

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Kebutuhan Sistem

Sebelum mengimplementasikan dan menjalankan aplikasi Peramalan PAD Kabupaten Gresik dengan Penerapan Sistem Neuro-Fuzzy Struktur Mod_ANFIS, terlebih dahulu komponen-komponen utama komputer yang mendukung setiap proses harus sudah terpasang.

4.1.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak aplikasi Peramalan PAD Kabupaten Gresik dengan Penerapan Sistem Neuro-Fuzzy Struktur Mod_ANFIS ini adalah:

1. Sistem operasi windows XP
2. Borland Delphi 6.0
3. Microsoft Access
4. Power Designer
5. Microsoft Visio

4.1.2 Kebutuhan Perangkat Keras

Untuk dapat menjalankan aplikasi Peramalan PAD Kabupaten Gresik dengan Penerapan Sistem Neuro-Fuzzy Struktur Mod_ANFIS ini dibutuhkan persyaratan minimal sebagai berikut:

1. Processor minimal Pentium III
2. RAM minimal 64 Mb
3. Harddisk minimal 10 Gb

4. Monitor dengan resolusi 1024 x 768
5. Keyboard, mouse dan printer.

4.2 Implementasi Sistem

Setelah semua komponen yang mendukung aplikasi Peramalan PAD Kabupaten Gresik dengan Penerapan Sistem Neuro-Fuzzy Struktur Malar NFIS terpasang, proses selanjutnya bagaimana mewujudkan rancangan sistem ke dalam sebuah program Peramalan PAD. Berikut ditunjukkan modul bagian yang penting dalam implementasi sistem.

4.2.1 Proses Belajar

Berdasarkan alur proses belajar pada Gambar 3.3 di muka, maka proses yang terjadi dapat dijabarkan sebagai berikut:

A. Tahap Maju

Tahap maju terdiri dari lapisan 1 sampai dengan lapisan 5, dijelaskan sebagai berikut:

1. Proses Lapisan 1 (Fuzzyfikasi)

Proses fuzzyfikasi adalah proses mengubah masukan nilai *crisp* (nilai aktual) ke derajat tertentu yang sesuai dengan aturan besaran fungsi keanggotaan. Terlebih dahulu ditentukan parameter fungsi keanggotaan ($a_1..a_4$, $b_1..b_4$, $c_1..c_4$) dan pengambilan data PAD dari database Data_PAD, setelah itu fungsi keanggotaan diterapkan pada lapisan ini (dalam Tugas Akhir ini dipilih fungsi bell), dengan demikian simpul di lapisan ini merupakan fungsi bell, dimana implementasinya dalam program adalah sebagai berikut:

```

for i:=1 to 12 do
  begin
    //n1a
    temp:=abs((in1[i]-(c1))/(a1))*abs((in1[i]-(c1))/(a1));
    temp:=exp(ln(temp)*(b1));
    n1a[i]:=1/(1+temp);

    //n2a
    temp:=abs((in1[i]-(c2))/(a2))*abs((in1[i]-(c2))/(a2));
    temp:=exp(ln(temp)*(b2));
    n2a[i]:=1/(1+temp);

    //n3a
    temp:=abs((in2[i]-(c3))/(a3))*abs((in2[i]-(c3))/(a3));
    temp:=exp(ln(temp)*(b3));
    n3a[i]:=1/(1+temp);

    //n4a
    temp4:=abs((in2[i]-(c4))/(a4))*abs((in2[i]-(c4))/(a4));
    temp4:=exp(ln(temp4)*(b4));
    n4a[i]:=1/(1+temp4);
  end;

```

Keterangan:

Untuk semua keluaran simpul pada tabel di atas ini diberi simbol 'a', sehingga pada lapisan 1 ini diperoleh keluaran simpul n1a s.d n4a. Tanda a untuk membedakan dengan nilai keluaran simpul yang baru yang diberi simbol 'b' (setelah dikoreksi).

2. Proses Lapisan 2 (Logika AND)

Pada lapisan ini diterapkan logika fuzzy AND sebagai fungsi simpul, implementasi dalam program adalah sebagai berikut:

```

n5a[i] :=Min(n1a[i],n3a[i]);
n6a[i] :=Min(n2a[i],n4a[i]);

```

Keterangan :

Penerapan logika fuzzy AND pada lapisan ini adalah memilih nilai yang terkecil dari simpul-simpul yang masuk, sehingga pada lapisan 2 ini diperoleh keluaran simpul n5a dan n6a.

3. Proses Lapisan 3 (Normalisasi)

Pada lapisan ini dilakukan normalisasi dari sinyal yang masuk, implementasinya dalam program adalah sebagai berikut:

```

ntot_a[i]:=n5a[i]+n6a[i];
if ntot_a[i]=0 then
  begin
    n7a[i]:=0;
    n8a[i]:=0;
  end
else
  begin
    n7a[i]:=n5a[i]/ntot_a[i];
    n8a[i]:=n6a[i]/ntot_a[i];
  end;

```

Keterangan:

Proses normalisasi pada lapisan ini adalah dengan cara membagi nilai salah satu simpul yang masuk dengan nilai total dari simpul-simpul yang masuk, sehingga pada lapisan 3 ini diperoleh keluaran simpul n7a dan n8a.

4. Proses Lapisan 4 (Defuzzyfikasi dengan metode LSE)

Proses defuzzyfikasi adalah proses mengubah kembali nilai besaran fuzzy menjadi nilai *crisp* (nilai aktual). Dengan metode *Least-Squares Estimator* (LSE) diperoleh parameter-parameter konsekuen (p_1 , q_1 , r_1 , p_2 , q_2 dan r_2). Implementasinya dalam program sebagai berikut:

```

For i:= 1 to 12 do
  begin
    A[i,1]:=n7a[i]*in1[i];
    A[i,2]:=n7a[i]*in2[i];
    A[i,3]:=n7a[i];
    A[i,4]:=n8a[i]*in1[i];
    A[i,5]:=n8a[i]*in2[i];
    A[i,6]:=n8a[i];
  end;

```

Keterangan:

Dari sinyal yang masuk pada lapisan ini diperoleh matrik A, untuk Mod_ANFIS matrik A dituliskan sebagai berikut:

$$A = [(n7a \ x) \ (n7a \ y) \ n7a \ (n8a \ x) \ (n8a \ y) \ n8a]$$

```
//Proses mencari A Transpose
for l:= 1 to 6 do
begin
for j:= 1 to 12 do
begin
ATranspose[l,j]:= A[j,l];
end;
end;
end;
```

Keterangan:

Penggalan program di atas adalah untuk mencari matrik transpose. Dimana baris, kolom matrik A diubah menjadi kolom, baris.

```
//proses mencari ATranspose * A
for l:=1 to 6 do
begin
for j:=1 to 6 do
begin
bantu:=0;
for k:=1 to 12 do
begin
bantu:=bantu + ATranspose[l,k]*A[k,j];
end;
AHasil[l,j]:=bantu;
end;
end;
end;
```

Keterangan:

Penggalan program di atas adalah untuk mencari hasil perkalian matrik A transpose dengan matrik A. Dimana hasil dari perkalian matrik ini adalah matrik dengan ordo 6x6

```
//invers Matrik ---> invers(AT[6,12]*A[12,6])=mInvers[6,6]
mat_A:=TMatrix.Create(nil);
mat_A.Resize(6,6); //resize(col,baris)
for l:=1 to 6 do
for j:= 1 to 6 do
mat_A.Elem[j,l]:=AHasil[l,j];
```

```

if mat_A.Invert then
    cekMatrik:=true
else
    cekMatrik:=false;

//Hasil Invers
for l:=1 to 6 do
    for j:= 1 to 6 do
        begin
            AHasil[l,j]:=mat_A.Elem[j,l]; //mat_A.elem[col,row]
            mInvers[l,j]:=mat_A.Elem[j,l];
        end;

```

Keterangan:

Penggalan program di atas adalah untuk menginverskan hasil perkalian matrik A transpose dengan matrik A.

```

//Perkalian matrik mInvers dg Transpose
for k:= 1 to 6 do
    begin
        bantu:=0;
        for l:= 1 to 12 do
            begin
                for j:= 1 to 6 do
                    begin
                        bantu:=bantu+(AHasil[k,j]*ATranspose[j,l]);
                    end;
                AHasil2[k,l]:=bantu;
                bantu:=0;
            end;
        end;
    end;

```

Keterangan:

Penggalan program di atas adalah untuk mencari hasil perkalian matrik invers ($A^T \times A$) dengan matrik A transpose.

```

//Perkalian AHasil2[6,12] * U[12,1]
for j:= 1 to 6 do
    begin
        bantu:=0;
        for l:= 1 to 1 do
            begin
                for k:= 1 to 12 do
                    begin
                        bantu:=bantu+(AHasil2[j,k]*out1[k]);
                    end;
                mat_Hasil[j]:=bantu;
                bantu:=0;
            end;
        end;
    end;

```

Keterangan:

Penggalan program di atas adalah untuk mencari hasil perkalian matrik $[A^T A]^{-1}$

A^T dengan output data aktual.

```
//Teta
p1:=mat_Hasil[1];
q1:=mat_Hasil[2];
r1:=mat_Hasil[3];
p2:=mat_Hasil[4];
q2:=mat_Hasil[5];
r2:=mat_Hasil[6];
```

Proses *LSE* ini akan menghasilkan parameter konsekuensi ($p1, q1, r1, p2, q2$ dan $r2$). Kemudian hasilnya digunakan untuk mencari nilai $f1$ dan $f2$. Hasil defuzzyfikasi berupa simpul $n9a$ dan $n10a$.

```
//Nilai f1 dan f2
for i:=1 to 12 do
begin
f1[i]:=((p1*in1[i])+(q1*in2[i])+r1);
f2[i]:=((p2*in1[i])+(q2*in2[i])+r2);

// Keluaran Layer emg
n9a[i] := n7a[i]*f1[i];
n10a[i]:= n8a[i]*f2[i];
end;
```

5. Proses Lapisan 5 (Keluaran Jaringan)

Lapisan 5 yang berarti keluaran dari jaringan adalah hasil penjumlahan semua sinyal yang masuk, implementasinya dalam program sebagai berikut :

```
//Output Jaringan --> penjumlahan semua sinyal yg masuk
out1[i]:=n9a[i]+n10a[i];

//error keluaran sistem
selisih[i]:=out1[i]-n11a[i];
```

Keterangan:

Hasil dari proses lapisan ini adalah $n11a$ yang merupakan keluaran jaringan., sekaligus dilakukan perhitungan untuk mencari besar *error* (selisih). Sampai di sini selesai tahap maju dari sistem aplikasi Peramalan PAD Kabupaten Gresik

Dengan Penerapan Sistem Neuro-Fuzzy Struktur Mod_ANFIS kemudian diteruskan ke tahap mundur.

B. Tahap Mundur

Proses Mundur dengan algoritma koreksi kesalahan dari lapisan 5 ke lapisan 1. Setelah diperoleh keluaran jaringan dari tahap maju, untuk selanjutnya *error* keluaran jaringan ini dipropagasibalik menggunakan aturan koreksi kesalahan yang baru yaitu dengan menggunakan algoritma EBP yang telah dimodifikasi. Implementasi aturan koreksi kesalahan Sistem Mod_ANFIS dalam program sebagai berikut:

```

for i:= 1 to 12 do
  begin
    //Hitung error di layer lima
    e11[i]:= -2 * (e11sin[i])

    //Hitung error di layer empat
    d11[i]:=-e11[i]/2;
    ftot[i]:=f1[i]+f2[i];
    d9[i]:=(d11[i]*f1[i])/ftot[i];
    d10[i]:=(d11[i]*f2[i])/ftot[i];

    //Hitung error di Layer tiga
    d7[i]:=d9[i]/f1[i];
    d8[i]:=d10[i]/f2[i];

    //Hitung error di Layer dua
    ntot_b[i]:=ntot_a[i];
    n5b1[i]:=(n7a[i]+d7[i])*ntot_b[i];
    n6b1[i]:=(n8a[i]+d8[i])*ntot_b[i];

    //Hitung error di Layer satu
    n1b[i]:= n5b1[i]; n3b[i] := n5b1[i];
    n2b[i]:= n6b1[i]; n4b[i] := n6b1[i];
  end;

```

Proses selanjutnya perubahan parameter fungsi keanggotaan pada layer 1 tahap mundur. Implementasi dalam program sebagai berikut:

```

//cari turunan parameter
//n1b
for i:= 1 to 12 do
  begin
    dal[i]:=(2*b1/a1)*n1b[i]*(1-n1b[i]);

```



```

tampung_a1[i]:=a1+da1[i];
if in1[i]=c1 then
begin
db1[i]:=0;
dc1[i]:=0;
tampung_b1[i]:=b1;
tampung_c1[i]:=c1;
end
else
begin
db1[i]:=(-2)*ln(abs((in1[i]-c1)/a1))*n1b[i]*(1-n1b[i]);
dc1[i]:=(2 * b1/(in1[i]-c1))*n1b[i]*(1-n1b[i]);
tampung_b1[i]:=b1+db1[i];
tampung_c1[i]:=c1+dc1[i];
end;

//n2b
da2[i]:=(2*b2/a2)*n2b[i]*(1-n2b[i]);
tampung_a2[i]:=a2+da2[i];
if in1[i]=c2 then
begin
db2[i]:=0;
dc2[i]:=0;
tampung_b2[i]:=b2;
tampung_c2[i]:=c2;
end
else
begin
db2[i]:=(-2)*ln(abs((in1[i]-c2)/a2))*n2b[i]*(1-n2b[i]);
dc2[i]:=(2 * b2/(in1[i]-c2))*n2b[i]*(1-n2b[i]);
tampung_b2[i]:=b2+db2[i];
tampung_c2[i]:=c2+dc2[i];
end;

//n3b
da3[i]:=(2*b3/a3)*n3b[i]*(1-n3b[i]);
tampung_a3[i]:=a3+da3[i];
if in2[i]=c3 then
begin
db3[i]:=0;
dc3[i]:=0;
tampung_b3[i]:=b3;
tampung_c3[i]:=c3;
end
else
begin
db3[i]:=(-2)*ln(abs((in2[i]-c3)/a3))*n3b[i]*(1-n3b[i]);
dc3[i]:=(2 * b3/(in2[i]-c3))*n3b[i]*(1-n3b[i]);
tampung_b3[i]:=b3+db3[i];
tampung_c3[i]:=c3+dc3[i];
end;

//n4b
da4[i]:=(2*b4/a4)*n4b[i]*(1-n4b[i]);
tampung_a4[i]:=a4+da4[i];
if in2[i]=c4 then
begin
db4[i]:=0;
dc4[i]:=0;

```

```

    tampung_b4[i]:=b4;
    tampung_c4[i]:=c4;
end
else
    begin
        db4[i]:=(-2)*ln(abs((in2[i]-c4)/a4))*n4b[i]*(1-n4b[i]);
        dc4[i]:=(2 * b4/(in2[i]-c4))*n4b[i]*(1-n4b[i]);
        tampung_b4[i]:=b4+db4[i];
        tampung_c4[i]:=c4+dc4[i];
    end;
end;

a1_baru:=0;      b1_baru:=0;      c1_baru:=0;
a2_baru:=0;      b2_baru:=0;      c2_baru:=0;
a3_baru:=0;      b3_baru:=0;      c3_baru:=0;
a4_baru:=0;      b4_baru:=0;      c4_baru:=0;

for i:= 1 to 12 do
    begin
        a1_baru:=a1_baru+tampung_a1[i];
        b1_baru:=b1_baru+tampung_b1[i];
        c1_baru:=c1_baru+tampung_c1[i];

        a2_baru:=a2_baru+tampung_a2[i];
        b2_baru:=b2_baru+tampung_b2[i];
        c2_baru:=c2_baru+tampung_c2[i];

        a3_baru:=a3_baru+tampung_a3[i];
        b3_baru:=b3_baru+tampung_b3[i];
        c3_baru:=c3_baru+tampung_c3[i];

        a4_baru:=a4_baru+tampung_a4[i];
        b4_baru:=b4_baru+tampung_b4[i];
        c4_baru:=c4_baru+tampung_c4[i];
    end;

a1:=a1_baru/12;      b1:=b1_baru/12;      c1:=c1_baru/12;
a2:=a2_baru/12;      b2:=b2_baru/12;      c2:=c2_baru/12;
a3:=a3_baru/12;      b3:=b3_baru/12;      c3:=c3_baru/12;
a4:=a4_baru/12;      b4:=b4_baru/12;      c4:=c4_baru/12;

```

Keterangan:

Perhitungan turunan fungsi keanggotaan diterapkan dari $da1$ sampai $dc4$. Hasil dari lapisan 1 tahap mundur adalah parameter premis yang telah diperbarui.

Proses Tahap maju maupun tahap mundur dilakukan sampai didapat *error* yang minimum atau sebanyak iterasi yang diinputkan oleh *user*.

4.2.2 Proses Validasi

Proses validasi adalah menguji sistem dengan menggunakan parameter fungsi keanggotaan yang telah diperbarui pada proses belajar, untuk data yang digunakan sama dengan data yang digunakan pada proses belajar. Pada proses validasi hanya digunakan tahap maju dari algoritma *Neuro-Fuzzy* seperti tahap maju pada proses belajar, tidak ada proses mundur untuk koreksi kesalahan.

4.2.3 Proses Uji Coba

Proses uji coba adalah menguji sistem dengan menggunakan parameter fungsi keanggotaan yang telah diperbarui pada proses belajar. Pada proses uji coba hanya digunakan tahap maju dari algoritma *Neuro-Fuzzy* seperti pada proses belajar dan validasi, perbedaannya hanya terletak pada data yang digunakan. Proses uji coba dibagi menjadi Uji Coba 1 dan Uji Coba 2.

Untuk uji coba yang pertama, sistem diberikan masukan data tahun 2001 dan tahun 2002. Keluaran jaringan adalah hasil peramalan untuk tahun 2003. Selanjutnya untuk menguji kebenaran hasil peramalan dibandingkan dengan data PAD tahun 2003 yang sudah disiapkan.

Untuk uji coba yang kedua, sistem diberikan masukan data tahun 2002 dan tahun 2003. Keluaran jaringan adalah hasil peramalan untuk tahun 2004. Selanjutnya untuk menguji kebenaran hasil peramalan dibandingkan dengan data PAD tahun 2004.

4.2.4 Proses Peramalan

Proses Peramalan adalah suatu proses untuk meramalkan data PAD yang akan datang dimana untuk *output* data aktualnya belum diketahui, dengan

menggunakan parameter fungsi keanggotaan yang telah diperbarui pada proses belajar. Pada proses Peramalan hanya digunakan tahap maju dari algoritma *Neuro-Fuzzy* seperti pada proses sebelumnya, perbedaannya hanya terletak pada data yang digunakan.

4.3 Pengujian Sistem

Selanjutnya untuk melihat apakah sistem yang telah dibangun sudah sesuai dengan yang diharapkan, maka perlu dilakukan pengujian yang mewakili sistem secara keseluruhan.

4.3.1 Form Menu Utama

Form Menu Utama digunakan untuk mengakses form-form yang ada pada aplikasi Peramalan PAD.



Gambar 4.1 Form Menu Utama

Form Menu Utama ini terdiri dari beberapa menu yaitu Master, Proses, Laporan, Help, dan Exit. Setiap menu terdiri dari beberapa sub menu, untuk lebih jelasnya dapat dilihat gambar sebagai berikut:

1. Menu Master



Gambar 4.2 Menu Master

Menu Master ini digunakan untuk menampilkan form Data PAD

2. Menu Proses



Gambar 4.3 Menu Proses

Pada menu Proses terdiri dari 4 (empat) sub menu yaitu sub menu Belajar, Validasi, Uji Coba, dan Peramalan. Sub menu Belajar digunakan untuk menampilkan form Belajar. Sub menu Validasi digunakan untuk menampilkan form Validasi. Pada sub menu Uji Coba terdiri dari 2 (dua) sub-sub menu yaitu sub-sub menu Uji Coba 1 dan Uji Coba 2, sub-sub menu Uji Coba 1 digunakan untuk menampilkan form Uji Coba 1 dan sub-sub menu Uji Coba 2 digunakan untuk menampilkan form Uji Coba 2. Sub menu Peramalan digunakan untuk menampilkan form Peramalan.

3. Menu Laporan



Gambar 4.4 Menu Laporan

Pada menu Laporan ini terdiri dari 4 (empat) sub menu yaitu sub menu Data PAD, Belajar, Validasi dan Uji Coba, dan Peramalan. Sub menu Data PAD digunakan untuk menampilkan laporan Data PAD. Sub menu Belajar digunakan untuk menampilkan laporan Belajar. Sub menu validasi dan Uji Coba digunakan untuk menampilkan laporan Validasi dan Uji Coba. Sub menu Peramalan digunakan untuk menampilkan laporan Peramalan.

4. Menu Help

A small screenshot showing a single menu item labeled 'Help' in a light-colored box.

Gambar 4.5 Menu Help

Menu Help digunakan untuk menampilkan file bantuan.

5. Menu Exit

A small screenshot showing a single menu item labeled 'Exit' in a light-colored box.

Gambar 4.6 Menu Exit

Menu Exit digunakan untuk keluar dari form Menu Utama.

4.3.2 Form Data

Form Data digunakan untuk meng-*input*-kan dan meng-*update* data PAD.

Langkah-langkah penggunaan form Data adalah sebagai berikut :

1. Untuk penambahan data, tekan tombol Add kemudian *input*-kan data PAD yang ingin di-*input*-kan. Lalu tekan tombol Save untuk menyimpan hasil *input*-an.
2. Untuk pencarian data, *input*-kan tahun dan pilih bulan yang ingin dicari, kemudian tekan tombol Find.
3. Untuk perubahan data, *click* 2 (dua) kali pada grid data yang ingin diubah, kemudian ubah data PAD yang ingin diubah dan tekan tombol Edit untuk menyimpan data PAD yang telah diubah.

KOD PAD	TAHUN	BULAN	PAD
PAD000001	2000	JANUARI	973.775.586,107
PAD000002	2000	FEBRUARI	987.624.574,101
PAD000003	2000	MARET	2.760.040.437,112
PAD000004	2000	APRIL	2.525.020.431,204
PAD000005	2000	MEI	2.039.025.092,062
PAD000006	2000	JUNI	1.874.611.241,150
PAD000007	2000	JULI	1.509.239.137,076
PAD000008	2000	AGUSTUS	1.699.769.000,029
PAD000009	2000	SEPTEMBER	1.866.400.916,151
PAD000010	2000	OKTOBER	1.420.796.351,149
PAD000011	2000	NOVEMBER	1.901.144.970,120
PAD000012	2000	DESEMBER	1.022.348.437,156

Gambar 4.7 Form Data

Pada form Data ini terdapat 6 (enam) tombol, yaitu:

1. Save : Untuk menyimpan data PAD.
2. Edit : Untuk melakukan perubahan data.
3. Add : Untuk melakukan penambahan data.

4. Cancel : Untuk melakukan pembatalan *input-an*/untuk mengosongkan form Data.
5. Find : Untuk mencari data PAD berdasarkan tahun dan bulan.
6. Exit : Untuk keluar dari form Data

4.3.3 Form Belajar

Form proses Belajar digunakan untuk proses belajar yang akan menghasilkan parameter-parameter premis yang baru melalui proses Neuro-Fuzzy Mod_ANFIS dengan pemilihan *error* yang minimum dan sejumlah iterasi yang dilakukan.

Pasangan data latih yang digunakan untuk proses belajar adalah data PAD tahun 2000 dan tahun 2001 sebagai *input* dan sedangkan tahun 2002 digunakan sebagai target yang diinginkan atau sebagai *output*.

Langkah-langkah penggunaan form proses Belajar adalah sebagai berikut:

1. Meng-*input*-kan jumlah iterasi yang diinginkan.
2. Untuk parameter premis dapat diambil dari sistem dengan menekan tombol Parameter.
3. Selanjutnya dilakukan proses belajar dengan menekan tombol Proses.

The screenshot shows the 'Form Belajar' application interface. It includes a title bar, a header with the application name and logo, and several functional panels. The 'Parameter Awal' and 'Parameter Akhir' panels contain input fields for numerical values. The 'Data Belajar' panel displays a table with columns for 'Bulan', 'Input 1 PAD 2000', 'Input 2 PAD 2001', and 'Output PAD 2002'. The 'Hasil Belajar' panel shows a table with columns for 'Kode Belajar', 'Output Data Aktual', and 'Output'. Below this, there are sections for 'MAPE' and 'MAPE Terkecil' with numerical values. The bottom of the window features a toolbar with buttons for 'Save', 'Matrik', 'Grafik MF', 'Grafik Output', and 'Exit'.

Gambar 4.8 Form Proses Belajar

Pada form proses Belajar ini terdapat 8 (delapan) tombol, yaitu :

1. Parameter : Untuk mengambil nilai parameter premis dari sistem
2. Proses : Untuk melakukan proses belajar
3. Clear : Untuk melakukan pembatalan/untuk mengosongkan form proses Belajar
4. Save : Untuk menyimpan hasil proses belajar.
5. Matrik : Untuk melihat hasil perhitungan LSE proses belajar.
6. Grafik MF : Untuk melihat grafik keanggotaan (*membership function*).
7. Grafik Output : Untuk melihat grafik output data aktual dan output

sistem serta selisih

8. Exit : Untuk keluar dari form proses Belajar.

A Form Matrik

Form Matrik digunakan untuk melihat hasil perhitungan LSE dari proses belajar.

The screenshot shows the 'Form Matrik' application. It features a blue header with the application name and a logo. The main area is divided into several sections:

- Matrik A:** A table of numerical values.
- Matrik B:** Another table of numerical values.
- Hasil - Invers(AT * A) A:** A list of values labeled P1, Q1, R1, P2, Q2, R2.
- Invers (AT * A):** A table of numerical values.
- Iterasi Ke : 69**
- Exit** button at the bottom right.

Gambar 4.9 Form Matrik

B. Form Grafik Fungsi Keanggotaan

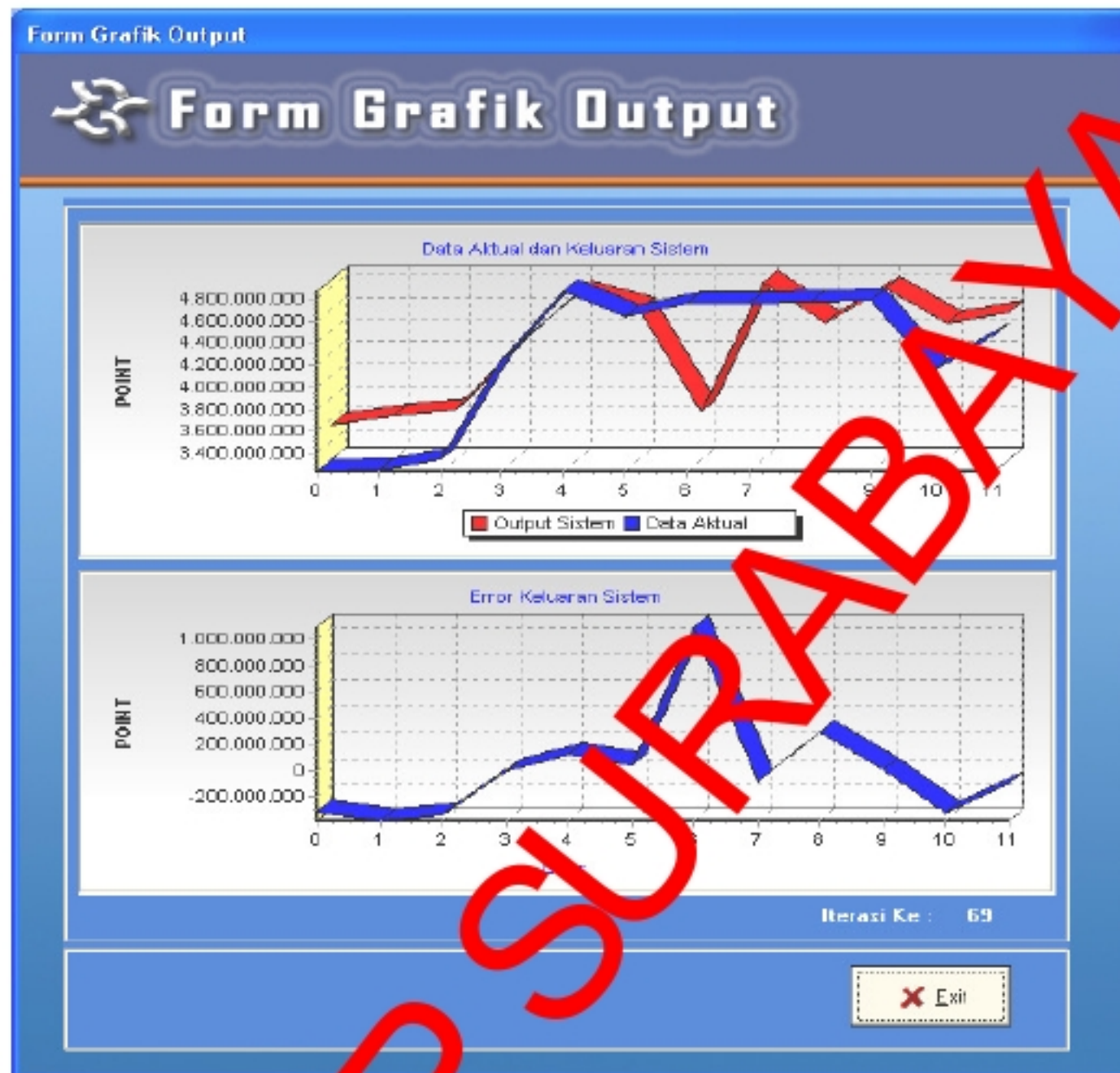
Form Grafik Fungsi Keanggotaan digunakan untuk menampilkan grafik fungsi keanggotaan awal dan fungsi keanggotan hasil belajar.



Gambar 4.10 Form Grafik Membership Function

C. Form Grafik Output dan Error

Form Grafik Output dan Error digunakan untuk menampilkan data aktual, keluaran sistem hasil belajar dan *error*.

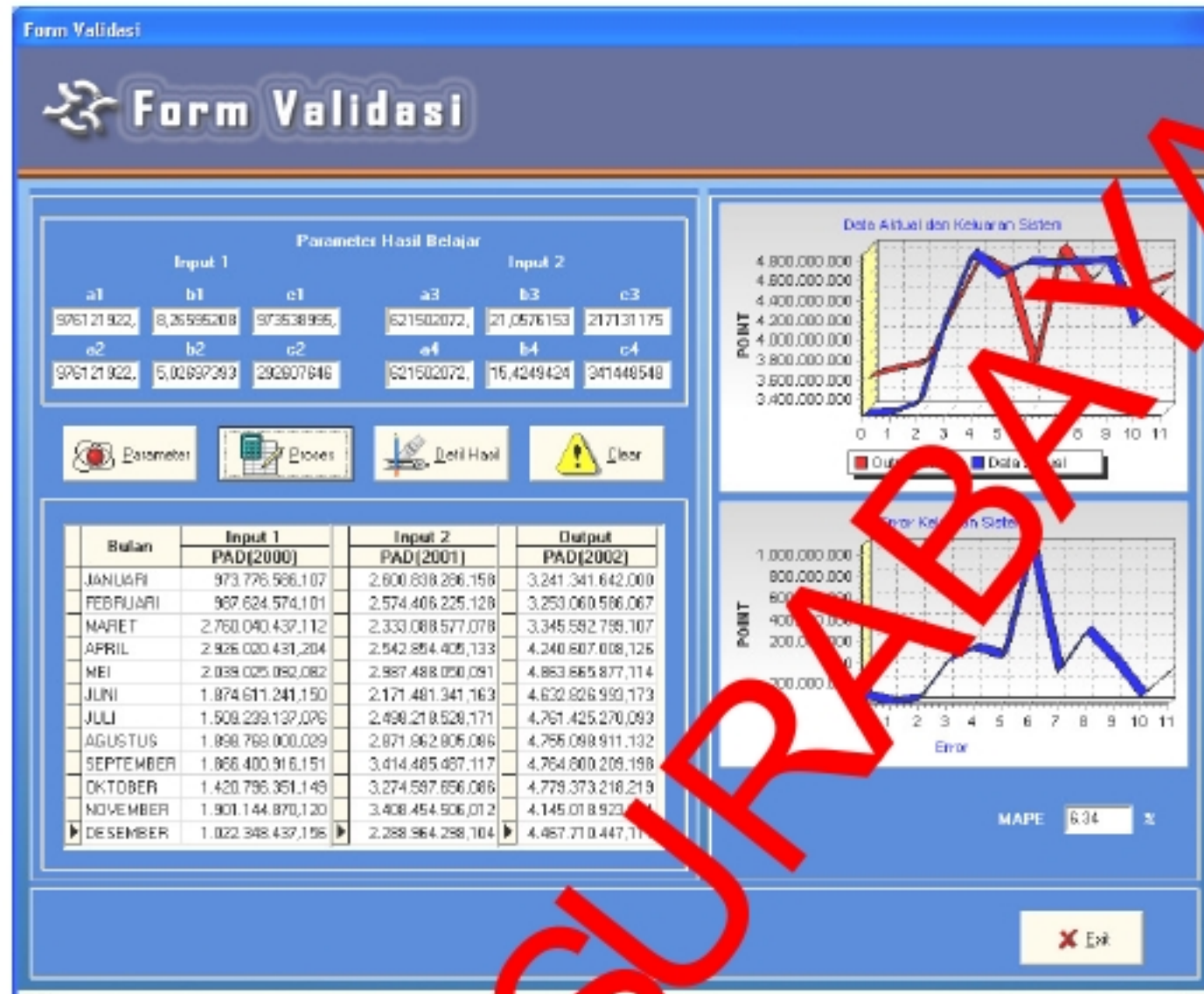


Form 4.1 Form Grafik Output dan Error

4.3.4 Form Validasi

Form Validasi digunakan untuk mengevaluasi hasil proses belajar yang telah dilakukan. Dengan menggunakan data yang sama seperti proses Belajar dan menggunakan parameter premis hasil proses Belajar. Langkah-langkah penggunaan form Validasi adalah sebagai berikut:

1. Tekan tombol Parameter untuk mengambil nilai parameter premis dari *database* Belajar.
2. Tekan tombol Proses untuk melakukan proses Validasi.
3. Untuk melihat hasil Validasi secara detil, tekan tombol Detil Hasil.



Gambar 4.12 Form Validasi

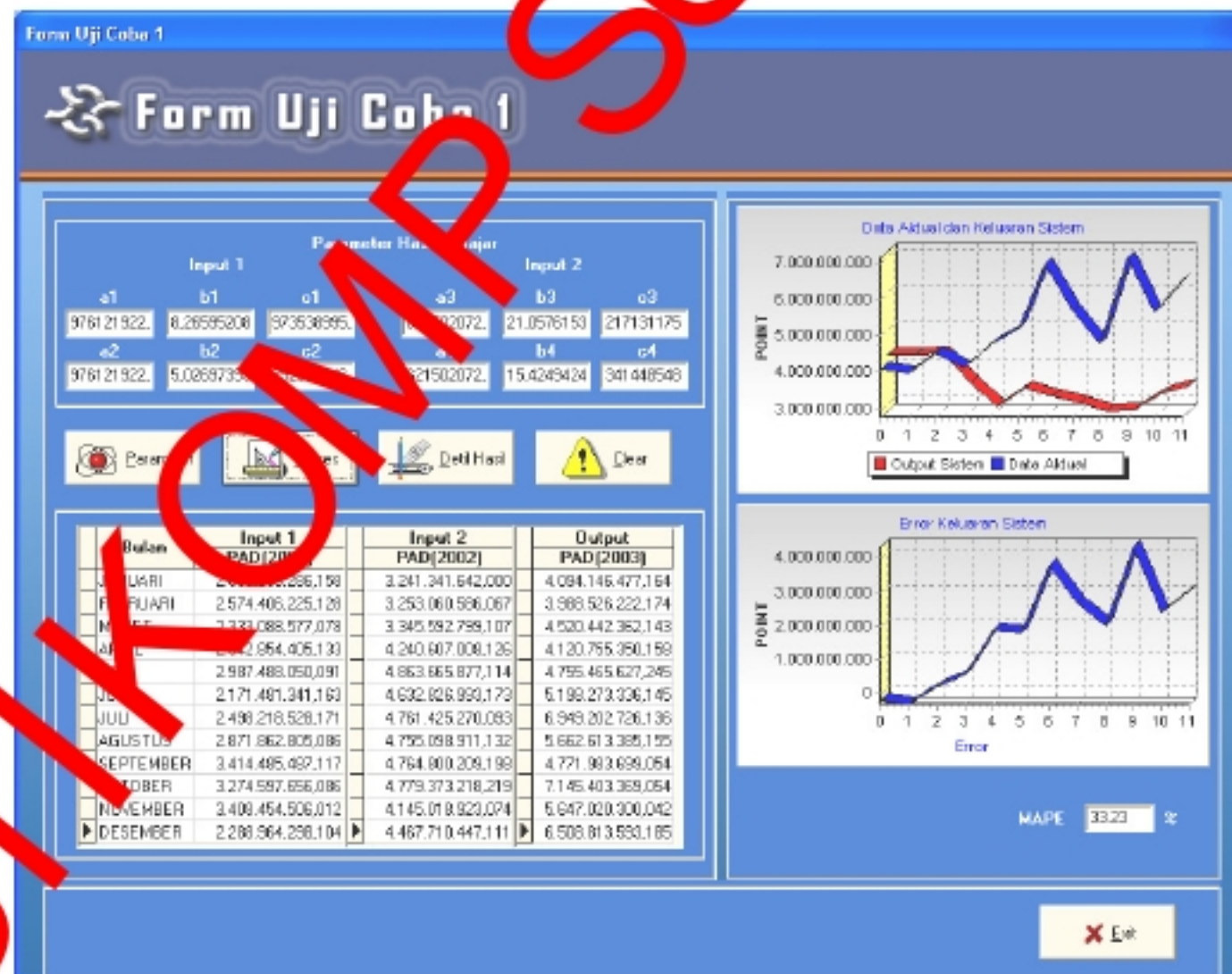
Pada form Validasi terdapat 5 (lima) tombol, yaitu:

1. **Parameter** : Untuk mengambil nilai parameter premis dari *database* Belajar.
2. **Proses** : Untuk melakukan proses Validasi.
3. **Detail Hasil** : Untuk melihat hasil dari proses Validasi secara detail.
4. **Clear** : Untuk membatalkan *input-an*/untuk mengosongkan form Validasi.
5. **Exit** : Untuk keluar dari form Validasi.

4.3.5 Form Uji Coba 1

Form Uji Coba 1 digunakan untuk melakukan proses Uji Coba 1 dengan menggunakan parameter premis hasil belajar. Pada proses Uji Coba 1 sistem diberikan masukan data tahun 2001 dan tahun 2002. Keluaran jaringan adalah hasil peramalan untuk tahun 2003. Selanjutnya untuk menguji kebenaran hasil peramalan dibandingkan dengan data PAD tahun 2003 yang sudah disiapkan. Langkah-langkah penggunaan form Uji Coba 1 adalah sebagai berikut:

1. Tekan tombol Parameter untuk mengambil nilai parameter premis dari *database* Belajar.
2. Tekan tombol Proses untuk melakukan proses Uji Coba 1.
3. Untuk melihat hasil Uji Coba 1 secara detail tekan tombol Detil Hasil.



Gambar 4.13 Form Uji Coba 1

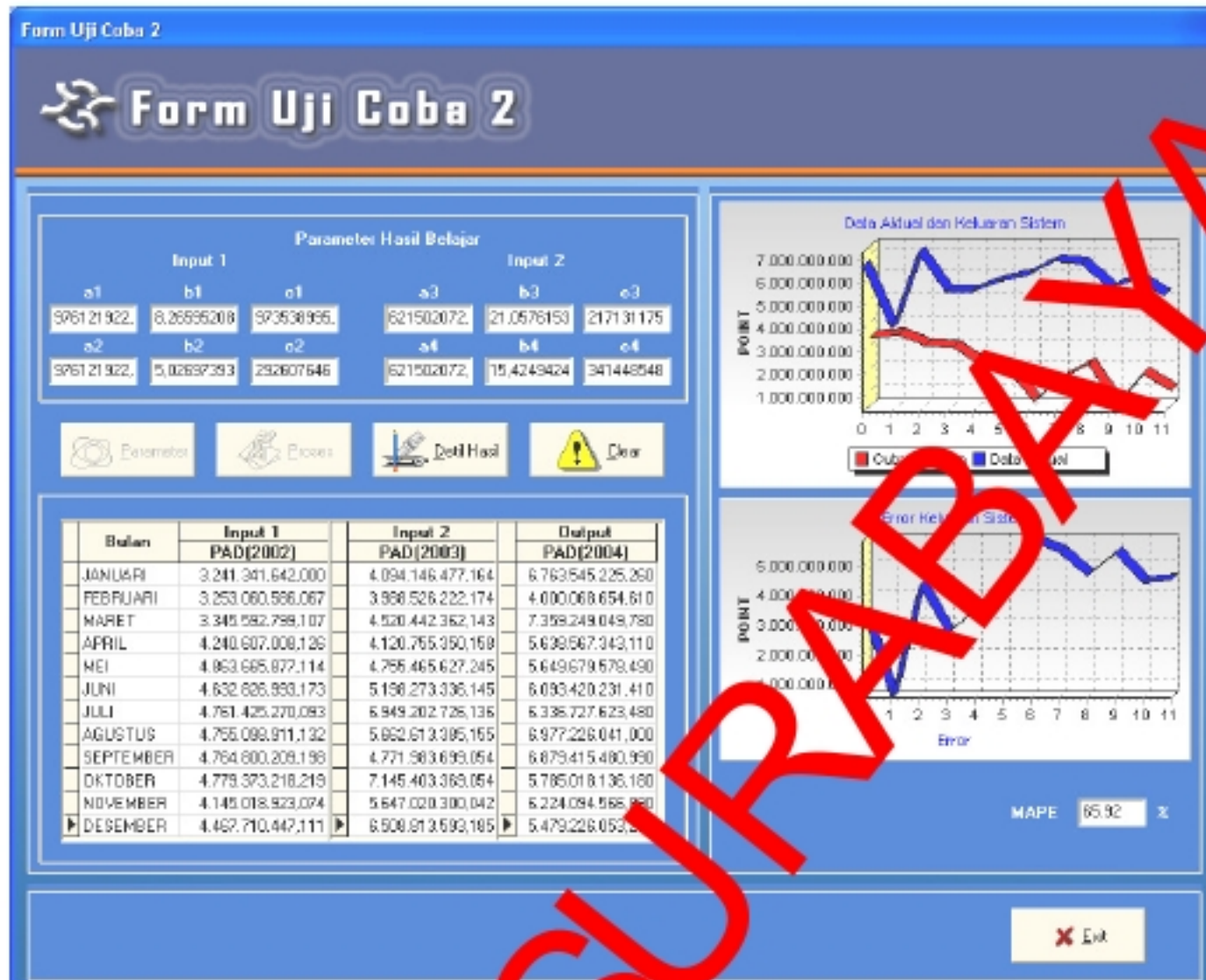
Pada form Uji Coba 1 ini terdapat 5 (lima) tombol, yaitu:

1. Parameter : Untuk mengambil nilai parameter premis dari *database* Belajar.
2. Proses : Untuk melakukan proses Uji Coba 1.
3. Detil Hasil : Untuk melihat hasil dari proses Uji Coba 1 secara detil.
4. Clear : Untuk membatalkan *input*-an/untuk mengosongkan form Uji Coba 1.
5. Exit : Untuk keluar dari form Uji Coba 1.

4.3.6 Form Uji Coba 2

Form Uji Coba 2 digunakan untuk melakukan proses Uji Coba 2 dengan menggunakan parameter premis hasil belajar. Pada proses Uji Coba 2, sistem diberikan masukan data tahun 2002 dan tahun 2003. Keluaran jaringan adalah hasil peramalan untuk tahun 2004. Selanjutnya untuk menguji kebenaran hasil peramalan dibandingkan dengan data PAD tahun 2004 yang sudah disiapkan. Langkah-langkah penggunaan form Uji Coba 2 adalah sebagai berikut:

1. Tekan tombol Parameter untuk mengambil nilai parameter premis dari *database* Belajar.
2. Tekan tombol Proses untuk melakukan proses Uji Coba 2.
3. Untuk melihat hasil Uji Coba 2 secara detil, tekan tombol Detil Hasil.



Gambar 4.14 Form Uji Coba 2

Pada form Uji Coba 2 ini terdapat 5 (lima) tombol, yaitu:

1. **Parameter** : Untuk mengambil nilai parameter premis dari *database* Belajar.
2. **Proses** : Untuk melakukan proses Uji Coba 2.
3. **Detail Hasil** : Untuk melihat hasil dari proses Uji Coba 2 secara detail.
4. **Clear** : Untuk membatalkan *input-an*/untuk mengosongkan form Uji Coba 2.
5. **Exit** : Untuk keluar dari form Uji Coba 2.

4.3.7 Form Peramalan

Form Peramalan digunakan untuk meramalkan data PAD yang akan datang dimana untuk *output* data aktualnya belum diketahui dengan menggunakan parameter premis hasil belajar. Langkah-langkah penggunaan form Peramalan adalah sebagai berikut:

1. Memilih tahun yang ada pada *combo box* sebagai *input* pertama. Kemudian tahun *input*-an kedua dan nilai parameter premis akan otomatis muncul.
2. Tekan tombol Proses untuk melakukan proses peramalan.

Gambar 4.15 Form Peramalan

Pada form Peramalan ini terdapat 3 (tiga) tombol, yaitu:

1. Proses : Untuk melakukan proses Peramalan.
2. Cancel : Untuk membatalkan *input-an*/mengosongkan form Peramalan.
3. Exit : Untuk keluar dari form Peramalan.

4.3.8 Laporan Data

Digunakan untuk menampilkan data PAD secara keseluruhan atau per tahun.



Gambar 4.16 Form Cetak Laporan Data PAD

Print Preview

Dipetak Pada Tanggal : 11/07/2006

LAPORAN DATA PENDAPATAN ASLI DAERAH
KABUPATEN GRESIK

Tahun : 2000

Kode PAD	Bulan	Jumlah PAD
PAD00001	JANUARI	973.774.896,107
PAD00002	FEBRUARI	987.424.874,101
PAD00003	MARET	2.788.048.437,112
PAD00004	APRIL	2.926.828.431,20
PAD00005	MAY	2.839.825.891,112
PAD00006	JUNI	1.874.611.241,112
PAD00007	JULI	1.589.239.137,876
PAD00008	AGUSTUS	1.589.239.137,876
PAD00009	SEPTEMBER	1.629.825.891,112
PAD00010	OKTOBER	1.629.825.891,112
PAD00011	NOVEMBER	1.589.239.137,876
PAD00012	DESEMBER	1.629.825.891,112

Page 1 of 1

Gambar 4.17 Laporan Data PAD

4.3.9 Laporan Belajar

Digunakan untuk menampilkan hasil dari proses Belajar.

Print Preview

Dipetak Pada Tanggal : 11/07/2006

LAPORAN BELAJAR PENDAPATAN ASLI DAERAH
KABUPATEN GRESIK

Kode Belajar	Output Data Aktual	Output Sistem	Selisih
MOD200201	3.241.341.642,000	3.555.551.323,869	-314.209.681,869
MOD200202	3.253.868.586,867	3.436.478.178,448	-382.609.591,573
MOD200203	3.345.592.799,187	3.686.422.870,310	-340.830.071,123
MOD200204	4.248.487.898,126	4.249.211.974,366	-864.966,240
MOD200205	4.863.665.877,114	4.737.782.369,697	125.983.507,417
MOD200206	4.632.826.893,173	4.598.396.141,728	42.438.851,445
MOD200207	4.761.425.278,893	3.667.726.690,929	1.093.698.579,164
MOD200208	4.755.898.911,132	4.841.541.482,271	-85.442.491,139
MOD200209	4.764.888.288,188	4.477.898.884,425	286.889.224,873
MOD200210	4.779.373.218,238	4.777.535.742,198	1.837.476,029
MOD200211	4.145.818.823,874	4.467.871.840,416	-322.853.017,342
MOD200212	4.467.718.447,111	4.562.892.265,372	-94.381.818,261

Page 1 of 1

Gambar 4.18 Laporan Belajar

4.3.10 Laporan Validasi dan Uji Coba

Digunakan untuk menampilkan hasil dari proses Validasi, Uji Coba 1, dan Uji Coba 2 secara keseluruhan ataupun per proses.

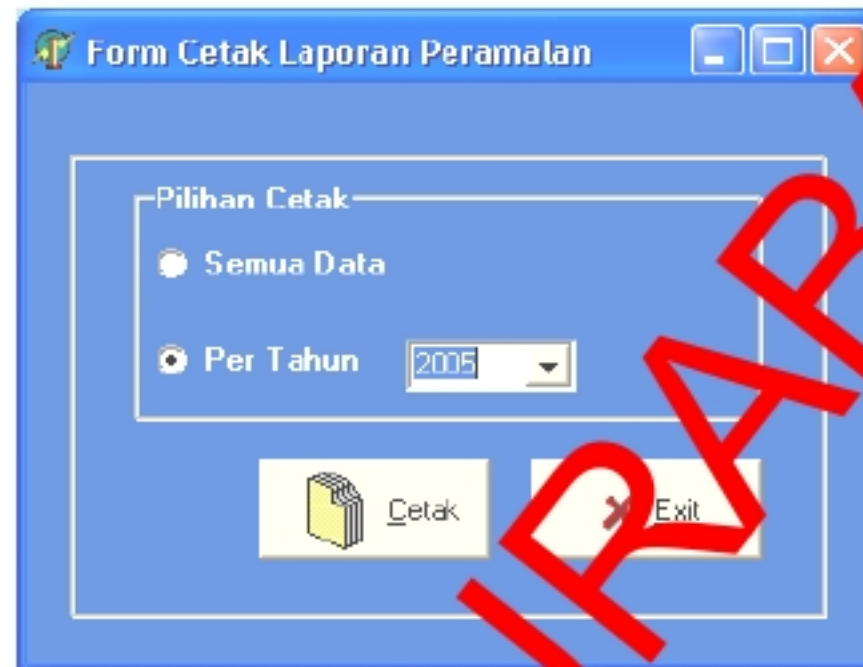
Gambar 4.19 Form Cetak Laporan Validasi dan Uji Coba

Kode Uji	Output Data Aktual	Output Sistem	Selisih
UJ1200301	4.894.146.477,164	4.299.584.459,446	-205.437.982,282
UJ1200302	3.988.526.222,174	4.301.719.943,763	-313.193.721,589
UJ1200303	4.528.442.362,143	4.332.597.924,655	187.934.437,488
UJ1200304	4.128.755.358,158	3.543.988.653,371	576.794.694,787
UJ1200305	4.755.465.627,245	2.867.581.804,797	1.887.963.822,448
UJ1200306	5.198.273.336,145	3.396.243.433,877	1.802.029.902,268
UJ1200307	6.949.282.726,136	3.156.284.241,737	3.792.918.484,399
UJ1200308	5.462.413.385,155	3.881.644.523,524	2.660.948.861,631
UJ1200309	4.771.983.699,854	2.762.367.646,642	2.009.616.052,412
UJ1200310	7.145.483.369,854	2.818.724.368,638	4.334.679.288,424
UJ1200311	5.647.828.308,842	3.248.966.787,118	2.398.853.512,932
UJ1200312	6.588.813.693,185	3.475.822.962,748	3.033.798.630,437

Gambar 4.20 Laporan Uji Coba 1

4.3.11 Laporan Peramalan

Digunakan untuk menampilkan hasil dari proses peramalan secara keseluruhan atau per tahun.



Gambar 4.21 Form Cetak Laporan Peramalan

Dicetak Pada Tanggal : 11/07/2005

**LAPORAN PERAMALAN PENDAPATAN ASLI DAERAH
KABUPATEN GRESIK**

Tahun	Kode Re...	Output Data Aktual	Output Sistem	Selisih
2005	RAM200501	,000	936.006.127,242	,000
2005	RAM200502	,000	3.138.728.679,505	,000
2005	RAM200503	,000	291.435.828,920	,000
2005	RAM200504	,000	1.803.466.503,214	,000
2005	RAM200505	,000	1.527.606.717,267	,000
2005	RAM200506	,000	994.661.716,908	,000
2005	RAM200507	,000	67.591.352,871	,000
2005	RAM200508	,000	100.900.854,594	,000
2005	RAM200509	,000	560.279.262,499	,000
2005	RAM200510	,000	415.862.699,404	,000
2005	RAM200511	,000	703.709.487,065	,000
2005	RAM200512	,000	922.647.499,701	,000

Page 1 of 1

Gambar 4.22 Laporan Peramalan

4.4 Hasil Peramalan

Setelah proses belajar, proses validasi dan uji coba selesai dilakukan maka hasil perhitungan peramalan secara keseluruhan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Kesalahan Peramalan Sistem

Tahun	MAPE
2002	6,34 %
2003	33,23 %
2004	65,92 %

Dari hasil peramalan yang dilakukan sistem yang terdapat pada Tabel 4.1 dapat dilihat untuk tahun 2003 mengalami kenaikan yang besar dibandingkan tahun 2002, hal ini dapat dilihat pada perhitungan MAPE sebesar 33,23 %. Demikian pula untuk tahun 2004 juga mengalami kenaikan yang besar, hal ini dapat dilihat pada perhitungan MAPE sebesar 65,92 %.

4.5 Evaluasi Sistem

Setelah dilakukan pengujian sistem, kemudian dilakukan evaluasi sistem. Tahap ini mengevaluasi hasil pengujian sistem dengan cara membandingkannya dengan hasil peramalan yang telah dilakukan oleh Dinas Pendapatan Daerah (DISPENDA) kabupaten Gresik dan mengukur kesalahan peramalan dengan ukuran ketepatan metode peramalan.

4.5.1 Membandingkan Hasil Peramalan

Untuk membuktikan apakah hasil peramalan dengan menggunakan NeuroFuzzy Mod_ANFIS mempunyai nilai *error* yang lebih baik, maka hasil peramalan dibandingkan dengan hasil peramalan yang telah dilakukan oleh DISPENDA. Hasil perbandingan dapat dilihat pada Tabel 4.2.

Tabel 4.2. Perbandingan hasil peramalan

Metode	Tahun 2002	Tahun 2003	Tahun 2004
	MAPE	MAPE	MAPE
A	6,34 %	33,23 %	65,92 %
B	45,99 %	54,27 %	67,33 %

Keterangan:

1. Metode A = Neuro-Fuzzy Mod_ANFIS.
2. Metode B = Peramalan oleh DISPENDA.

Berdasarkan Tabel 4.2 dapat dilihat bahwa hasil peramalan sistem memiliki jumlah kesalahan yang lebih kecil daripada hasil peramalan yang telah dilakukan DISPENDA. Maka dapat disimpulkan bahwa hasil peramalan menggunakan Neuro-Fuzzy Mod_ANFIS menghasilkan kesalahan lebih baik, hal ini dibuktikan dengan *error* yang lebih kecil.