE = 0.00213175

NEWRB, neurons = 7, MSE = 0.00192093

NEWRB, neurons = 8, MSE = 0.00170004

NEWRB, neurons = 9, MSE = 0.00154827

NEWRB, neurons = 10, MSE = 0.00147464

NEWRB, neurons = 11, MSE = 0.0013623

NEWRB, neurons = 12, MSE = 0.00109063

NEWRB, neurons = 13, MSE = 0.000874285

NEWRB, neurons = 14, MSE = 0.000817355

NEWRB, neurons = 15, MSE = 0.000719241

NEWRB, neurons = 16, MSE = 0.000622921

NEWRB, neurons = 17, MSE = 0.000592787
```

```
NEWRB, neurons = 18, MSE = 0.00055254

NEWRB, neurons = 19, MSE = 0.000422932

NEWRB, neurons = 20, MSE = 0.000408825

NEWRB, neurons = 21, MSE = 0.000380524

NEWRB, neurons = 22, MSE = 0.000366776

NEWRB, neurons = 23, MSE = 0.0003383

NEWRB, neurons = 24, MSE = 0.00030169

NEWRB, neurons = 25, MSE = 0.00028717


net =


    Neural Network object:


    architecture:


         numInputs: 1

         numLayers: 2

       biasConnect: [1; 1]

      inputConnect: [1; 0]

      layerConnect: [0 0; 1 0]

     outputConnect: [0 1]


        numOutputs: 1  (read-only)

    numInputDelays: 0  (read-only)

    numLayerDelays: 0  (read-only)


    subobject structures:
            inputs: {1x1 cell} of inputs

            layers: {2x1 cell} of layers

           outputs: {1x2 cell} containing 1 output

            biases: {2x1 cell} containing 2 biases

      inputWeights: {2x1 cell} containing 1 input weight

      layerWeights: {2x2 cell} containing 1 layer weight


    functions:


          adaptFcn: (none)

         divideFcn: (none)

       gradientFcn: (none)

           initFcn: (none)

        performFcn: 'mse'

          plotFcns: {}

          trainFcn: (none)


    parameters:


        adaptParam: (none)

       divideParam: (none)

     gradientParam: (none)

         initParam: (none)

      performParam: (none)

        trainParam: (none)


    weight and bias values:
```

```
                IW: {2x1 cell} containing 1 input weight matrix

                LW: {2x2 cell} containing 1 layer weight matrix

                 b: {2x1 cell} containing 2 bias vectors


    other:


              name: ''

          userdata: (user information)




tr =


    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]

     perf: [1x26 double]




Bobot_input =


    0.6760    0.0656    0.6896    0.5976

    0.0903    0.0904    0.3758    0.5498

    0.9140    0.1442    0.3724    0.8846

    0.6972    0.1406    0.8455    0.6242

    0.2873    0.1295    0.7536    0.7783

    0.8521    0.1084    0.6115    0.4186

    0.2738    0.0909    0.3752    0.3830

    0.5937    0.0638    0.5201    0.7623

    0.9067    0.1313    0.8812    0.6136
    0.5002    0.0549    0.8776    0.8633

    0.1635    0.1318    0.6432    0.4489

    0.2733    0.0279    0.5720    0.4234

    0.1158    0.0326    0.1808    0.0833

    0.8539    0.1199    0.5421    0.7198

    1.0000    0.0888    0.6064    0.6348

    0.3003    0.0567    0.4732    0.3134

    0.1020    0.0271    0.1435    0.1285

    0.7891    0.1202    0.6654    0.3926

    0.2655    0.0642    0.2967    0.3426

    0.0881    0.0252    0.1482    0.0865

    0.4800    0.1262    0.8109    0.7464

    0.3416    0.0921    0.5249    0.3224

    0.2243    0.0400    0.2255    0.2395

    0.9691    0.1058    0.4391    0.5764

    0.1073    0.1515    0.5213    0.5551


Bobot_bias_input =


    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752
    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752



Bobot_Lapisan =


  Columns 1 through 12


    0.1712    0.0150    0.7158    0.6485    0.2728    2.0169   -
5.1934    0.4468   -0.1120    0.2716    0.4407   -0.6228


  Columns 13 through 24
```

```
    1.9744   -0.9889    0.5585    0.1221    2.4790   -1.7558
6.6520   -3.4063   -0.6105    1.2502   -3.7783   -0.4175


  Column 25


    0.5237



Bobot_Bias_lapisan =


    0.1697



Y1 =


  Columns 1 through 12


    0.0651    0.0484    0.0435    0.0534    0.0377    0.0275
0.0278    0.0204    0.0203    0.0302    0.0233    0.0425


  Columns 13 through 24


    0.0899    0.1863    0.3734    0.4488    0.4547    0.4933
0.4981    0.5776    0.5526    0.5879    0.6185    0.5345


  Columns 25 through 36


    0.5315    0.5153    0.5626    0.6176    0.6225    0.5658
0.5875    0.5305    0.4681    0.4789    0.4285    0.3076
```

```
  Columns 37 through 48

    0.3226    0.3294    0.3028    0.3100    0.3054    0.3230
0.2964    0.2921    0.2951    0.3143    0.2681    0.2150



Y2 =

  Columns 1 through 12

    0.0837    0.1254    0.0419    0.0323    0.0207    0.0200
0.0174    0.0168    0.0142    0.0199    0.0308    0.0455

  Columns 13 through 24

    0.0755    0.2285    0.3829    0.3757    0.3205    0.2793
0.3122    0.2354    0.2107    0.1809    0.1979    0.1927

  Columns 25 through 36

    0.2217    0.2591    0.2142    0.2583    0.2207    0.2813
0.3287    0.2257    0.2693    0.3259    0.2897    0.2703

  Columns 37 through 48

    0.2538    0.2648    0.2510    0.2627    0.2704    0.2699
0.2750    0.2913    0.2823    0.2882    0.2849    0.2544



C =
  232.0023

  233.0018
```

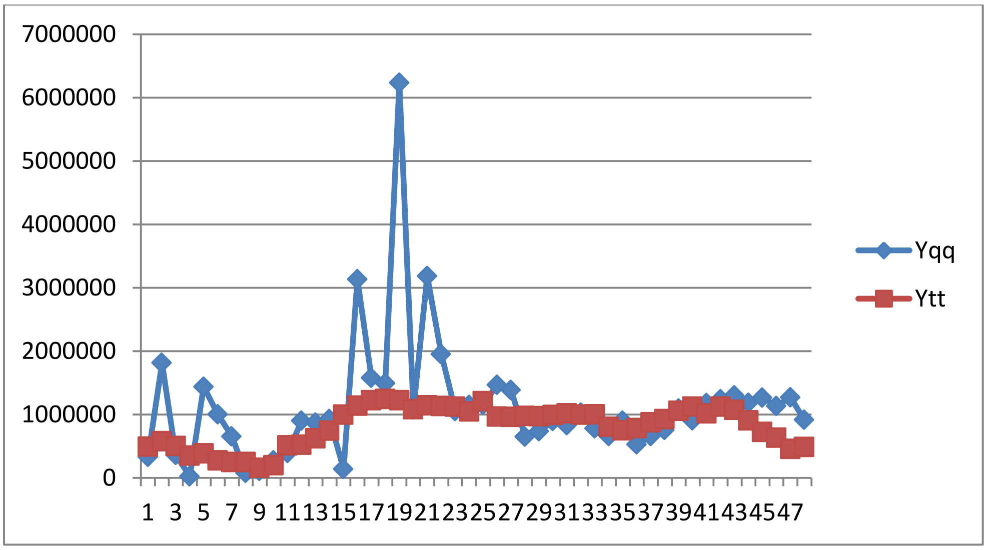
Root Meant Square Error = 0.027053

Berikut nilai Yqq dan Ytt pada prediksi hari jum’at

prediksi jumat	
Yqq	Ytt
287830	297732
368080	266264
207313	256943
188976	275607
166554	246051
165264	226913
160202	227480
158947	213627
154071	213293
165006	231978
186010	218918
214245	255218
272089	344385
566640	525775
863904	877859
850110	1019881
743678	1030994
664420	1103605
727707	1112615
579907	1262267
532320	1215270
474930	1281698
507720	1339298
497722	1181216
553487	1175542
625637	1144970
539097	1234018
624069	1337598

551562	1346700
668387	1240042
759496	1280953
561328	1173533
645285	1056096
754231	1076429
684474	981605
647064	754148
615340	782382
636437	795138
609881	745119
632501	758677
647380	750012
646305	783145
656073	732981
687477	724943
670189	730607
681527	766776
675199	679832
616425	579865

Grafiknyan :

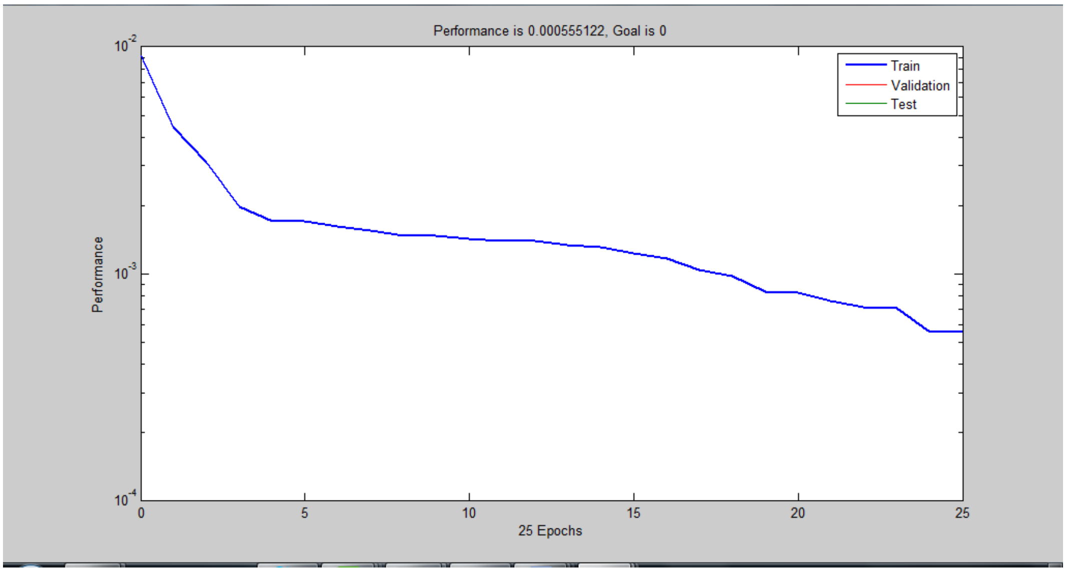


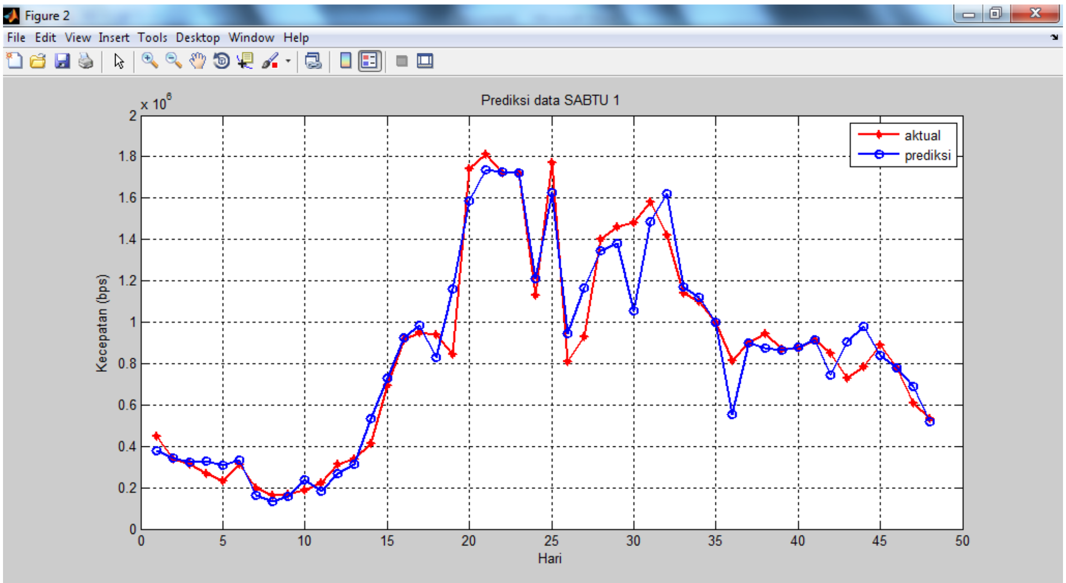
SABTU

```
clc;
clear;
%load target
   load sabtu_target.mat
    j=sabtu_target'


%load data 1
    load sabtu_data01.mat
    P1=sabtu_data01'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load sabtu_data02.mat
    P2=sabtu_data02';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];


%bentuk jaringan RBF
    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,0.3,25,1)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);
```

```
%data uji

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
     yqq=abs(yqq);
     yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF


    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi data SABTU
1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
    grid;



%menampilkan RMSE
    rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
    disp(['Root Meant Square Error = ' num2str(rmse)])
```

maka didapatkan grafik :





```
NEWRB, neurons = 0, MSE = 0.00912809
NEWRB, neurons = 2, MSE = 0.00308902
NEWRB, neurons = 3, MSE = 0.0019818
NEWRB, neurons = 4, MSE = 0.00171262
NEWRB, neurons = 5, MSE = 0.00171145
NEWRB, neurons = 6, MSE = 0.00162157
NEWRB, neurons = 7, MSE = 0.00154923
NEWRB, neurons = 8, MSE = 0.00147733
NEWRB, neurons = 9, MSE = 0.00147273
NEWRB, neurons = 10, MSE = 0.00142721
NEWRB, neurons = 11, MSE = 0.00139974
NEWRB, neurons = 12, MSE = 0.00139972
NEWRB, neurons = 13, MSE = 0.00134286
NEWRB, neurons = 14, MSE = 0.00131029
NEWRB, neurons = 15, MSE = 0.00123599
NEWRB, neurons = 16, MSE = 0.0011644
NEWRB, neurons = 17, MSE = 0.00104471
NEWRB, neurons = 18, MSE = 0.000983945
NEWRB, neurons = 19, MSE = 0.000838721
NEWRB, neurons = 20, MSE = 0.000824266
NEWRB, neurons = 21, MSE = 0.000759695
NEWRB, neurons = 22, MSE = 0.000713984
NEWRB, neurons = 23, MSE = 0.000704833
NEWRB, neurons = 24, MSE = 0.000555155
NEWRB, neurons = 25, MSE = 0.000555122

net =

    Neural Network object:

    architecture:

         numInputs: 1
         numLayers: 2
       biasConnect: [1; 1]
      inputConnect: [1; 0]
      layerConnect: [0 0; 1 0]
     outputConnect: [0 1]

        numOutputs: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)

    subobject structures:

            inputs: {1x1 cell} of inputs
            layers: {2x1 cell} of layers
           outputs: {1x2 cell} containing 1 output
            biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1 input weight
      layerWeights: {2x2 cell} containing 1 layer weight

    functions:

          adaptFcn: (none)
         divideFcn: (none)
       gradientFcn: (none)
           initFcn: (none)
        performFcn: 'mse'
          plotFcns: {}
          trainFcn: (none)

    parameters:

        adaptParam: (none)
       divideParam: (none)
     gradientParam: (none)
         initParam: (none)
      performParam: (none)
        trainParam: (none)

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight matrix
                LW: {2x2 cell} containing 1 layer weight matrix
                 b: {2x1 cell} containing 2 bias vectors

    other:

              name: ''
          userdata: (user information)


tr =

    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]
     perf: [1x26 double]


Bobot_input =
    0.1036    0.0340    0.4096    0.3302
    0.0857    0.0425    0.9052    0.3064
    0.0824    0.0139    0.4862    0.2014
    0.1173    0.0471    0.3664    0.2846
    0.0566    0.0143    0.3035    0.1835
    0.0901    0.0059    1.0000    0.1500
    0.0814    0.0330    0.1723    0.2529
    0.1451    0.0158    0.2239    0.2826
    0.0432         0    0.0190    0.0088
    0.1185    0.0308    0.2891    0.2747
    0.1046    0.0284    0.3843    0.3084
    0.1130    0.0217    0.2212    0.2787
    0.1011    0.0109    0.2086    0.1439
    0.0892    0.0112    0.1300    0.0924
    0.0920    0.0213    0.1795    0.2331
    0.0969    0.0171    0.1835    0.1613
    0.0740    0.0101    0.4230    0.2192
    0.0919    0.0293    0.3664    0.2985
    0.0779    0.0140    0.0802    0.0518
    0.1168    0.0172    0.1889    0.1689
    0.0817    0.0210    0.1785    0.1756
    0.0612    0.0115    0.0440    0.0345
    0.0810    0.0179    0.3664    0.2707
    0.0794    0.0234    0.3664    0.2846
    0.0693    0.0033    0.0964    0.0787


Bobot_bias_input =

    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
```

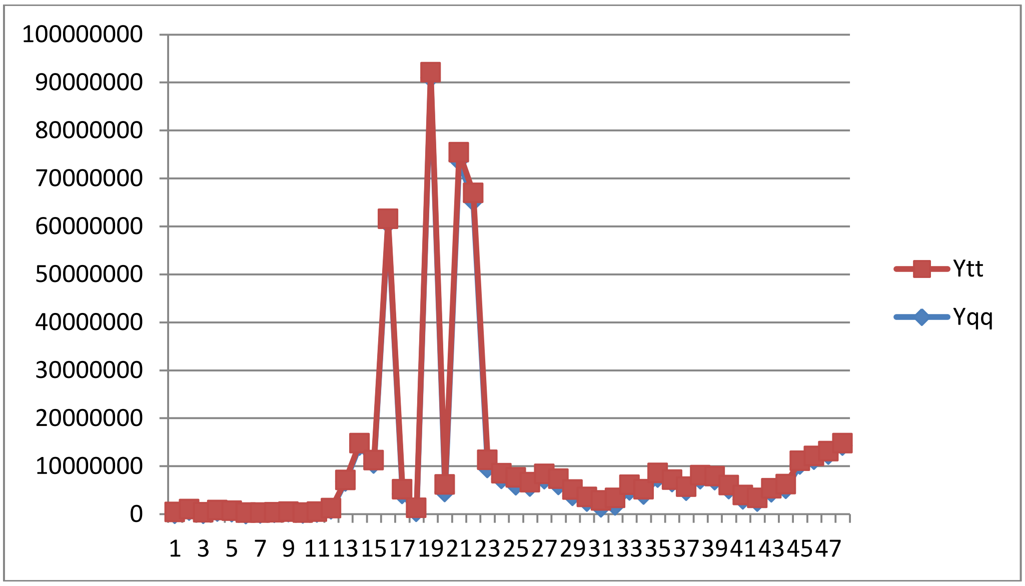
```
Bobot_Lapisan =

  1.0e+003 *

  Columns 1 through 12

   -2.5490    0.0158   -0.1883   -0.2071    0.2442    0.0063   -
0.4993    0.0509   -0.0155   -0.1179    6.3309   -0.6038

  Columns 13 through 24

    1.2240   -1.2031    1.5377   -0.8802    0.8075   -5.3521
0.7132    0.1086   -0.1090   -0.0947   -9.1546    9.9737

  Column 25

    0.0238


Bobot_Bias_lapisan =

  -16.1353


Y1 =

  Columns 1 through 12

    0.0483    0.0419    0.0377    0.0386    0.0347    0.0393
0.0053   -0.0006    0.0044    0.0204    0.0096    0.0264

  Columns 13 through 24

    0.0356    0.0794    0.1179    0.1564    0.1686    0.1381
0.2030    0.2880    0.3176    0.3155    0.3142    0.2132

  Columns 25 through 36

    0.2960    0.1611    0.2041    0.2406    0.2473    0.1830
0.2682    0.2948    0.2055    0.1952    0.1719    0.0834

  Columns 37 through 48

    0.1516    0.1468    0.1451    0.1474    0.1548    0.1208
0.1523    0.1677    0.1403    0.1278    0.1101    0.0763


Y2 =

  Columns 1 through 12

   -0.0111   -0.1387   -0.0098    0.0608   -0.0837   -0.0224
0.0006    0.0217    0.0364   -0.0360    0.0327    0.1343

  Columns 13 through 24
```

```
    1.0864    2.2991    1.6952   -9.9015   -0.7079    0.0574  -
14.8461    0.7312  -12.0327  -10.6554    1.5452    1.1687

  Columns 25 through 36

    0.9564    0.9145    1.1578    0.9647    0.5910    0.3957
0.2067   -0.3123   -0.8276   -0.6821   -1.2623   -1.1029

  Columns 37 through 48

   -0.8117   -1.2056   -1.1713   -0.8696   -0.5230   -0.4467   -
0.7540   -0.8821   -1.7010   -1.8676   -2.0451   -2.3446


C =

  457.0023
  458.0018

Root Meant Square Error = 0.072145

Nilai Ytt dan Yqq
```

prediksi Sabtu	
Yqq	Ytt
60059	377430
722931	345084
68091	323953
501322	328464
385186	308972
9379	332341
131928	160685
261114	130797
351135	155799
92768	236573
329020	182274
952048	266972
6793454	313364
14234534	534362
10528845	728831
60623002	923104
4215112	984778
480069	830954
90960642	1157993
4614611	1587071
73698900	1736609

65248347	1725784
9608648	1719523
7298982	1209891
5996051	1627526
5739132	946599
7231906	1163964
6047201	1347899
3754050	1381891
2556231	1057233
1396026	1487079
1787723	1621390
4949760	1170786
4056669	1118934
7616847	1001387
6638857	554930
4852080	898983
7268587	874659
7058572	865982
5207347	877704
3080602	915195
2612470	743221
4498181	902502
5283983	980180
10308188	841940
11330556	778931
12419513	689656
14257130	518820

Grafiknya :

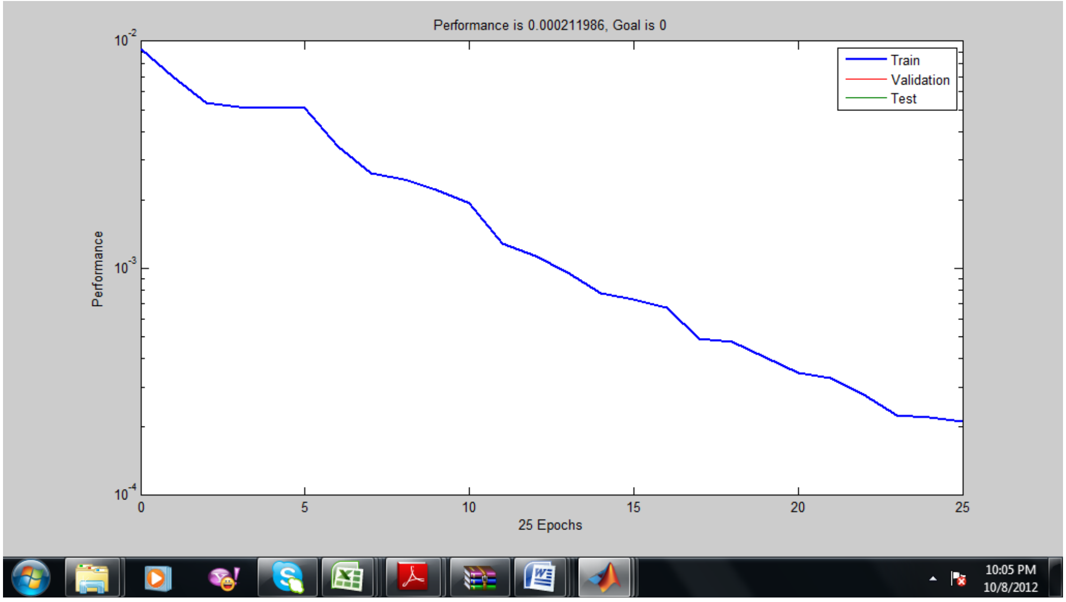


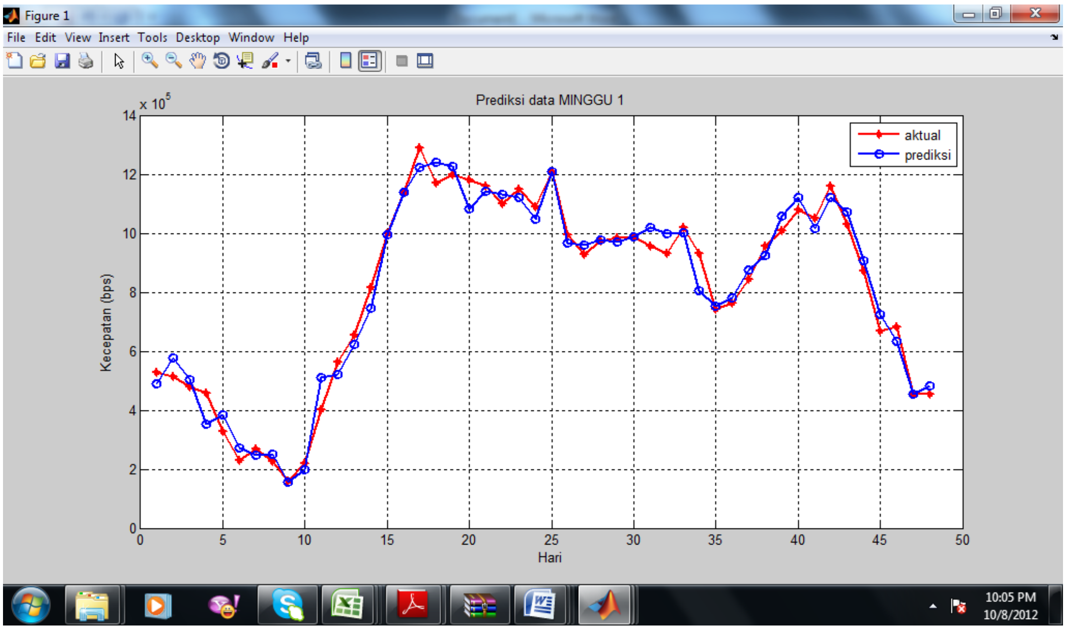
MINGGU

```
clc;
clear;
%load target
   load minggu_target.mat
    j=minggu_target'


%load data 1
    load minggu_data01.mat
    P1=minggu_data01'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load minggu_data02.mat
    P2=minggu_data02';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];
```

```
%bentuk jaringan RBF
    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,0.3,25,1)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);


%data uji

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
     yqq=abs(yqq);
     yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF


    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi data MINGGU
1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
    grid;




%menampilkan RMSE
    rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
    disp(['Root Meant Square Error = ' num2str(rmse)])
```

maka didapatkan grafik





```
NEWRB, neurons = 0, MSE = 0.00921849
NEWRB, neurons = 2, MSE = 0.00535492
NEWRB, neurons = 3, MSE = 0.00511997
NEWRB, neurons = 4, MSE = 0.00507302
NEWRB, neurons = 5, MSE = 0.00505907
NEWRB, neurons = 6, MSE = 0.00346165
NEWRB, neurons = 7, MSE = 0.00261249
NEWRB, neurons = 8, MSE = 0.0024549
NEWRB, neurons = 9, MSE = 0.00221658
NEWRB, neurons = 10, MSE = 0.00193319
NEWRB, neurons = 11, MSE = 0.00128017
NEWRB, neurons = 12, MSE = 0.00113718
NEWRB, neurons = 13, MSE = 0.000957039
```

```
NEWRB, neurons = 14, MSE = 0.000774649
NEWRB, neurons = 15, MSE = 0.000733154
NEWRB, neurons = 16, MSE = 0.000673898
NEWRB, neurons = 17, MSE = 0.000491465
NEWRB, neurons = 18, MSE = 0.000476396
NEWRB, neurons = 19, MSE = 0.000406645
NEWRB, neurons = 20, MSE = 0.000346571
NEWRB, neurons = 21, MSE = 0.000329835
NEWRB, neurons = 22, MSE = 0.000276922
NEWRB, neurons = 23, MSE = 0.000226174
NEWRB, neurons = 24, MSE = 0.000221451
NEWRB, neurons = 25, MSE = 0.000211986

net =

    Neural Network object:

    architecture:

         numInputs: 1
         numLayers: 2
       biasConnect: [1; 1]
      inputConnect: [1; 0]
      layerConnect: [0 0; 1 0]
     outputConnect: [0 1]

        numOutputs: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)

    subobject structures:

            inputs: {1x1 cell} of inputs
            layers: {2x1 cell} of layers
           outputs: {1x2 cell} containing 1 output
            biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1 input weight
      layerWeights: {2x2 cell} containing 1 layer weight

    functions:

          adaptFcn: (none)
         divideFcn: (none)
       gradientFcn: (none)
           initFcn: (none)
        performFcn: 'mse'
          plotFcns: {}
          trainFcn: (none)

    parameters:

        adaptParam: (none)
       divideParam: (none)
     gradientParam: (none)
         initParam: (none)
      performParam: (none)
```

```
        trainParam: (none)

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight matrix
                LW: {2x2 cell} containing 1 layer weight matrix
                 b: {2x1 cell} containing 2 bias vectors

    other:

              name: ''
          userdata: (user information)


tr =

    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]
     perf: [1x26 double]


Bobot_input =

    0.1406    0.0330    0.1568    0.3038
    0.1596    0.0438    0.1082    1.0000
    0.1724    0.0336    0.1056    0.3098
    0.1409    0.0227    0.7362    0.1338
    0.1079    0.0217    0.1264    0.0028
    0.1086    0.0146    0.0227    0.0612
    0.2213    0.0245    0.1489    0.2767
    0.1237    0.0278    0.1120    0.3309
    0.0980    0.0231    0.1134    0.0782
    0.1599    0.0339    0.1657    0.3219
    0.1235    0.0345    0.1760    0.2414
    0.1203    0.0224    0.1601    0.0932
    0.1371    0.0249    0.1346    0.2044
    0.0540    0.0190    0.0461    0.0145
    0.1086    0.0043    0.0288    0.0275
    0.1479    0.0265    0.1204    0.1980
    0.1892    0.0245    0.1628    0.2095
    0.1309    0.0653    0.1522    0.2172
    0.0871    0.0090    0.0413    0.0194
    0.1110    0.0130    0.0127    0.0222
    0.2018    0.0237    0.1007    0.2767
    0.1403    0.0451    0.0994    0.2314
    0.0781    0.0218    0.0930    0.0055
    0.1213    0.0362    0.1306    0.3068
    0.0784    0.0060    0.0500    0.0139


Bobot_bias_input =

    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752
    2.7752


Bobot_Lapisan =

  1.0e+003 *

  Columns 1 through 12

   -1.4197    0.0007    0.2879    0.0004   -0.0215    0.2195
0.5559    0.1616   -0.6017    0.3869    0.9672    0.1985

  Columns 13 through 24

   -2.3988   -0.3897   -1.1132    3.4205   -0.7561   -0.0587
0.4492    0.2705   -0.6278   -0.6699    0.1342    0.3885

  Column 25

    0.6223


Bobot_Bias_lapisan =

   -1.4061


Y1 =

  Columns 1 through 12

    0.1079    0.1345    0.1120    0.0663    0.0764    0.0422
0.0346    0.0356    0.0074    0.0203    0.1145    0.1172

  Columns 13 through 24
```

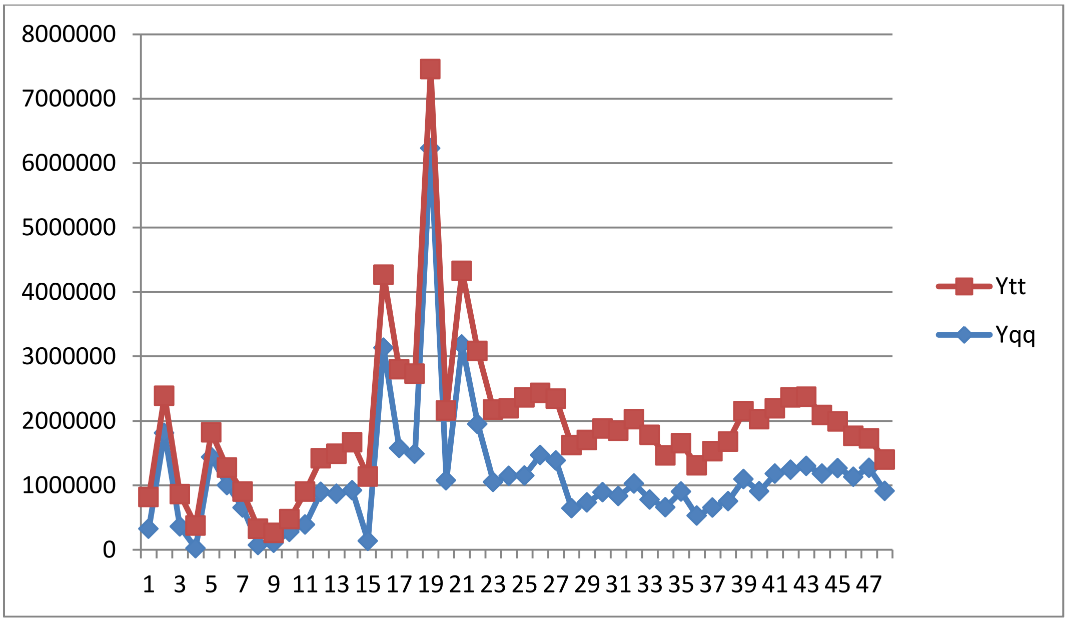
```
    0.1480    0.1849    0.2604    0.3030    0.3287    0.3342
0.3294    0.2860    0.3047    0.3011    0.2983    0.2762

  Columns 25 through 36

    0.3249    0.2512    0.2494    0.2543    0.2524    0.2583
0.2672    0.2609    0.2626    0.2029    0.1866    0.1958

  Columns 37 through 48

    0.2241    0.2390    0.2793    0.2983    0.2668    0.2983
0.2837    0.2334    0.1789    0.1514    0.0977    0.1061


Y2 =

  Columns 1 through 12

    0.0509   -0.3568    0.0570   -0.0160   -0.2853   -0.2026   -
0.1363    0.0027    0.0085   -0.0642   -0.0860   -0.1827

  Columns 13 through 24

   -0.1768   -0.1874    0.0149   -0.6081    0.2887    0.2725   -
1.1987    0.1937   -0.6176   -0.3833    0.1888    0.2071

  Columns 25 through 36

    0.2078    0.2678    0.2524    0.1114    0.1278    0.1586
0.1466    0.1838    0.1367    0.1137    0.1602    0.0889

  Columns 37 through 48

    0.1133    0.1322    0.1968    0.1609    0.2129    0.2246
0.2358    0.2137    0.2295    0.2038   -0.2535   -0.1856


C =

  232.0029
  233.0020

Root Meant Square Error = 0.04279

Nilai ytt dan Yqq
```

prediksi Minggu	
Yqq	Ytt
328110	490102
1812552	578325
359709	503434

23618	351746
1437370	385262
1003302	271851
654921	246606
74582	250150
105371	156395
276401	199394
391167	511963
898620	520731
867459	622961
923176	745361
139016	995788
3132120	1137397
1576269	1222417
1491275	1240966
6232921	1224793
1077803	1080779
3181978	1142999
1952070	1130944
1052052	1121812
1147817	1048351
1151573	1209986
1466892	965586
1385870	959578
645768	975711
731523	969415
893254	988929
830552	1018431
1025724	997589
778563	1003173
657560	805058
901518	751005
527424	781710
655526	875461
754602	924936
1093850	1058645
905276	1121693
1178657	1017339
1239937	1121706
1298472	1073373
1182836	906258

1265643	725554
1130803	634394
1270253	455982
913753	483964

Maka didapatkan grafik



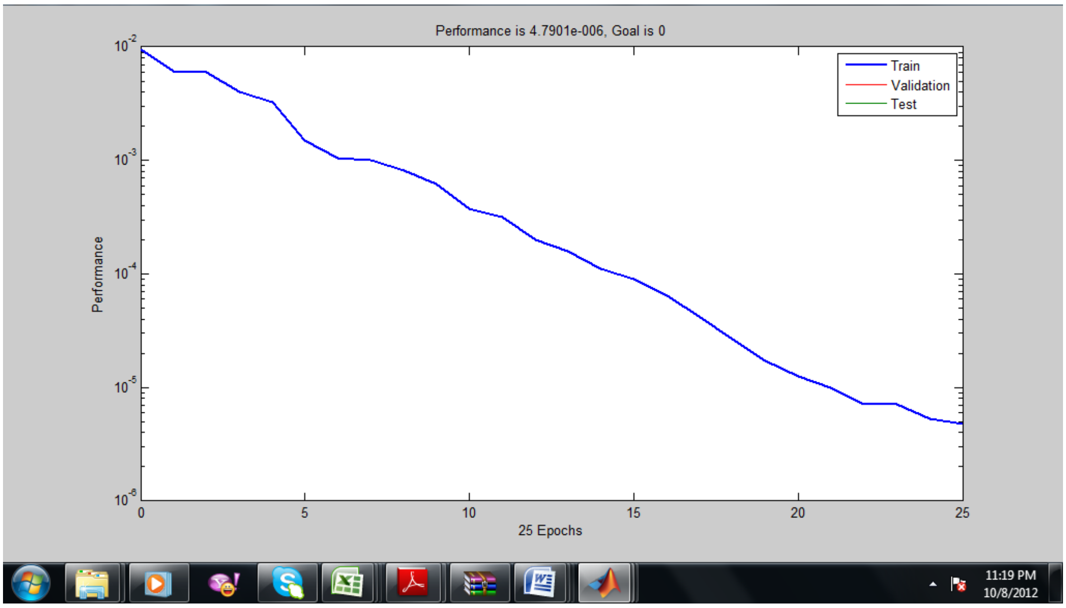
SENIN

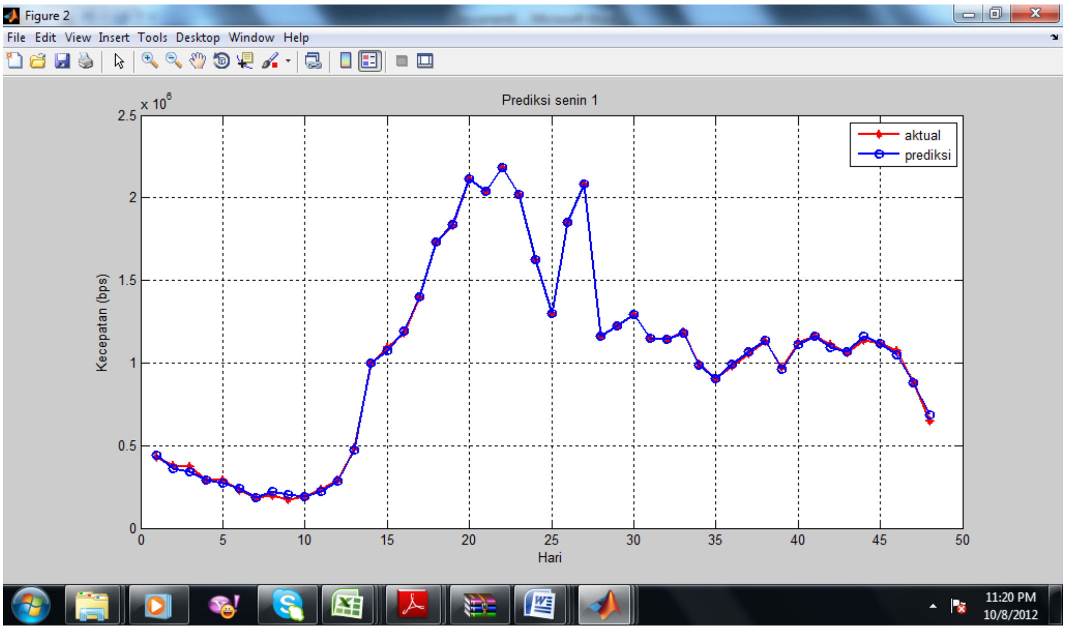
Listingnya :

```
clc;
clear;
%load target
   load senin_target2.mat
    j=senin_target2'


%load data 1
    load senin_out1.mat
    P1=senin_out1'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load senin_out2.mat
    P2=senin_out2';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];



%bentuk jaringan RBF
%     spread=0.1;

    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,0.1,25,1)

    % [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}
%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
     yqq=abs(yqq);
     yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF

    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi senin 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
    grid;




%menampilkan RMSE
```

```
rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
disp(['Root Meant Square Error = ' num2str(rmse)])
```

maka didapatkan grafiknya :





```
NEWRB, neurons = 0, MSE = 0.00955207

NEWRB, neurons = 2, MSE = 0.00596048

NEWRB, neurons = 3, MSE = 0.00402613
```

```
NEWRB, neurons = 4, MSE = 0.00323521
NEWRB, neurons = 5, MSE = 0.00149223
NEWRB, neurons = 6, MSE = 0.0010592
NEWRB, neurons = 7, MSE = 0.00101258
NEWRB, neurons = 8, MSE = 0.000812728
NEWRB, neurons = 9, MSE = 0.000624934
NEWRB, neurons = 10, MSE = 0.00037513
NEWRB, neurons = 11, MSE = 0.000316039
NEWRB, neurons = 12, MSE = 0.0002017
NEWRB, neurons = 13, MSE = 0.000157989
NEWRB, neurons = 14, MSE = 0.000111634
NEWRB, neurons = 15, MSE = 9.07916e-005
NEWRB, neurons = 16, MSE = 6.46464e-005
NEWRB, neurons = 17, MSE = 4.19809e-005
NEWRB, neurons = 18, MSE = 2.66117e-005
NEWRB, neurons = 19, MSE = 1.73374e-005
NEWRB, neurons = 20, MSE = 1.25126e-005
NEWRB, neurons = 21, MSE = 9.91835e-006
NEWRB, neurons = 22, MSE = 7.09467e-006
NEWRB, neurons = 23, MSE = 7.09244e-006
NEWRB, neurons = 24, MSE = 5.25804e-006
NEWRB, neurons = 25, MSE = 4.7901e-006

net =


    Neural Network object:


    architecture:
         numInputs: 1
         numLayers: 2
       biasConnect: [1; 1]
      inputConnect: [1; 0]
      layerConnect: [0 0; 1 0]
     outputConnect: [0 1]

        numOutputs: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)

    subobject structures:

            inputs: {1x1 cell} of inputs
            layers: {2x1 cell} of layers
           outputs: {1x2 cell} containing 1 output
            biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1 input weight
      layerWeights: {2x2 cell} containing 1 layer weight

    functions:

          adaptFcn: (none)
         divideFcn: (none)
       gradientFcn: (none)
           initFcn: (none)
        performFcn: 'mse'
```

```
          plotFcns: {}
          trainFcn: (none)

    parameters:

        adaptParam: (none)
       divideParam: (none)
     gradientParam: (none)
         initParam: (none)
      performParam: (none)
        trainParam: (none)

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight matrix
                LW: {2x2 cell} containing 1 layer weight matrix
                 b: {2x1 cell} containing 2 bias vectors

    other:

              name: ''
          userdata: (user information)


tr =

    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]
```

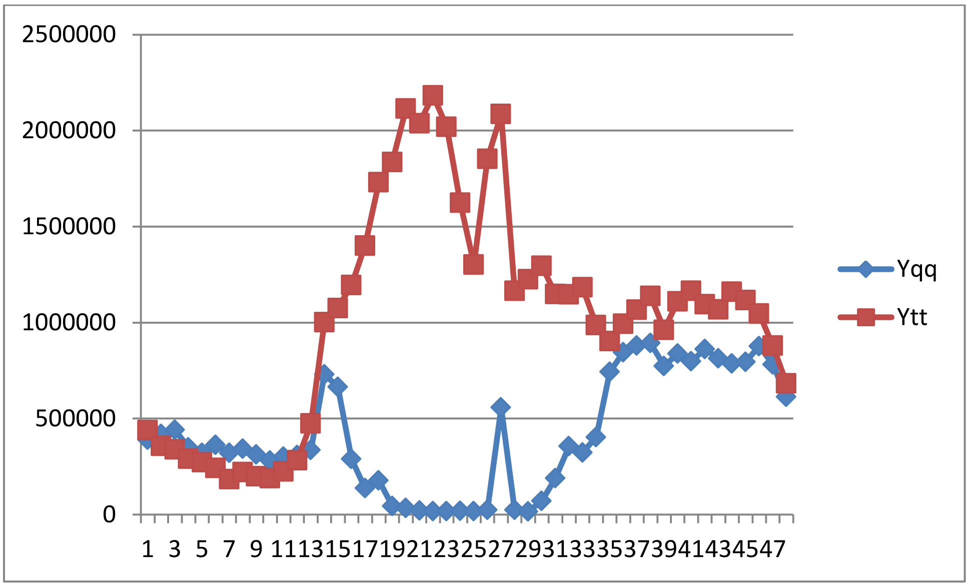
```
     perf: [1x26 double]


Bobot_input =

    0.0165    0.1842    0.2870    0.2701
    0.1086    0.0981    0.1294    0.1736
    0.3391    0.3061    0.2755         0
    0.3136    0.2582    0.1790    0.3019
    0.0316    0.3138    0.3065    0.3231
    0.3511    0.2960    0.1695    1.0000
    0.0151    0.0533    0.1520    0.0950
    0.2058    0.1956    0.1550    0.2836
    0.3061    0.3027    0.1888    0.1956
    0.0165    0.1139    0.2061    0.1610
    0.3793    0.3185    0.1548         0
    0.3055    0.3288    0.3050         0
    0.2423    0.2119    0.1601    0.3633
    0.1475    0.0945    0.1369    0.1221
    0.0165    0.0957    0.1713    0.2364
    0.1535    0.1186    0.1349    0.2370
    0.0321    0.2280    0.3236    0.2720
    0.0192    0.1454    0.2642    0.2417
    0.0320    0.3104    0.2423    0.3183
    0.1837    0.1839    0.1650    0.2566
    0.0774    0.0767    0.0765    0.0708
    0.0151    0.2633    0.3102    0.3154
    0.0098    0.0452    0.0183    0.0404
    0.0318    0.0577    0.0247    0.0418
    0.0284    0.2825    0.3314    0.2967


Bobot_bias_input =

    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255
    8.3255

    8.3255

    8.3255


Bobot_Lapisan =


  Columns 1 through 12


   -0.0893   -0.1837   -0.0600    0.1054    0.1979    0.2199
0.1195   -0.1870    0.1661    0.0663    0.1661    0.3891


  Columns 13 through 24


    0.1744    0.2204    0.0827    0.0927    0.2462    0.1796
0.0835    0.2092   -0.0246   -0.1766   -0.1256    0.1552


  Column 25


    0.1456


Bobot_Bias_lapisan =


   -0.0109


Y1 =
  Columns 1 through 12

    0.0751    0.0611    0.0581    0.0498    0.0468    0.0415
0.0315    0.0376    0.0342    0.0325    0.0384    0.0483

  Columns 13 through 24

    0.0810    0.1711    0.1837    0.2041    0.2391    0.2954
0.3135    0.3610    0.3479    0.3727    0.3447    0.2772

  Columns 25 through 36

    0.2224    0.3161    0.3559    0.1988    0.2091    0.2212
0.1960    0.1957    0.2020    0.1685    0.1542    0.1696

  Columns 37 through 48

    0.1822    0.1944    0.1643    0.1896    0.1989    0.1870
0.1823    0.1980    0.1907    0.1786    0.1503    0.1167


Y2 =

  Columns 1 through 12

    0.0540    0.0591    0.0628    0.0471    0.0420    0.0493
0.0423    0.0459    0.0407    0.0351    0.0387    0.0401

  Columns 13 through 24
```

```
    0.0447    0.1129    0.1016    0.0366    0.0103    0.0173   -
0.0057   -0.0074   -0.0098   -0.0104   -0.0103   -0.0102


  Columns 25 through 36


   -0.0104   -0.0093    0.0831   -0.0093   -0.0108   -0.0011
0.0192    0.0481    0.0424    0.0563    0.1151    0.1326


  Columns 37 through 48


    0.1388    0.1411    0.1204    0.1315    0.1246    0.1356
0.1273    0.1225    0.1241    0.1383    0.1219    0.0927




C =


  457.0090

  458.0032


Root Meant Square Error = 0.0063797

Nilai Yqq dan Ytt
```

Prediksi Senin	
Yqq	Ytt
390106	439939
419427	358280
441068	340151
350288	291895
320891	273939
363127	243055
322276	184296
343231	220596
313278	200503

280947	190420
301877	224729
309781	282810
336366	474834
730556	1002458
665473	1076250
289604	1195792
137572	1401031
178202	1730652
44973	1836893
35081	2115006
21228	2038336
17597	2183594
18156	2019642
19040	1624426
17944	1302988
24267	1852101
558279	2085524
24191	1164947
15674	1224904
71779	1296065
189121	1148421
355909	1146475
323232	1183348
403540	987028
743394	903267
844515	993579
880341	1067476
893251	1139285
774023	962639
837832	1110927
797951	1165225
861988	1095647
813638	1068073
786188	1159904
795326	1117123
877071	1046425
782521	880754
613793	683475

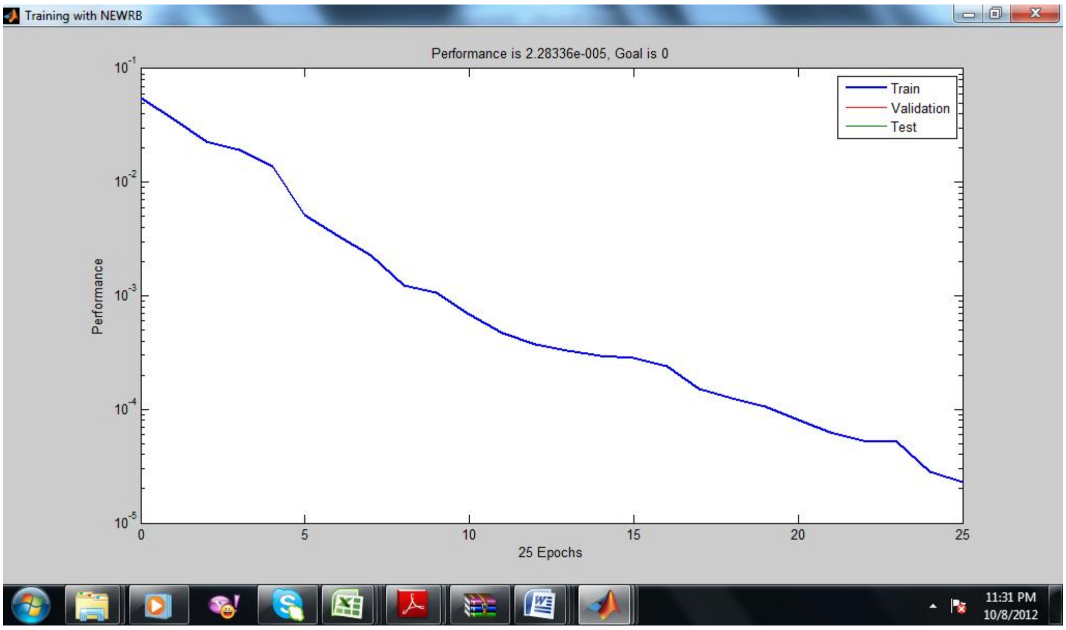
Maka didapakan grafik (senin) :

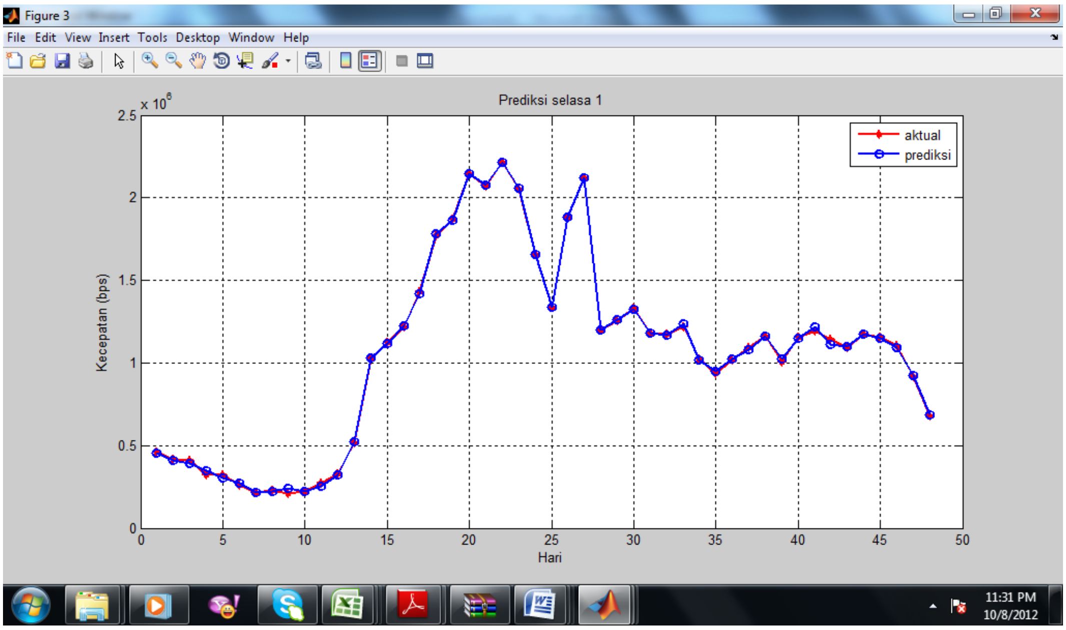


SELASA

```
LISTINGnya :

clc;
clear;
%load target
   load selasa_targetout.mat
    j=selasa_targetout'


%load data 1
    load selasa_out1.mat
    P1=selasa_out1'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load selasa_out2.mat
    P2=selasa_out2';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];


%bentuk jaringan RBF
%     spread=0.1;

    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,0.3,25,1)

    % [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}
%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
     yqq=abs(yqq);
     yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF

    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi selasa 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
    grid;




%menampilkan RMSE
    rmse =  (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
    disp(['Root Meant Square Error = ' num2str(rmse)])
```

didapakan grafik





```
NEWRB, neurons = 0, MSE = 0.054807

NEWRB, neurons = 2, MSE = 0.0227326

NEWRB, neurons = 3, MSE = 0.0193193

NEWRB, neurons = 4, MSE = 0.0137552

NEWRB, neurons = 5, MSE = 0.00518992
```

```
NEWRB, neurons = 6, MSE = 0.0033464
NEWRB, neurons = 7, MSE = 0.00225552
NEWRB, neurons = 8, MSE = 0.00122945
NEWRB, neurons = 9, MSE = 0.00106632
NEWRB, neurons = 10, MSE = 0.000691742
NEWRB, neurons = 11, MSE = 0.000472772
NEWRB, neurons = 12, MSE = 0.000376281
NEWRB, neurons = 13, MSE = 0.000326892
NEWRB, neurons = 14, MSE = 0.000297034
NEWRB, neurons = 15, MSE = 0.000284718
NEWRB, neurons = 16, MSE = 0.000237764
NEWRB, neurons = 17, MSE = 0.000149772
NEWRB, neurons = 18, MSE = 0.000126492
NEWRB, neurons = 19, MSE = 0.00010647
NEWRB, neurons = 20, MSE = 8.06961e-005
NEWRB, neurons = 21, MSE = 6.30512e-005
NEWRB, neurons = 22, MSE = 5.2969e-005
NEWRB, neurons = 23, MSE = 5.18919e-005
NEWRB, neurons = 24, MSE = 2.85214e-005
NEWRB, neurons = 25, MSE = 2.28336e-005

net =

    Neural Network object:

    architecture:
         numInputs: 1
         numLayers: 2
       biasConnect: [1; 1]
      inputConnect: [1; 0]
      layerConnect: [0 0; 1 0]
     outputConnect: [0 1]

        numOutputs: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)

    subobject structures:

            inputs: {1x1 cell} of inputs
            layers: {2x1 cell} of layers
           outputs: {1x2 cell} containing 1 output
            biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1 input weight
      layerWeights: {2x2 cell} containing 1 layer weight

    functions:

          adaptFcn: (none)
         divideFcn: (none)
       gradientFcn: (none)
```

```
           initFcn: (none)
        performFcn: 'mse'
          plotFcns: {}
          trainFcn: (none)

    parameters:

        adaptParam: (none)
       divideParam: (none)
     gradientParam: (none)
         initParam: (none)
      performParam: (none)
        trainParam: (none)

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight
matrix
                LW: {2x2 cell} containing 1 layer weight
matrix
                 b: {2x1 cell} containing 2 bias vectors

    other:

              name: ''
          userdata: (user information)
```

```
tr =


    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25]

     perf: [1x26 double]



Bobot_input =


    0.0161    0.4214    0.6676    0.7273

    0.7888    0.7135    0.6399    0.7442

    0.2221    0.2695    0.3516    0.3182

    0.0522    0.7319    0.7143    0.9621

    0.0446    0.6569    0.7740    0.3545

    0.0161    0.2530    0.4739    0.9321

    0.5569    0.4878    0.3637    0.7993

    0.7084    0.7679    0.7107    0.5885

    0.0126    0.1080    0.3441    0.3911

    0.0533    0.5263    0.7553    0.9838

    0.8175    0.6891    0.3861    0.7708

    0.2937    0.2439    0.3711    0.3399

    0.2900    0.2013    0.3438    0.3969

    0.0128    0.6108    0.7233    0.4664

    0.3230    0.2318    0.2645    0.4616

    0.7096    0.7053    0.4324    0.9190

    0.8851    0.7431    0.3509    0.6583
    0.0532    0.7236    0.5605    1.0000

    0.2832    0.2522    0.3102    0.4054

    0.0161    0.2095    0.3904    0.7330

    0.4695    0.4487    0.3513    0.5793

    0.2272    0.2441    0.3256    0.2624

    0.0899    0.0894    0.0558    0.2289

    0.2668    0.2341    0.3450    0.3518

    0.0160    0.1022    0.0923    0.2067



Bobot_bias_input =


    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752
    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752

    2.7752



Bobot_Lapisan =


  Columns 1 through 12


    0.1621    0.0961   -0.0297    0.5631    1.0356    0.3179
0.2229    0.6618    0.5244    0.3115   -0.7613    3.8449


  Columns 13 through 24


    3.0533   -0.2910   -1.4716    0.5099    0.6213   -0.1003
3.9059   -0.0265   -0.0638    2.1449    0.1296  -11.0927


  Column 25
```

```
   -0.1983


Bobot_Bias_lapisan =

   -0.0139


Y1 =

  Columns 1 through 12

    0.1573    0.1391    0.1322    0.1124    0.0966    0.0836
0.0600    0.0615    0.0695    0.0623    0.0752    0.1031

  Columns 13 through 24

    0.1852    0.3929    0.4283    0.4725    0.5527    0.7005
0.7325    0.8477    0.8202    0.8773    0.8118    0.6488

  Columns 25 through 36

    0.5174    0.7420    0.8385    0.4620    0.4876    0.5126
0.4538    0.4481    0.4782    0.3886    0.3602    0.3899

  Columns 37 through 48
```

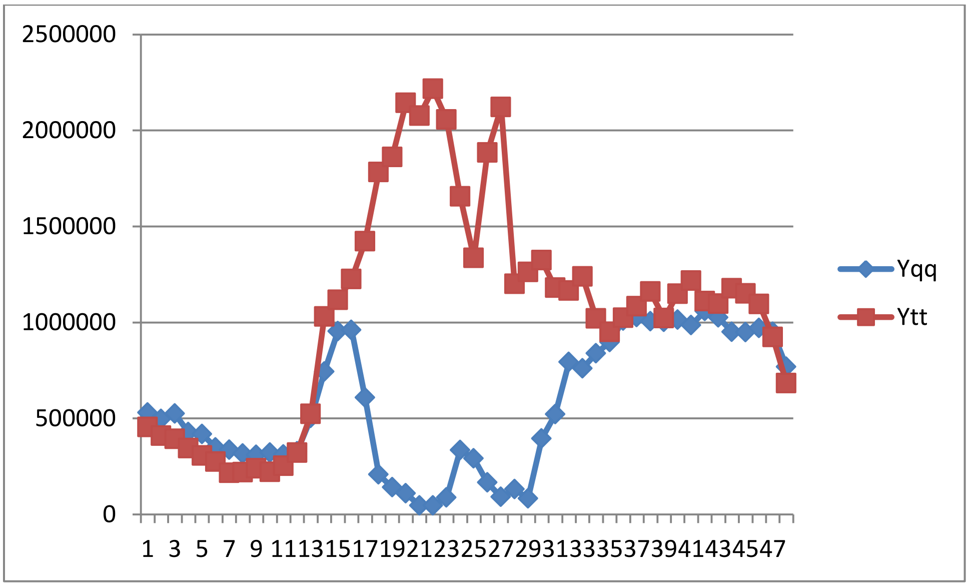
```
    0.4145    0.4458    0.3893    0.4415    0.4690    0.4253
0.4201    0.4526    0.4419    0.4194    0.3493    0.2510



Y2 =


  Columns 1 through 12


    0.2125    0.1981    0.2101    0.1680    0.1639    0.1334
0.1285    0.1199    0.1170    0.1223    0.1179    0.1249


  Columns 13 through 24


    0.1998    0.3048    0.3957    0.3989    0.2463    0.0724
0.0436    0.0304    0.0025    0.0024    0.0209    0.1280


  Columns 25 through 36


    0.1086    0.0546    0.0221    0.0393    0.0189    0.1538
0.2087    0.3265    0.3121    0.3460    0.3715    0.4199


  Columns 37 through 48


    0.4274    0.4180    0.4171    0.4217    0.4097    0.4415
0.4270    0.3946    0.3937    0.4030    0.3943    0.3159



C =
  682.0012

  683.0007
```

```
Root Meant Square Error = 0.005815

Nilai Yqq dan Ytt :
```

Prediksi Selasa	
Yqq	Ytt
530573	454551
497297	409899
524994	393198
427951	344736
418458	305955
348003	274299
336681	216566
316878	220230
310090	239872
322320	222104
312167	253729
328380	321926
501355	522733
743671	1030735
953645	1117291
960902	1225559
608761	1421676
207294	1783238
140727	1861474
110303	2143187
45966	2076126
45560	2215666
88425	2055528
335643	1656746
290886	1335408
166124	1884833
91110	2120683
130898	1199852
83678	1262367

395123	1323640
521883	1179785
793911	1165896
760546	1239505
838873	1020317
897620	950784
1009407	1023435
1026664	1083574
1004983	1160188
1002833	1022092
1013548	1149668
985823	1217026
1059335	1110061
1025827	1097379
951058	1176738
948822	1150569
970362	1095696
950350	924086
769376	683621

Grafiknya :



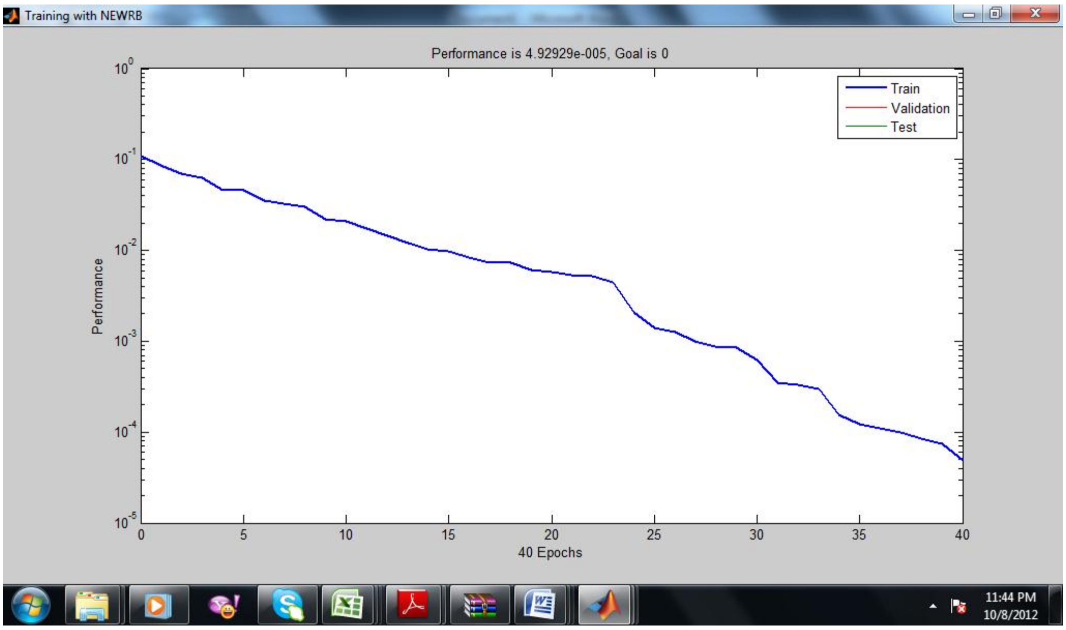
RABU

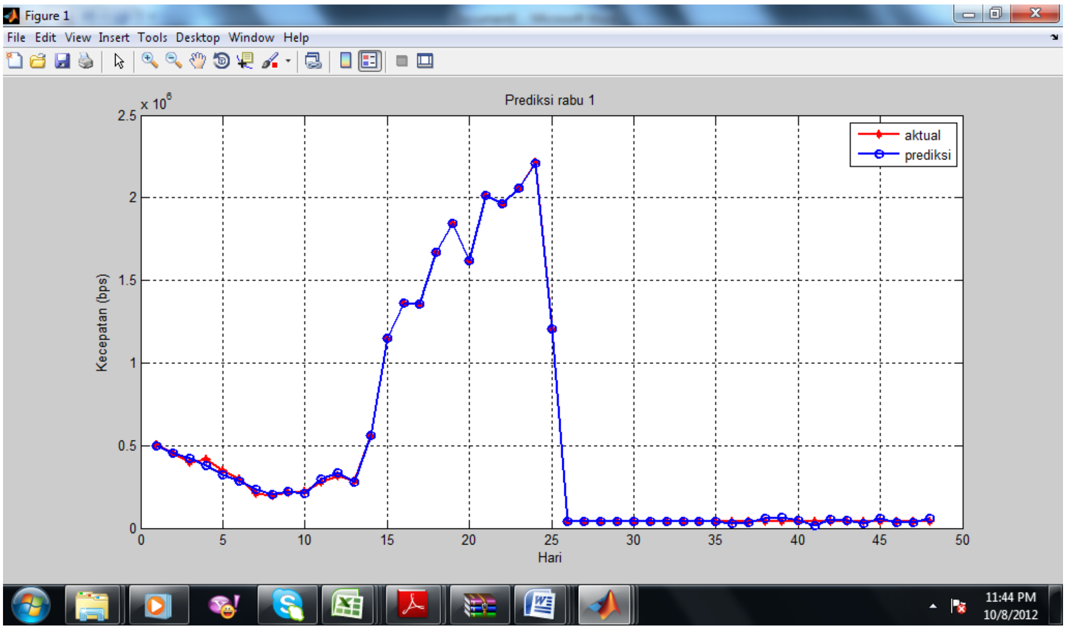
Listingnya :

```
clc;
clear;
%load target
   load rabu_targetout.mat
    j=rabu_targetout'


%load data 1
    load rabu_out1.mat
    P1=rabu_out1'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load rabu_out2.mat
    P2=rabu_out2';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];


%bentuk jaringan RBF
%     spread=0.1;

    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,1,40,1)

    % [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}
%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
```

```
      yqq=abs(yqq);
      yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF

     n=1:length(Y1);
     figure
     C=plot(n,j,'-r*',n,ytt,'-bo')
     set(C,'LineWidth',1.5)
     title('Prediksi rabu 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
     grid;



%menampilkan RMSE
     rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
     disp(['Root Meant Square Error = ' num2str(rmse)])
```

grafik yg didapat :





```
NEWRB, neurons = 0, MSE = 0.109797

NEWRB, neurons = 2, MSE = 0.0691805

NEWRB, neurons = 3, MSE = 0.0629055

NEWRB, neurons = 4, MSE = 0.045829

NEWRB, neurons = 5, MSE = 0.0457044

NEWRB, neurons = 6, MSE = 0.0356485

NEWRB, neurons = 7, MSE = 0.0330355

NEWRB, neurons = 8, MSE = 0.0303033

NEWRB, neurons = 9, MSE = 0.0222442

NEWRB, neurons = 10, MSE = 0.0208013

NEWRB, neurons = 11, MSE = 0.01717

NEWRB, neurons = 12, MSE = 0.0143135

NEWRB, neurons = 13, MSE = 0.0121677

NEWRB, neurons = 14, MSE = 0.0102948

NEWRB, neurons = 15, MSE = 0.00976695

NEWRB, neurons = 16, MSE = 0.00839639

NEWRB, neurons = 17, MSE = 0.00736112
```

```
NEWRB, neurons = 18, MSE = 0.00728911
NEWRB, neurons = 19, MSE = 0.00615862
NEWRB, neurons = 20, MSE = 0.00576371
NEWRB, neurons = 21, MSE = 0.00533229
NEWRB, neurons = 22, MSE = 0.0051674
NEWRB, neurons = 23, MSE = 0.00443429
NEWRB, neurons = 24, MSE = 0.00208753
NEWRB, neurons = 25, MSE = 0.00139638
NEWRB, neurons = 26, MSE = 0.00125377
NEWRB, neurons = 27, MSE = 0.00100219
NEWRB, neurons = 28, MSE = 0.000869988
NEWRB, neurons = 29, MSE = 0.00085127
NEWRB, neurons = 30, MSE = 0.000625253
NEWRB, neurons = 31, MSE = 0.000351418
NEWRB, neurons = 32, MSE = 0.000332854
NEWRB, neurons = 33, MSE = 0.000299703
NEWRB, neurons = 34, MSE = 0.00015609
NEWRB, neurons = 35, MSE = 0.000123612
NEWRB, neurons = 36, MSE = 0.000111441
NEWRB, neurons = 37, MSE = 9.96903e-005
NEWRB, neurons = 38, MSE = 8.48226e-005
NEWRB, neurons = 39, MSE = 7.45234e-005
NEWRB, neurons = 40, MSE = 4.92929e-005

net =


    Neural Network object:
```

```
architecture:

     numInputs: 1
     numLayers: 2
   biasConnect: [1; 1]
  inputConnect: [1; 0]
  layerConnect: [0 0; 1 0]
 outputConnect: [0 1]

    numOutputs: 1  (read-only)
numInputDelays: 0  (read-only)
numLayerDelays: 0  (read-only)

subobject structures:

        inputs: {1x1 cell} of inputs
        layers: {2x1 cell} of layers
       outputs: {1x2 cell} containing 1 output
        biases: {2x1 cell} containing 2 biases
  inputWeights: {2x1 cell} containing 1 input weight
  layerWeights: {2x2 cell} containing 1 layer weight

functions:

      adaptFcn: (none)
     divideFcn: (none)
   gradientFcn: (none)
       initFcn: (none)
```

```
        performFcn: 'mse'
          plotFcns: {}
          trainFcn: (none)

    parameters:

        adaptParam: (none)
       divideParam: (none)
     gradientParam: (none)
         initParam: (none)
      performParam: (none)
        trainParam: (none)

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight matrix
                LW: {2x2 cell} containing 1 layer weight matrix
                 b: {2x1 cell} containing 2 bias vectors

    other:

              name: ''
          userdata: (user information)


tr =

    epoch: [1x41 double]
```

```
         perf: [1x41 double]


Bobot_input =

    0.9169    1.0000    0.8203    0.8476
    0.5840    0.9007    0.9952    0.9234
    0.8284    0.8080    0.7822    0.9686
    0.7528    0.8318    0.4375    0.9184
    0.3824    0.3901    0.7564    0.4085
    0.8925    0.9151    0.3901    0.8872
    0.9451    0.6893    0.6463    0.9649
    0.5600    0.8697    0.4481    0.8275
    0.6596    0.4756    0.7651    0.5656
    0.8070    0.8941    0.7394    0.7906
    0.5527    0.5049    0.6475    0.6104
    0.2457    0.2634    0.4723    0.6010
    0.1807    0.0971    0.4277    0.5642
    0.4431    0.4762    0.7964    0.5503
    0.2718    0.2710    0.6203    0.7328
    0.4484    0.2449    0.6386    0.3946
    0.1861    0.0513    0.3472    0.1799
    0.3288    0.1408    0.2750    0.4909
    0.3240    0.1652    0.2810    0.3966
    0.3044    0.1655    0.2504    0.4519
    0.2756    0.1678    0.4546    0.7298
    0.2493    0.1759    0.2631    0.4225
    0.6932    0.6971    0.8392    0.7640
    0.8260    0.6306    0.5859    0.8259

    0.1092    0.0758    0.1196    0.0765

    0.4562    0.8369    0.4807    0.8612

    0.5596    0.3732    0.7296    0.3748

    0.1532    0.1065    0.1134    0.0971

    0.1525    0.1075    0.1572    0.1943

    0.0901    0.0347    0.1891    0.1113

    0.0614    0.0698    0.0466    0.0403

    0.3033    0.1660    0.2369    0.3902

    0.0639    0.1491    0.0247    0.0126

    0.0771    0.0671    0.0364    0.0269

    0.2820    0.2119    0.2126    0.4637

    0.0880    0.0809    0.0612    0.0604

    0.3419    0.2233    0.1880    0.4851

    0.0817    0.0904    0.0088    0.0124

    0.3184    0.1819    0.2048    0.4207

    0.3128    0.1914    0.2227    0.3928


Bobot_bias_input =


    0.8326

    0.8326

    0.8326

    0.8326

    0.8326

    0.8326

    0.8326
```

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

```
    0.8326

    0.8326

    0.8326

    0.8326

    0.8326



Bobot_Lapisan =


  1.0e+005 *


  Columns 1 through 12


    0.0545    0.0137   -0.0244    0.0104    0.0224   -0.0021
0.0167    0.0172    0.0587   -0.0867    0.0632   -0.3624


  Columns 13 through 24


   -0.1316    0.0015    0.0789   -0.1526    0.1999   -0.2274
2.0359   -1.9750    0.1114    3.2278   -0.0385   -0.0042


  Columns 25 through 36


   -2.7061    0.0079    0.0085    1.9088   -1.3684    0.3710   -
0.1520   -3.6117    0.3315    1.8476   -0.8836    1.1800


  Columns 37 through 40


   -0.3763   -1.9217    4.8778   -2.4071
```

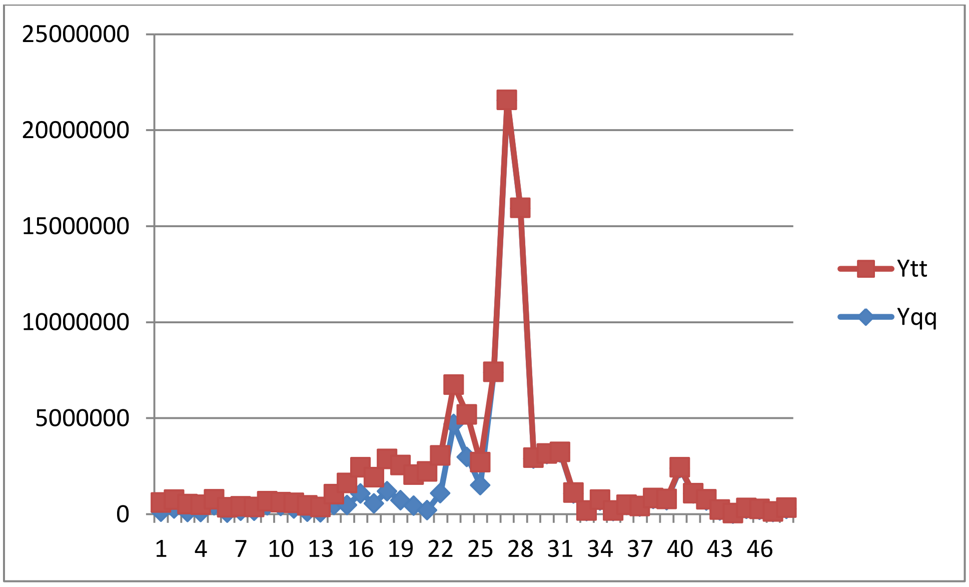
```
Bobot_Bias_lapisan =

 -295.0441


Y1 =

  Columns 1 through 12

    0.1786    0.1588    0.1420    0.1210    0.0925    0.0744
0.0495    0.0342    0.0432    0.0377    0.0827    0.0998

  Columns 13 through 24

    0.0710    0.2092    0.4984    0.6028    0.5983    0.7522
0.8394    0.7283    0.9221    0.8974    0.9412    1.0171

  Columns 25 through 36

    0.5243   -0.0443   -0.0444   -0.0444   -0.0444   -0.0443   -
0.0445   -0.0441   -0.0443   -0.0440   -0.0437   -0.0515

  Columns 37 through 48

   -0.0473   -0.0355   -0.0325   -0.0400   -0.0705   -0.0370   -
0.0410   -0.0505   -0.0360   -0.0472   -0.0459   -0.0366
```

```
Y2 =


  Columns 1 through 12


   -0.0614    0.1177    0.0387    0.0385    0.1841   -0.0423   -
0.0882   -0.0838   -0.2025   -0.1875   -0.1337   -0.0618


  Columns 13 through 24


   -0.0503   -0.2155    0.1970    0.4485    0.2331    0.5015   -
0.3160    0.1823   -0.0960   -0.4764   -1.9980    1.2564


  Columns 25 through 36


   -0.6516   -3.1384   -9.1423   -6.7618    1.2230   -1.3371   -
1.3696   -0.4694    0.0593    0.2952    0.0584   -0.2107


  Columns 37 through 48


   -0.1791   -0.3459   -0.3207   -1.0242   -0.4653   -0.3128   -
0.0945    0.0047   -0.1247   -0.1094   -0.0568    0.1116



C =


  223.0034

  224.0024


Root Meant Square Error = 0.0058814

Nilai Yqq dan Ytt :
```

Prediksi Rabu	
Yqq	Ytt
123849	496693
298567	456186
112092	421796
111800	378727
455155	320604
78887	283411
187150	232640
176780	201316
456656	219642
421285	208406
294404	300384
124787	335418
97780	276556
487293	559273
485599	1150847
1078753	1364281
570615	1355214
1203840	1669992
724278	1848237
450822	1620997
205504	2017418
1102704	1967021
4691340	2056452
2984282	2211724
1515959	1203686
7380946	40710
21541373	40531
15926783	40439
2905449	40585
3132540	40770
3209347	40334
1086143	41151
160699	40636
717166	41378
158710	41997
476071	25965
401445	34592
794827	58618

735542	64870
2394591	49535
1076388	12804
716867	55670
202020	47366
31984	28112
273086	57764
237173	34752
113112	37387
284054	56493

Grafiknya :



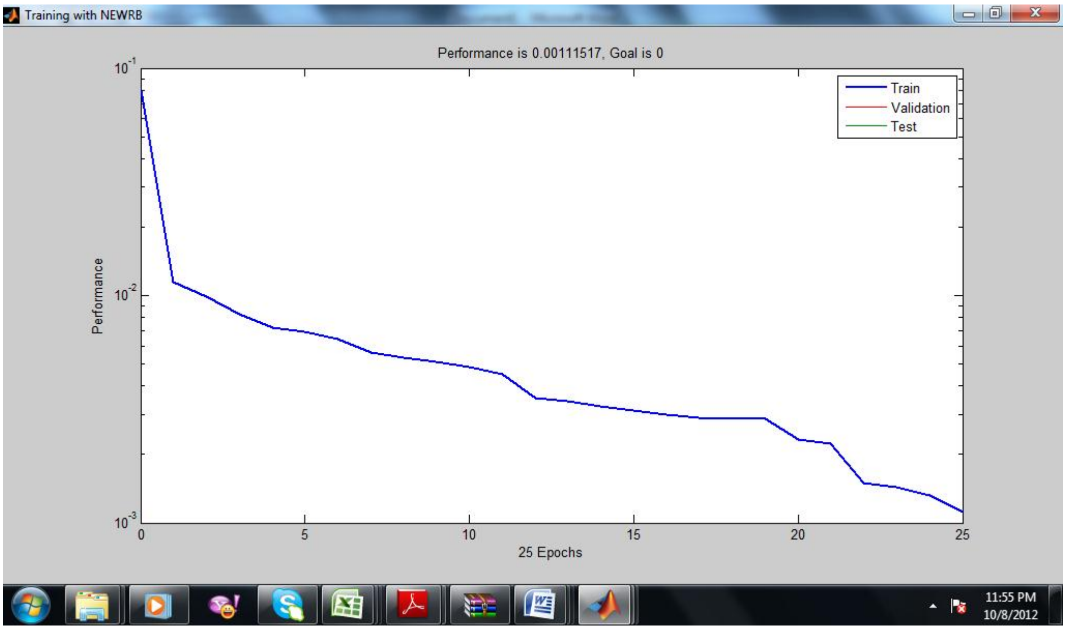
KAMIS

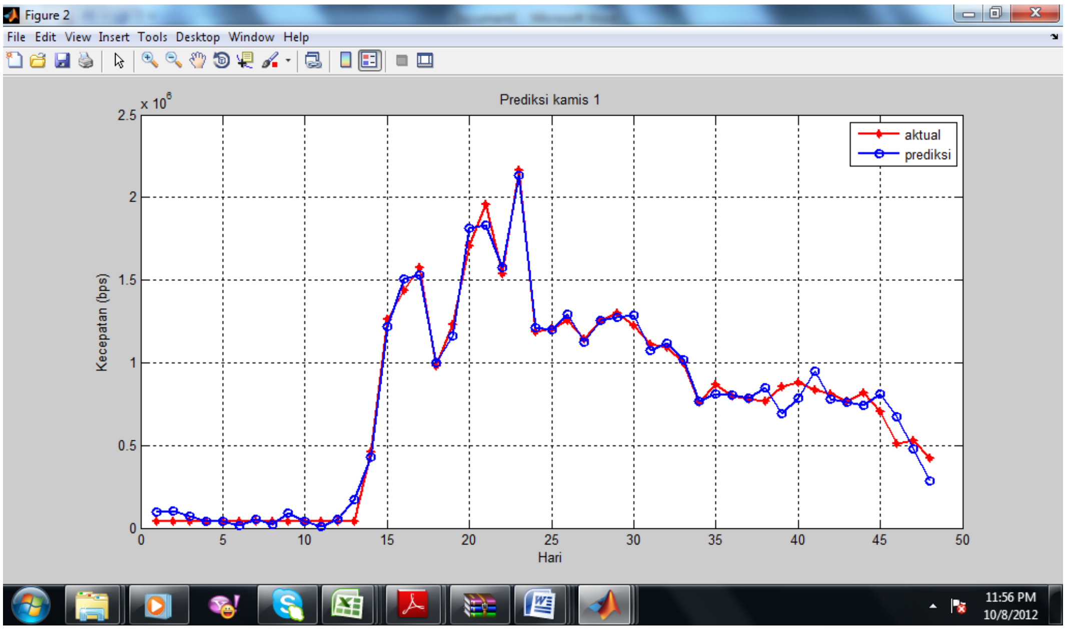
Listingnya :

```
clc;
clear;
%load target
   load kamis_targetout.mat
    j=kamis_targetout'


%load data 1
    load kamis_out1.mat
    P1=kamis_out1'
    P=P1
    size(P)

%normalisasi
    a=min(min(P));
    b=max(max(P));
    p=[(P-a)/(b-a)];
    t=[(j-a)/(b-a)];


%load data 2
    load kamis_out2.mat
    P2=kamis_out2';
    R=P2;
    size(R)

%normalisasi
    c=min(min(R));
    d=max(max(R));
    q=[(R-c)/(d-c)];


%bentuk jaringan RBF
%     spread=0.1;

    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
   [net,tr] = newrb(p,t,0,3,25,1)

    % [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
            Bobot_input = net.IW{1,1}
            Bobot_bias_input = net.b{1,1}
            Bobot_Lapisan = net.LW{2,1}
            Bobot_Bias_lapisan = net.b{2,1}
%simulasi
%data training
     Y1 = sim(net,p)
     ytt=((b-a)*Y1)+ a;
     ytt=abs(ytt);
     ytt=round(ytt);

     Y2 = sim(net,q)
     yqq=((d-c)*Y2)+ c;
     yqq=abs(yqq);
     yqq=round(yqq);

 %menggambar grafik perbandingan aktual,RBF

    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi kamis 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
```

```
    grid;


%menampilkan RMSE
    rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
    disp(['Root Meant Square Error = ' num2str(rmse)])
```

grafik dari matlab :





```
NEWRB, neurons = 0, MSE = 0.0796076
NEWRB, neurons = 2, MSE = 0.00991672
NEWRB, neurons = 3, MSE = 0.0083172
NEWRB, neurons = 4, MSE = 0.0072616
NEWRB, neurons = 5, MSE = 0.00689846
NEWRB, neurons = 6, MSE = 0.00641666
NEWRB, neurons = 7, MSE = 0.00562462
NEWRB, neurons = 8, MSE = 0.00531991
NEWRB, neurons = 9, MSE = 0.00512123
NEWRB, neurons = 10, MSE = 0.00482702
NEWRB, neurons = 11, MSE = 0.00450072
NEWRB, neurons = 12, MSE = 0.00355826
NEWRB, neurons = 13, MSE = 0.00341649
NEWRB, neurons = 14, MSE = 0.00326473
NEWRB, neurons = 15, MSE = 0.0031066
NEWRB, neurons = 16, MSE = 0.00300334
NEWRB, neurons = 17, MSE = 0.00288882
NEWRB, neurons = 18, MSE = 0.00286977
NEWRB, neurons = 19, MSE = 0.00286674
NEWRB, neurons = 20, MSE = 0.00233775
NEWRB, neurons = 21, MSE = 0.00223474
NEWRB, neurons = 22, MSE = 0.00149242
NEWRB, neurons = 23, MSE = 0.00143247
NEWRB, neurons = 24, MSE = 0.00131617
NEWRB, neurons = 25, MSE = 0.00111517
```

```
net =

    Neural Network object:

    architecture:

         numInputs: 1
         numLayers: 2
       biasConnect: [1; 1]
      inputConnect: [1; 0]
      layerConnect: [0 0; 1 0]
     outputConnect: [0 1]

        numOutputs: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)

    subobject structures:

            inputs: {1x1 cell} of inputs
            layers: {2x1 cell} of layers
           outputs: {1x2 cell} containing 1 output
            biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1 input weight
      layerWeights: {2x2 cell} containing 1 layer weight
    functions:

          adaptFcn: (none)
         divideFcn: (none)
       gradientFcn: (none)
           initFcn: (none)
        performFcn: 'mse'
          plotFcns: {}
          trainFcn: (none)

    parameters:

        adaptParam: (none)
       divideParam: (none)
     gradientParam: (none)
         initParam: (none)
      performParam: (none)
        trainParam: (none)

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight
matrix
                LW: {2x2 cell} containing 1 layer weight
matrix
                 b: {2x1 cell} containing 2 bias vectors
    other:

              name: ''
          userdata: (user information)



tr =

    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25]
     perf: [1x26 double]



Bobot_input =

    1.0000    0.6865    0.8508    0.9407
    0.9894    0.6942    0.7647    0.8919
    0.6223    0.6652    0.5123    0.6433
    0.0836    0.0622    0.0240    0.0571
    0.2114    0.1159         0    0.0885
    0.1546    0.0704    0.0153    0.0610
    0.5252    0.3383    0.7318    0.3740
    0.1061    0.0763    0.0028    0.0921
    0.8473    0.6032    0.4559    0.5464
    0.9159    0.6126    0.4131    0.7643
    0.9047    0.6233    0.8661    0.7195
    0.1899    0.1609    0.4045    0.4129
    0.1401    0.0675    0.0078    0.0848

    0.2357    0.1223    0.3296    0.5424

    0.1039    0.0621    0.4049    0.3349

    0.7903    0.7334    0.6681    0.5340

    0.2992    0.1371    0.4215    0.3648

    0.7980    0.7147    0.9278    0.6449

    0.6021    0.3982    0.4287    0.6134

    0.7347    0.7137    0.8552    0.5974

    0.4548    0.1845    0.3552    0.6526

    0.7244    0.4998    0.4838    0.6026

    0.1808    0.0995    0.0985    0.1152

    0.1188    0.0824    0.0473    0.1113

    0.2997    0.1816    0.4140    0.3641


Bobot_bias_input =


    0.2775

    0.2775

    0.2775

    0.2775

    0.2775

    0.2775

    0.2775

    0.2775

    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775
    0.2775


Bobot_Lapisan =

  1.0e+007 *

  Columns 1 through 12

   -0.0200    0.0097   -0.0206    0.8246    0.2903   -1.6293
-0.0141   -0.1331    0.0133    0.0253    0.0831    0.0267
```

```
  Columns 13 through 24

    1.2593    0.0926   -0.0498   -0.0184    0.1869   -0.1688
0.1974    0.1778   -0.1082   -0.2125    0.7345   -1.3061

  Column 25

   -0.2406



Bobot_Bias_lapisan =

   -4.6778



Y1 =

  Columns 1 through 12

    0.0006    0.0044   -0.0128   -0.0279   -0.0296   -0.0399
-0.0214   -0.0388   -0.0024   -0.0683   -0.0544   -0.0220

  Columns 13 through 24

    0.0368    0.1638    0.5576    0.7000    0.7141    0.4499
0.5309    0.8523    0.8632    0.7364    1.0112    0.5544

  Columns 25 through 36
```

```
    0.5468    0.5939    0.5123    0.5752    0.5862    0.5913
0.4861    0.5078    0.4564    0.3346    0.3539    0.3507

  Columns 37 through 48

    0.3429    0.3737    0.2970    0.3426    0.4245    0.3398
0.3292    0.3198    0.3538    0.2872    0.1905    0.0925


Y2 =

  Columns 1 through 12

    0.0212   -0.0146   -0.0180   -0.0194   -0.0305   -0.0450
-0.0491   -0.0553   -0.0837   -0.0695   -0.0315   -0.0163

  Columns 13 through 24

    0.0083    0.0537    0.5021    0.5434    0.6248    0.3504
0.1574    0.1208    0.7300    0.6032    0.7585    0.6767

  Columns 25 through 36

    0.5417   -0.2981    0.4836    0.8121    0.0294    0.2864
0.5295    0.3729    0.4518    0.3966    0.2605    0.2761

  Columns 37 through 48
```

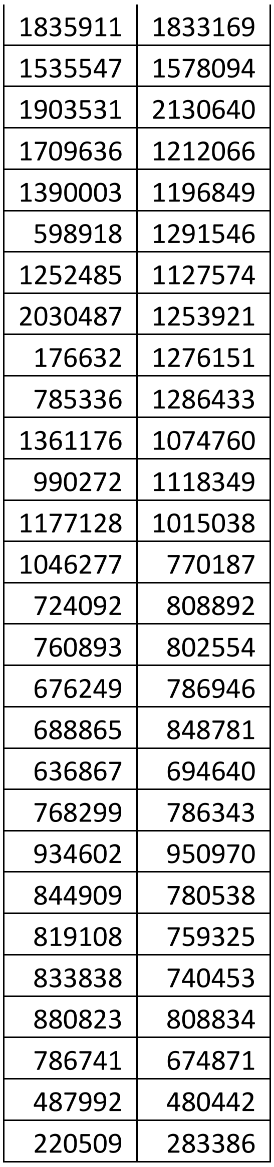
```
    0.2403    0.2457    0.2237    0.2792    0.3494    0.3116
0.3007    0.3069    0.3267    0.2870    0.1609    0.0479
```

```
C =

  448.0044

  449.0034
```

```
Root Meant Square Error = 0.030948
```

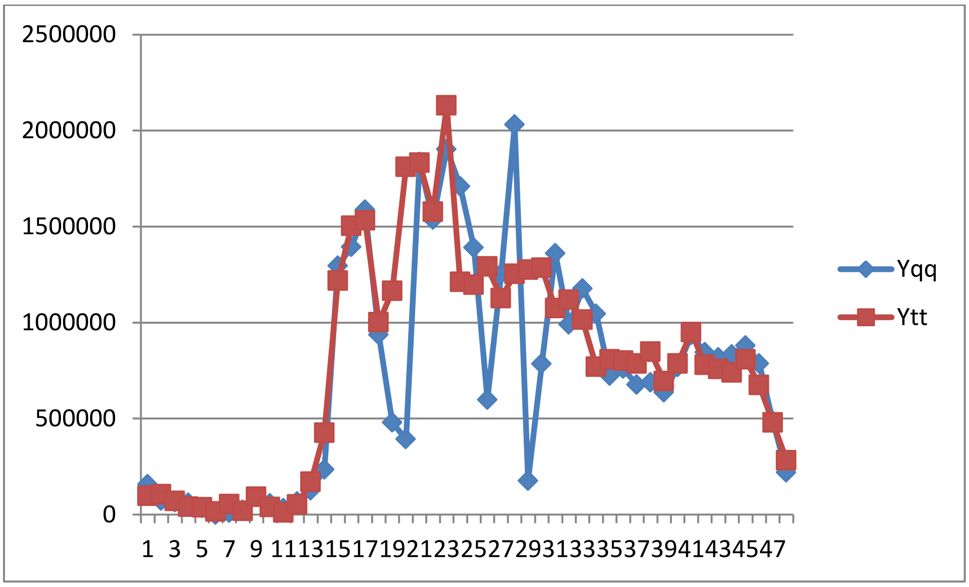
Nilai Yqq dan Ytt

Prediksi kamis	
Yqq	Ytt
157251	98630
72538	106081
64513	71595
61099	41266
34791	37872
381	17050
9158	54332
23845	19215
91286	92504
57653	39985
32330	12059
68463	53094
126711	171280
234110	426631
1296145	1218630
1393982	1504977
1586822	1533296
936989	1002039
479808	1164918
393077	1811193



1835911	1833169
1535547	1578094
1903531	2130640
1709636	1212066
1390003	1196849
598918	1291546
1252485	1127574
2030487	1253921
176632	1276151
785336	1286433
1361176	1074760
990272	1118349
1177128	1015038
1046277	770187
724092	808892
760893	802554
676249	786946
688865	848781
636867	694640
768299	786343
934602	950970
844909	780538
819108	759325
833838	740453
880823	808834
786741	674871
487992	480442
220509	283386

Grafiknya :


2500000
2000000
1500000
1000000
500000
0
1 3 5 7 9 11131517192123252729313335373941434547
Yqq
Ytt