BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada Tugas Akhir ini akan dirancang sebuah Perangkat Lunak untuk *forecasting* produksi gula berdasarkan hasil laporan tahunan PTPN IX (PERSERO) / PG.Pangka dengan parameter – parameter yang telah ditentukan dengan metode Jaringan Saraf Tiruan (JST) untuk mendapatkan model JST yang paling optimal dengan menggunakan algoritma *Backpropagation*. Bab ini akan membahas analisis dan perancangan perangkat lunak yang dibuat.

3.1 Analisis Sistem

Sebelum memulai pelatihan, terlebih dahulu ditentukan arsitektur dan parameter jaringan, serta menormalisasi input. Ada sembilan faktor yang diambil dari data jumlah produksi gula. Maka jaringan memiliki sembilan neuron input. Sedangkan *output* yang diharapkan adalah prediksi jumlah produksi gula yang dilambangkan dengan satu *output* (jika *output* yang diinginkan sesuai dengan yang diharapkan berarti prediksi tersebut sudah benar). Sedangkan untuk parameter jaringan, dalam pelatihan akan dicoba dilakukan beberapa perubahan parameter untuk melihat parameter manakah yang dapat menghasilkan sistem jaringan yang terbaik. Parameter yang akan diubah adalah learning rate (α) dan epoch.

3.1.1 Analisis Masalah

Permasalahan yang akan dibahas dalam tugas akhir ini adalah membuat suatu sistem yang dapat memprediksi jumlah produksi gula berdasarkan data yang dilatihkan yaitu data yang diambil dari PTPN IX (PERSERO) / PG.Pangka. Penerapan jaringan saraf tiruan dalam permasalahan *forecasting* jumlah produksi gula meliputi penentuan komponen-komponen jaringan saraf tiruan yang digunakan dan mengaplikasikannya dalam permasalahannya. Untuk memprediksi jumlah produksi gula ini diterapkan algoritma *backpropagation*.

3.1.2 Analisis prosedur yang sedang berjalan

Analisis prosedur atau proses sistem memberikan gambaran tentang sistem yang saat ini sedang berjalan. Analisis sistem bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui, berikut akan terlihat lebih jelasnya dalam *flow map* dibawah ini

3.1.2.1 *Flow Map* sistem yang sedang berjalan

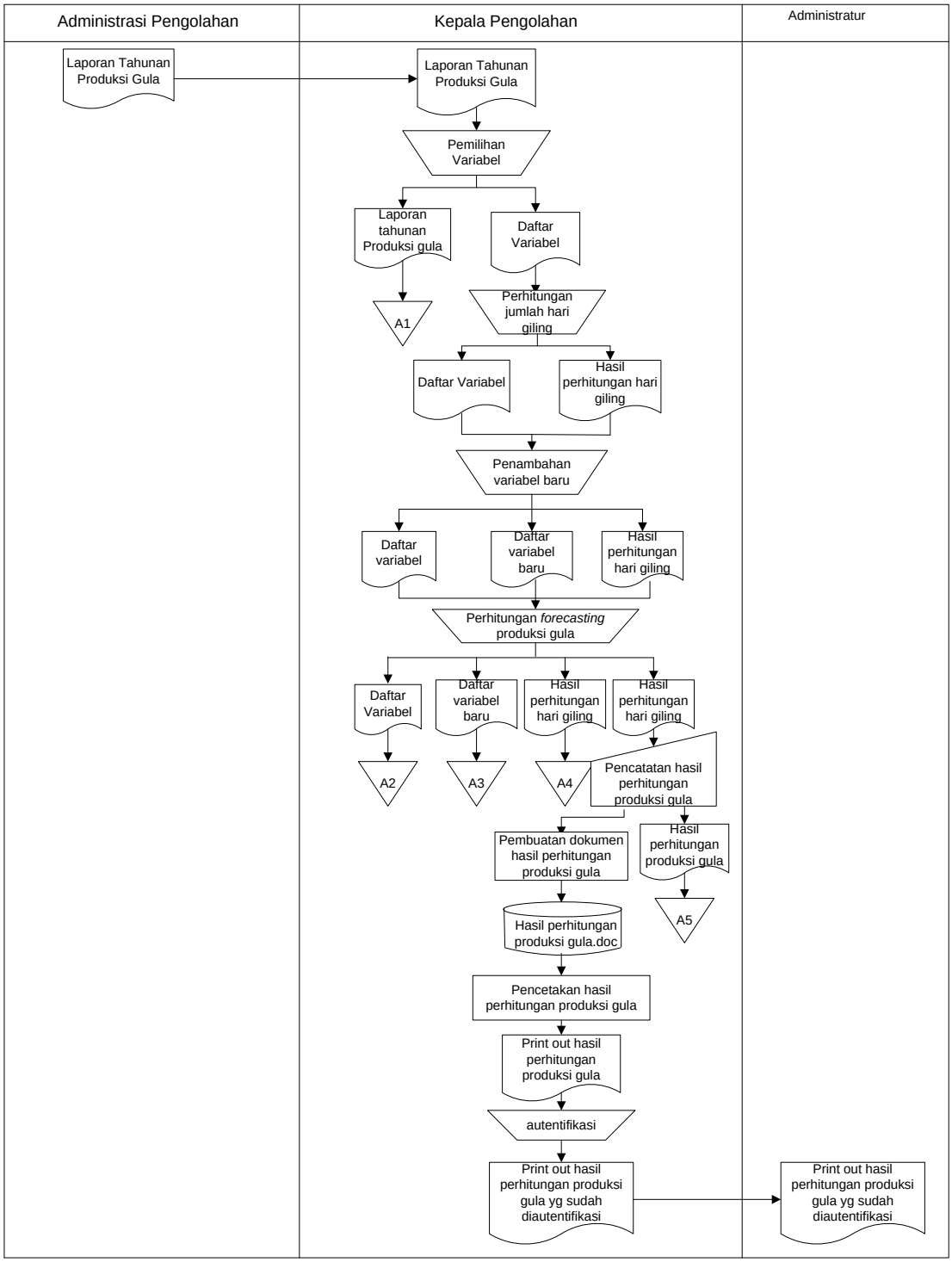
Bagan alir dokumen atau bagan alir formulir merupakan bagan alir yang menunjukan arus dari dokumen. Terdapat beberapa prosedur yang sedang berjalan di PTPN IX (PERSERO) / PG PANGKA. Prosedur tersebut merupakan aturan-aturan yang digunakan dalam pembuatan *forecasting* produksi gula.

Berikut merupakan prosedur pembuatan *forecasting* produksi gula di PTPN IX (PERSERO) / PG PANGKA.

1. Administrasi pengolahan menyerahkan laporan tahunan produksi gula kepada kepala pengolahan.

2. Setelah laporan tahunan sampai ke kepala pengolahan, maka kepala pengolahan melakukan pemilihan variabel untuk melakukan *forecasting* produksi gula untuk tahun berikutnya.
3. Kepala Pengolahan melakukan perhitungan untuk mendapatkan perkiraan jumlah produksi gula. Perhitungan pertama yaitu mencari jumlah hari giling dengan menggunakan variabel yang sudah di tentukan sebelumnya.
4. Setelah didapatkan jumlah hari giling, selanjutnya kepala pengolahan menambahkan variabel baru untuk perhitungan *forecasting* produksi gula.
5. Setelah proses perhitungan forecasting produksi gula selesai dilakukan maka kepala pengolahan melakukan proses pencatatan perhitungan *forecasting* ke dalam dokumen word.
6. Kepala pengolahan melakukan proses autentifikasi terhadap dokumen tersebut.
7. Kepala pengolahan menyerahkan dokumen hasil forecasting produksi gula kepada administratur.

Alur di atas dapat digambarkan dalam *flow map* berikut:



Gambar 3.1 *Flow Map* sistem yang sedang berjalan

A1 : Laporan tahunan produksi gula

A2 : Daftar variabel

A3 : Daftar variable baru

A4 : Hasil perhitungan hari giling

A5 : Hasil perhitungan produksi gula

3.1.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui kebutuhan non-fungsional. Spesifikasi kebutuhan non fungsional merinci tentang hal-hal yang dilakukan saat pengimplementasian.. Analisis ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran, volume data yang akan ditangani sistem, jumlah pemakai serta kontrol terhadap sistem.

3.1.3.1 Analisis Perangkat Keras

Adapun spesifikasi perangkat keras yang dibutuhkan untuk sistem yang akan dibangun sebagai berikut :

1. *Processor* Intel Pentium 4
2. RAM 256 MB
3. *Harddisk space* 512 MB
4. Keyboard dan Mouse

3.1.3.2 Analisis Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dapat digunakan selama pengembangan sistem adalah sbb:

1. Setara dengan Microsoft Win 98, 2000, Me, NT 4.0
2. Bahasa Pemrograman yang digunakan adalah Delphi 7.0

3.1.3.3 Analisis *User* (pengguna) Sistem

Setelah melakukan pengumpulan data dengan teknik wawancara dan observasi maka terdapat 2 user yang menggunakan aplikasi ini yaitu:

1. Kepala Pengolahan

   Minimal menguasai komputer dalam menginputkan data dan menggunakan aplikasi office.

2. Administratur

   Minimal menguasai komputer dalam menginputkan data dan menggunakan aplikasi office.

3.2 Analisis Jaringan Saraf Tiruan

Untuk membangun JST *Backpropagation* maka diperlukan langkah-langkah sbb:

1. Menentukan masukan

   Jumlah masukan ditentukan berdasarkan data produksi gula dari tahun-tahun sebelumnya yaitu tahun 1980-2009 yang akan diinputkan ke dalam sistem.

2. Menentukan Transformasi linear data produksi gula

   Data ditransformasi linear pada selang [0.1,0.9]. Oleh karena itu keluaran yang dihasilkan jaringan akan berada pada nilai 0 sampai 1. Untuk mendapatkan nilai sebenarnya dari keluaran perlu dilakukan proses detransformasi linear.

3. Menentukan pembagian data

   Langkah selanjutnya setelah transformasi data adalah pembagian data. Data dibagi menjadi data pelatihan, pengujian dan validasi. Adapun komposisi data

pelatihan, pengujian dan validasi yang digunakan adalah 80% untuk data pelatihan dan 20% untuk data pengujian dan validasi.

4. Menentukan jumlah neuron pada layer tersembunyi

   Banyaknya jumlah neuron yang digunakan pada lapisan tersembunyi akan menentukan dalam kinerja sistem ketika memprediksi jumlah produksi. Hal ini disebabkan karena semakin banyaknya neuron yang digunakan dalam lapisan tersembunyi, maka akan membuat setiap neuron dalam lapisan tersembunyi memiliki beban yang semakin sedikit dalam menampung data-data yang harus dipelajari. Untuk lapisan tersembunyi, ditentukan secara *trial* dan *error* dan di dapatkan hasil yang terbaik dengan menggunakan sembilan neuron.

5. Menentukan parameter pembelajaran, yaitu maksimum epoch, besar galat, dan learning rate

   Parameter-parameter ini akan berpengaruh terhadap kinerja sistem dalam proses pembelajaran terhadap proses prediksi. Untuk maksimum epoch dilakukan pengujian epoch sampai mendapatkan nilai epoch terbaik yaitu 1000, dan untuk galat digunakan nilai 0.001 sebagai kriteria pemberhentian JST.

6. Menentukan fungsi bobot yang akan digunakan berdasarkan fungsi aktifasi.

   Penentuan nilai bobot minimum dan maksimum ditentukan atas dasar $0<n<1$ yang merupakan jangkauan dari fungsi. Fungsi ini digunakan untuk jaringan saraf tiruan yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner mempunyai nilai antara 0 sampai 1. Oleh karena itu,

fungsi ini sering digunakan untuk JST yang membutuhkan nilai output yang terletak pada interval 0 sampai 1.

Fungsi sigmoid biner (log sigmoid) digunakan sebagai fungsi aktivasi pada simpul tersembunyi dan keluaran.

7. Menentukan keluaran

   Jumlah *output* ditentukan berdasarkan hasil yang diinginkan.

3.2.1 Jaringan Saraf Tiruan sebagai Peramal

JST mempunyai kemampuan untuk belajar sehingga dapat membangun pengertian tentang hubungan antar data masukan dan keluaran. Jaringan saraf tiruan akan melakukan pemetaan terurut sehingga dapat diperoleh pola lengkapnya. Dengan kemampuannya mengidentifikasi dan melengkapi pola inilah jaringan saraf tiruan akan dimanfaatkan sebagai peramal.

3.2.2 Jaringan Saraf Tiruan *Backpropagation*

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Pelatihan sebuah jaringan yang menggunakan backpropagation terdiri dari tiga langkah, yaitu pelatihan pola input secara *feedforward*, perhitungan *backpropagation* dari kumpulan kesalahan dan penyesuaian bobot. Sesudah pelatihan, aplikasi dari jaringan hanya terdiri dari fase *feedforward*. Bahkan jika pelatihan menjadi lambat, sebuah jaringan yang dilatih dapat menghasilkan *output*nya sendiri secara cepat.

3.2.3 Peramalan Jumlah Produksi Gula dengan JST menggunakan *Algoritma Backpropagation*

Pola produksi gula setiap tahunnya akan banyak berubah. Pola produksi akan berulang pada setiap tahun. Pengulangan ini juga akan terjadi pada pola kurva dari tahun ke tahun.

Pengalaman bagian pendataan di PTPN IX menunjukkan bahwa jumlah produksi gula pada setiap tahunnya dipengaruhi oleh jumlah produksi gula pada tahun sebelumnya.

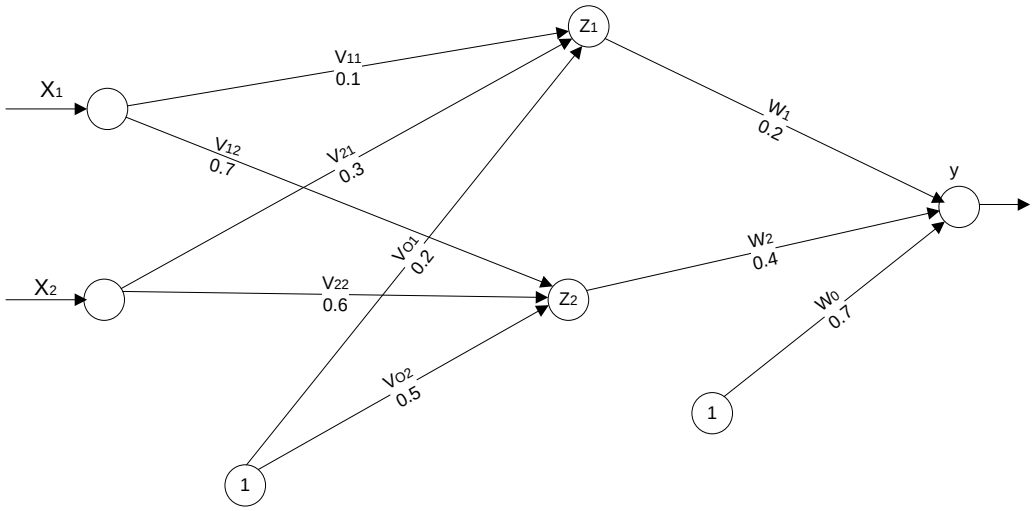
Berikut ini merupakan sebagian contoh untuk mengimplementasikan algoritma yang digunakan.

Contoh :

Akan dibentuk jaringan saraf untuk memprediksi jumlah produksi gula. Ada 2 input (X_1 , dan X_2) dengan 1 output (target). Inputan tersebut merupakan data produksi pada tahun 2006, 2007 dan 2008 yang meliputi luas lahan dan hasil taksasi tebu seperti tabel berikut ini :

Tahun	Luas Lahan (x_1)	hasil taksasi tebu (x_2)	Target
2006	3171.86	236091.9	18286.946
2007	3638.28	232708.2	16858.568
2008	3525.91	228316.5	18649.292

Jika digambarkan ke dalam arsitektur jaringan saraf *backpropagation* sebagai berikut:



Diket :

Data tahun 2006

Pola input : X_1= 3171.86

input : X_2= 236091.9

Target : 18286.946

Data tahun 2007

Pola input : X_1= 3638.28

input : X_2= 232708.2

Target : 16858.568

Data tahun 2008

Pola input : X_1= 3525.91

input : X_2= 228316.5

Target : 18649.292

Konstanta Belajar (α) : 0.1

Fungsi Aktivasi : $\frac{1}{1+e^{-x}}$

Bobot antara lapisan masukan dengan lapisan tersembunyi (V_{ij})

V_{11}= 0.1	V_{12}= 0.7
V_{21}= 0.3	V_{22}= 0.6

Bobot awal bias ke hidden

V_{01}= 0.2

V_{02}= 0.5

Bobot awal hidden ke output

W_1= 0.2

W_2 = 0.4

Bobot awal bias ke output

W_0 = 0.7

Ditanya : Hitunglah jumlah produksi gula untuk tahun 2009

Jawab :

>> **Transformasi Data :**

2006 → X_1 = 3171.86, x'= $\frac{(X-Xmin)(b-a)}{(Xmax-Xmin)} + a$

= $\frac{(3171.86-3171.86)(0.9-0.1)}{(3638.28-3171.86)} + 0.1$

= 0.1

X_2 = 236091.9, x'= $\frac{(236091.9-228316.5)(0.9-0.1)}{(236091.9-228316.5)} + 0.1$

= $\frac{7775.4(0.8)}{7775.4} + 0.1$

= $\frac{6220.32}{7775.4} + 0.1$

= 0.8+0.1

= 0.9

$$t = 18286.946,\ x' = \frac{(18286.946-16858.568)(0.9-0.1)}{(18649.292-16858.568)} + 0.1$$

$$= \frac{1428.378(0.8)}{1790.724} + 0.1$$

$$= \frac{1142.7024}{1790.724} + 0.1$$

$$= 0.638+0.1$$

$$= 0.738$$

$$2007 \rightarrow X_1 = 3638.28,\ x' = \frac{(3638.28-3171.86)(0.9-0.1)}{(3638.28-3171.86)} + 0.1$$

$$= \frac{466.42(0.8)}{466.42} + 0.1$$

$$= \frac{373.136}{466.42} + 0.1$$

$$= 0.8+0.1$$

$$= 0.9$$

$$X_2 = 232708.2,\ x' = \frac{(232708.2-228316.5)(0.9-0.1)}{(236091.9-228316.5)} + 0.1$$

$$= \frac{4391.7(0.8)}{7775.4} + 0.1$$

$$= \frac{3513.36}{7775.4} + 0.1$$

$$= 0.452+0.1$$

$$= 0.552$$

$$t = 16858.568,\ x' = \frac{(16858.568-16858.568)(0.9-0.1)}{(18649.292-16858.568)} + 0.1$$

$$= 0.1$$

$$2008 \rightarrow X_1 = 3525.91,\ x' = \frac{(3525.91-3171.86)(0.9-0.1)}{(3638.28-3171.86)} + 0.1$$

$$= \frac{354.05(0.8)}{466.42} + 0.1$$

$$= \frac{283.24}{466.42} + 0.1$$

$$= 0.607+0.1$$

$$= 0.707$$

$X_2 = 228316.5$, $x' = \frac{(228316.5-228316.5)(0.9-0.1)}{(236091.9-228316.5)} + 0.1$

$$= 0.1$$

t = 18649.292, $x' = \frac{(18649.292-16858.568)(0.9-0.1)}{(18649.292-16858.568)} + 0.1$

$$= \frac{1790.724(0.8)}{1790.724} + 0.1$$

$$= \frac{1432.5792}{1790.724} + 0.1$$

$$= 0.8+0.1$$

$$= 0.9$$

>>Data hasil transformasi

Tahun	Luas Lahan (x_1)	hasil taksasi tebu (x_2)	Target (t)
2006	0.1	0.9	0.738
2007	0.9	0.552	0.1
2008	0.707	0.1	0.9

- **Pelatihan Epoh ke-1**

  ⇨ **Data ke-1**

  Feedforward Propagation

  Operasi pada Hidden Layer →

  >> Penjumlahan terbobot :

$Z_in_1 = Vo_1 + V_{11} * X_1 + V_{21} * X_2$

$= (0.2) + (0.1*0.1) + (0.3*0.9)$

$= 0.48$

$Z_in_2 = Vo_2 + V_{12} * X_1 + V_{22} * X_2$

$= 0.5 + (0.7*0.1) + (0.6*0.9)$

$= 1.11$

>> Pengaktifan :

$Z_1 = \frac{1}{1+e^{-Zin1}}$

$= \frac{1}{1+e^{-0.48}} = 0.6177$

$Z_2 = \frac{1}{1+e^{-Zin2}}$

$= \frac{1}{1+e^{-1.11}} = 0.752$

Operasi pada *Output Layer* →

>> Perkalian :

$y_in = w_0 + w_1*z_1 + w_2*z_2$

$= 0.7 + 0.2 * 0.6177 + 0.4*0.752$

$= 0.7+0.12354+0.3008$

$= 1.12434$

>> Pengaktifan :

$y = \frac{1}{1+e^{-1.12434}} = 0.7547928542$

Error = target – y= 0.738- 0.754= -0.016

Jumlah kuadrat error = $(-0.016)^2 = 0.000256$

δ = (t-y) (y)(1-y)

= (-0.016) (0.754)(1-0.754)

= -0.0029

>> Koreksi bobot

$\Delta w_1 = \alpha * \delta * z_1$

$= 0.1 *(-0.0029) *0.6177= -0.000179$

$\Delta w_2 = \alpha * \delta * z_2$

$= 0.1 *(-0.0029) *0.752= -0.000218$

$\Delta w_0 = \alpha * \delta = 0.1*(-0.0029) = -0.00029$

>> Perbaharui bobot dan prasikap

$\delta_{in1} = \delta * w_1$

$= -0.0029 * 0.2 = -0.00058$

$\delta_{in2} = \delta * w_2$

$= -0.0029 * 0.4 = -0.00116$

$\delta_1 = \delta_{in1} * \left[\frac{1}{1+e^{-z1}}\right] * \left[1 - (\frac{1}{1+e^{-z1}})\right]$

$= -0.00058* \left(\frac{1}{1+e^{-0.6177}}\right) * \left[1 - (\frac{1}{1+e^{-0.6177}})\right]$

$= -0.00058* 0.6497*0.35= -0.00013$

$\delta_2 = \delta_{in2} * \left[\frac{1}{1+e^{-z2}}\right] * \left[1 - (\frac{1}{1+e^{-z2}})\right]$

$= -0.00116* \left(\frac{1}{1+e^{-0.752}}\right) * \left[1 - (\frac{1}{1+e^{-0.752}})\right]$

$= -0.00116*0.6796*0.32 = -0.00025$

>> Tiap unit keluaran memperbaharui bobot

$\Delta V_{11} = \alpha * \delta_1 * x_1 = 0.1 * -0.00013 * 0.1 = -0.0000013$

$\Delta V_{21} = \alpha * \delta_1 * x_2 = 0.1 * -0.00013 * 0.9 = -0.0000117$

$\Delta V_{12} = \alpha * \delta_2 * x_1 = 0.1 * -0.00025 * 0.1 = -0.0000025$

$\Delta V_{22} = \alpha * \delta_2 * x_2 = 0.1 * -0.00025 * 0.9 = -0.0000225$

$\Delta V_{01} = \alpha * \delta_1 = 0.1 * -0.00013 = -0.000013$

$\Delta V_{02} = \alpha * \delta_2 = 0.1 * -0.00025 = -0.000025$

>> Hitung bobot awal input ke hidden

$V_{11} = V_{11} + \Delta V_{11} = 0.1 + (-0.0000013) = 0.0999$

$V_{12} = V_{12} + \Delta V_{12} = 0.7 + (-0.0000025) = 0.6999$

$V_{21} = V_{21} + \Delta V_{21} = 0.3 + (-0.0000117) = 0.2999$

$V_{22} = V_{22} + \Delta V_{22} = 0.6 + (-0.0000225) = 0.5999$

$V_{01} = V_{01} + \Delta V_{01} = 0.2 + (-0.000013) = 0.1999$

$V_{02} = V_{02} + \Delta V_{02} = 0.5 + (-0.000025) = 0.4999$

$w_1 = w_1 + \Delta w_1 = 0.2 + (-0.000179) = 0.1998$

$w_2 = w_2 + \Delta w_2 = 0.4 + (-0.000218) = 0.3997$

$w_0 = w_0 + \Delta w_0 = 0.7 + (-0.00029) = 0.6997$

➔Data ke-2

Feedforward Propagation

Operasi pada Hidden Layer →

>> Penjumlahan terbobot :

$Z_in_1 = Vo_1 + V_{11} * X_1 + V_{21} * X_2$

$= (0.1999) + (0.0999 * 0.9) + (0.2999 * 0.552)$

$= 0.455$

$Z_in_2 = Vo_2 + V_{12} * X_1 + V_{22} * X_2$

$= 0.4999 + (0.6999*0.9) + (0.5999*0.552)$

$= 1.46$

>> Pengaktifan :

$Z_1 = \frac{1}{1+e^{-Zin1}}$

$= \frac{1}{1+e^{-0.455}} = 0.612$

$Z_2 = \frac{1}{1+e^{-Zin2}}$

$= \frac{1}{1+e^{-1.46}} = 0.812$

Operasi pada Output Layer →

>> Perkalian :

$y_in = w_0 + w_1 * z_1 + w_2 * z_2$

$= 0.6997 + 0.1998 * 0.612 + 0.3997 * 0.812$

$= 0.6997 + 0.1222776 + 0.3245564$

$= 1.1465$

>> Pengaktifan :

$y = \frac{1}{1+e^{-1.1465}} = 0.76$

Error = target – y= 0.1- 0.76= -0.66

Jumlah kuadrat error = $(-0.66)^2 = 0.4356$

$\delta = (t-y)\ (y)(1-y)$

$= (-0.66)\ (0.76)(1-0.76)$

$= -0.12$

>> Koreksi bobot

$\Delta w_1 = \alpha * \delta * z_1$

$= 0.1 *(-0.12) *0.612= -0.0073$

$\Delta w_2 = \alpha * \delta * z_2$

$= 0.1 *(-0.12) *0.812= -0.0097$

$\Delta w_0 = \alpha * \delta = 0.1*(-0.12) = -0.012$

>> Perbaharui bobot dan prasikap

$\delta_{in1} = \delta * w_1$

$= -0.12 * 0.1998= -0.0239$

$\delta_{in2} = \delta * w_2$

$= -0.12 * 0.3997= -0.0479$

$\delta_1 = \delta_{in1} * \left[\frac{1}{1+e^{-z1}}\right] * \left[1 - (\frac{1}{1+e^{-z1}})\right]$

$= -0.0239* \left(\frac{1}{1+e^{-0.612}}\right) * \left[1 - (\frac{1}{1+e^{-0.612}})\right]$

$= -0.0239* 0.6484*0.3516= -0.005448$

$\delta_2 = \delta_{in2} * \left[\frac{1}{1+e^{-z2}}\right] * \left[1 - (\frac{1}{1+e^{-z2}})\right]$

$= -0.0479* \left(\frac{1}{1+e^{-0.812}}\right) * \left[1 - (\frac{1}{1+e^{-0.812}})\right]$

$= -0.0479*0.6925*0.307 = -0.01018$

>> Tiap unit keluaran memperbaharui bobot

$\Delta V_{11} = \alpha*\delta_1*x_1 = 0.1*-0.005448*0.9 = -0.00049$

$\Delta V_{21} = \alpha*\delta_1*x_2 = 0.1*-0.005448*0.552= -0.0003$

$\Delta V_{12} = \alpha * \delta_2 * x_1 = 0.1 * -0.01018 * 0.9 = -0.000916$

$\Delta V_{22} = \alpha * \delta_2 * x_2 = 0.1 * -0.01018 * 0.552 = -0.00056$

$\Delta V_{01} = \alpha * \delta_1 = 0.1 * -0.005448 = -0.0005448$

$\Delta V_{02} = \alpha * \delta_2 = 0.1 * -0.01018 = -0.001018$

>> Hitung bobot awal input ke hidden

$V_{11} = V_{11} + \Delta V_{11} = 0.0999 + (-0.00049) = 0.0994$

$V_{12} = V_{12} + \Delta V_{12} = 0.6999 + (-0.000916) = 0.6989$

$V_{21} = V_{21} + \Delta V_{21} = 0.2999 + (-0.0003) = 0.2996$

$V_{22} = V_{22} + \Delta V_{22} = 0.5999 + (-0.00056) = 0.5993$

$V_{01} = V_{01} + \Delta V_{01} = 0.1999 + (-0.000013) = 0.1998$

$V_{02} = V_{02} + \Delta V_{02} = 0.4999 + (-0.000025) = 0.4998$

$w_1 = w_1 + \Delta w_1 = 0.1998 + (-0.0073) = 0.1925$

$w_2 = w_2 + \Delta w_2 = 0.3997 + (-0.0097) = 0.39$

$w_0 = w_0 + \Delta w_0 = 0.6997 + (-0.012) = 0.6877$

➔Data ke-3

Feedforward Propagation

Operasi pada Hidden Layer →

>> Penjumlahan terbobot :

$Z_in_1 = Vo_1 + V_{11} * X_1 + V_{21} * X_2$

$= (0.1998) + (0.0994 * 0.707) + (0.2996 * 0.1) = 0.3$

$Z_in_2 = Vo_2 + V_{12} * X_1 + V_{22} * X_2$

$= 0.4998 + (0.6989 * 0.707) + (0.5993 * 0.1) = 1.05$

>> Pengaktifan :

$Z_1 = \frac{1}{1+e^{-Zin1}}$

$= \frac{1}{1+e^{-0.3}} = 0.574$

$Z_2 = \frac{1}{1+e^{-Zin2}}$

$= \frac{1}{1+e^{-1.05}} = 0.74$

Operasi pada *Output Layer* →

>> Perkalian :

$y_in = w_0 + w_1 * z_1 + w_2 * z_2$

$= 0.6877 + (0.1925 * 0.574) + (0.39 * 0.74)$

$= 1.087$

>> Pengaktifan :

$y = \frac{1}{1+e^{-1.087}} = 0.748$

Error = target – y= 0.9- 0.748= 0.152

Jumlah kuadrat error = $(0.152)^2 = 0.0231$

$\delta = (t-y)\ (y)(1-y)$

$= (0.152)\ (0.748)(1-0.748)$

$= 0.0286$

>> Koreksi bobot

$\Delta w_1 = \alpha * \delta * z_1$

$= 0.1 * (0.0286) * 0.574 = 0.00164$

$\Delta w_2 = \alpha * \delta * z_2$

$= 0.1 * (0.0286) * 0.74 = 0.002$

$\Delta w_0 = \alpha * \delta = 0.1*(0.0286) = 0.00286$

>> Perbaharui bobot dan prasikap

$\delta_{in1} = \delta * w_1$

$= 0.0286* 0.1925= 0.0055$

$\delta_{in2} = \delta * w_2$

$= 0.0286* 0.39= 0.0112$

$\delta_1 = \delta_{in1} * \left[\frac{1}{1+e^{-z1}}\right] * \left[1 - \left(\frac{1}{1+e^{-z1}}\right)\right]$

$= 0.0055* \left(\frac{1}{1+e^{-0.574}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.574}}\right)\right]$

$= 0.0055* 0.6396*0.36= 0.00126$

$\delta_2 = \delta_{in2} * \left[\frac{1}{1+e^{-z2}}\right] * \left[1 - \left(\frac{1}{1+e^{-z2}}\right)\right]$

$= 0.0112* \left(\frac{1}{1+e^{-0.74}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.74}}\right)\right]$

$= 0.0112*0.6769*0.323 = 0.002448$

>> Tiap unit keluaran memperbaharui bobot

$\Delta V_{11} = \alpha*\delta_1*x_1 = 0.1*0.00126*0.707= 0.000089$

$\Delta V_{21} = \alpha*\delta_1*x_2 = 0.1*0.00126*0.1= 0.0000126$

$\Delta V_{12} = \alpha*\delta_2*x_1 = 0.1* 0.002448*0.707= 0.00017$

$\Delta V_{22} = \alpha*\delta_2*x_2 = 0.1* 0.002448* 0.1= 0.00002448$

$\Delta V_{01} = \alpha*\delta_1 = 0.1*0.00126= 0.000126$

$\Delta V_{02} = \alpha*\delta_2 = 0.1* 0.002448= 0.0002448$

>> Hitung bobot awal input ke hidden

$V_{11}= V_{11} + \Delta V_{11} = 0.0994+ 0.000089= 0.099489$

$V_{12}= V_{12} + \Delta V_{12} = 0.6989+ 0.00017= 0.699$

$V_{21}= V_{21} + \Delta V_{21} = 0.2996+ 0.0000126= 0.2996126$

$V_{22}= V_{22} + \Delta V_{22} = 0.5993+ 0.00002448= 0.59932448$

$V_{01}= V_{01} + \Delta V_{01} = 0.1998+ 0.000126= 0.199926$

$V_{02}= V_{02} + \Delta V_{02} = 0.4998+ 0.0002448= 0.5$

$w_1= w_1+ \Delta w_1 = 0.1925+ 0.00164= 0.19414$

$w_2= w_2+ \Delta w_2 = 0.39+ 0.002= 0.392$

$w_0= w_0+ \Delta w_0 = 0.6877+ 0.00286= 0.69$

Proses ini dilakukan secara terus menerus, berulang sampai maksimum epoch yang ditentukan atau kuadrat error < target error.

- **Pengujian**

  Misalkan dilakukan pengujian terhadap data ke-3: x[0.707 0.1]

  Operasi pada *Hidden Layer* :

  >> Penjumlahan terbobot :

  $Z_in_1 = Vo_1 + V_{11} * X_1 + V_{21} * X_2$

  $= (0.199926) + (0.099489*0.707) + (0.2996126*0.1)$

  $= 0.3$

  $Z_in_2 = Vo_2 + V_{12} * X_1 + V_{22} * X_2$

  $= 0.5+ (0.699*0.707) + (0.59932448*0.1)$

  $= 1.054$

  >> Pengaktifan :

  $Z_1 = \frac{1}{1+e^{-Zin_1}}$

$$= \frac{1}{1+e^{-0.3}} = 0.574$$

$$Z_2 = \frac{1}{1+e^{-Zin2}}$$

$$= \frac{1}{1+e^{-1.054}} = 0.742$$

Operasi pada *Output Layer* :

>> Perkalian :

$y_in = w_0 + w_1 * z_1 + w_2 * z_2$

$= 0.69 + (0.19414 * 0.574) + (0.392 * 0.742)$

$= 1.0923$

>> Pengaktifan :

$$y = \frac{1}{1+e^{-1.0923}} = 0.7488$$

output = detransformasi y → $y-0.1 = \frac{(x-16858.568)(0.9-0.1)}{(18649.292-16858.568)}$

$$0.6488 = \frac{(x-16858.568)0.8}{1790.724}$$

$1161.821731 = (x-16858.568)0.8$

$1452.277164 = x-16858.568$

$18310.84516 = x$

Jadi, jumlah produksi gula untuk tahun 2009 = 18310.84516 ton.

3.3 Variabel - variabel Masukan dan Keluaran

Prediktor ini akan dipakai untuk memprediksi jumlah produksi gula untuk beberapa tahun ke depan. Jadi jaringan akan menghasilkan satu keluaran sehingga hanya diperlukan satu sel lapisan keluaran.

Secara umum data yang akan menjadi masukan dalam sistem peramalan adalah data yang historis. Data historis merupakan sekumpulan data per tahun jumlah produksi gula di PTPN IX (PERSERO) / PG.PANGKA. Data historis ini berarti data jumlah produksi pada interval waktu sebelumnya. Data historis digunakan sebagai variable masukan untuk menentukan atau meramalkan jumlah produksi gula untuk beberapa tahun ke depan, maka diambil data dari 30 tahun ke belakang. Data historis dijadikan sebagai data pelatihan dan data pengujian dengam komposisi 80% untuk data pelatihan dan 20% untuk data pengujian . Pada program ini menggunakan data pelatihan selama 30 tahun dengan variabel data yang meliputi :

1. Luas Areal Perkebunan
2. Hasil taksasi tebu (jumlah tebu yang dihasilkan).
3. Jumlah penduduk
4. Jumlah Kebutuhan gula selama 1 tahun
5. Jumlah penggunaan pupuk
6. Jam berhenti giling
7. Jumlah Hari giling
8. Kapasitas pabrik
9. Rendemen

Data yang dikumpulkan adalah data historis dari tahun 1980 sampai tahun 2009.

3.4 Metode *Multiple Regression*

Multiple regression adalah regresi dengan dua atau lebih variabel X_1, X_2, X_3, …., Xn sebagai variabel bebas dan Y sebagai variabel tak bebas, sehingga merupakan perluasan dari regresi linier sederhana. Model probabilistik regresi berganda yang melibatkan (k-1) variabel X adalah sebagai berikut:

$$Y = a + b_1 X_1 + b_2 X_2 + b_3 X_3 + \ldots\ldots b_k X_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (1)$$

Y : Variabel terikat

X_1, X_2, X_3,….,X_k = variable bebas

a_1, b_1, b_2,…. b_k = koefisien variabel

Pada persamaan linear lebih dari dua variable, variable Y dipengaruhi oleh lebih dari dua variable, yaitu variable X_1, X_2, ….. X_k. Dalam hal demikian, variable Y disebut variabel terikat (*dependent variable*) dan variable-variabel X_1, X_2,…. X_k disebut variable bebas (independent variable), artinya nilai-nilai variable Y dapat ditentukan berdasarkan nilai-nilai dari variable X_1,X_2,….X_k.

Syarat-syarat Regresi Linier Berganda, sebagai berikut :

a. Model regresi linier;
b. Eksistensi (X diasumsikan non stokastik);
c. Nilai rata-rata kesalahan adalah nol, atau Ė(μ/ Xi) = 0;
d. Homoskedastisitas, artinya varian kesalahan sama untuk setiap periode (homo = sama, skedastisitas = sebaran) dinyatakan dalam bentuk matematis:

   Var (μ/ Xi) = 0;
e. Tidak ada autokorelasi antar kesalahan (antara i dan j tidak ada korelasinya). Dinyatakan dalam bahasa matematis : Covarians (μi , μ j) = 0;

f. Antara μ dan X saling bebas, sehingga covarians (μ i, X) = 0;

g. Tidak ada multikolinieritas yang sempurna antar variabel bebas;

h. Jumlah observasi n harus lebih besar daripada jumlah parameter yang diestimasi (jumlah variabel bebas);

i. Adanya variabilitas dalam nilai X, artinya nilai X harus berbeda (tidak boleh sama semua);

j. Model regresi telah dispesifikasikan secara benar. Dengan kata lain tidak ada bias (kesalahan) spesifikasi dalam model yang digunakan dalam analisis empiris.

contoh:

Y	X_1	X_2	X_1Y	X_2Y	X_1X_2	X_1^2	X_2^2	Y^2
29	45	16	1305	464	720	2025	256	841
24	42	14	1008	336	588	1764	196	576
27	44	15	1188	405	660	1936	225	729
25	45	13	1125	325	585	2025	169	625
26	43	13	1118	338	559	1849	169	676
28	46	14	1288	392	644	2116	196	784
30	44	16	1320	480	704	1936	256	900
28	45	16	1260	448	720	2025	256	784
28	44	15	1232	420	660	1936	225	784
27	43	15	1161	405	645	1849	225	729
∑=272	∑=441	∑=147	∑=12005	∑=4013	∑=6485	∑=19461	∑=2173	∑=7428

$\bar{Y}$= 27.2 $\bar{X}_1$= 44.1 $\bar{X}_2$=14.7

1) Dalam 3 persamaan Normal :

$272 = 10\,a + 441\,b_1 + 147\,b_2$ ……. (1)

12005 = 441 a +19461 b_1 + 6485 b_2 ……. (2)

4013 = 147 a + 6485 b_1 + 2173 b_2 ……. (3)

2) Menghilangkan nilai a dengan menjumlahkan persamaan (1) dengan (2). Persamaan (1) x -441 dan persamaan (2) x 10 :

(1) x (-441) : -119952 = -4410 a – 194481 b_1 – 64827 b_2

(2) x (10) : 120050 = 4410 a + 194610 b_1 + 64850 b_2

(4) : 98 = 129 b_1 + 23 b_2

3) Persamaan (1) kalikan dengan -147 dan persamaan (3) dengan 10. Jumlahkan persamaan (1) dan (3) :

(1) x (-147) : -39984 = -1470 a – 64827 b_1 – 21609 b_2

(3) x (10) : 40130 = 1470 a + 64850 b_1 + 21730 b_2

(5) : 146 = 23 b_1 + 121 b_2

4) Kalikan persamaan (4) dengan -23 dan (5) dengan 129. Jumlahkan (4) dan (5) untuk menduga nilai b :

(4) x(-23): -2254 = -2967 b_1 – 592 b_2

(5) x(129): 18834 = 2967 b_1 + 15609b_2

(6) : 16580 = 15080 b_2 , maka, b_2 = 1.099

5) Cari nilai penduga b_1 dari persamaan (4) :

(4) : 98 = 129 b_1 + 23 b_2

98 = 129 b_1 + 23 (1.099)

98 = 129 b_1 + 25.277, maka, b_1 = 0.564

6) Cari nilai a dari persamaan :

$a = \overline{Y} - b_1\overline{X}_1 - b_2\overline{X}_2$

= 27.2 – (0.564) (44.1) – (1.099)(14.7)

= 27.2 – (24.8724) – (16.1553)

= -13.8277 ≈ 13.828

Maka, persamaan garis regresi berganda :

$\hat{Y} = a + b_1X_1 + b_2X_2$

= -13828 + 0.564 X_1 + 1099 X_2

Nilai Y dari persamaan di atas :

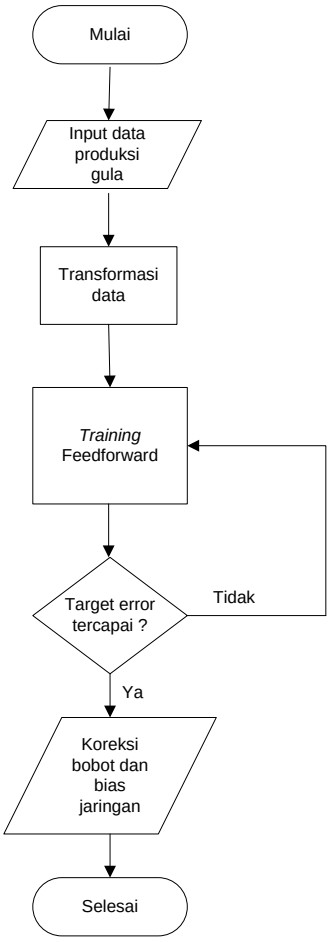
= -13828 + 0.564 (43) + 1099 (16)

= 28008

3.5 Perancangan Arsitektur Jaringan Saraf Tiruan

Arsitektur JST dengan metode *backpropagation* pada sistem yang akan dibangun adalah arsitektur JST berlapisan banyak yang terdiri dari lapisan masukan (layer input), satu lapisan tersembunyi (layer hidden), dan lapisan keluaran (layer output). Penghubung setiap lapisan adalah bobot. Nilai bobot dihitung dengan rumus : Bobot = {(random-0.5) x (Inisialisasi bobot x 2)}

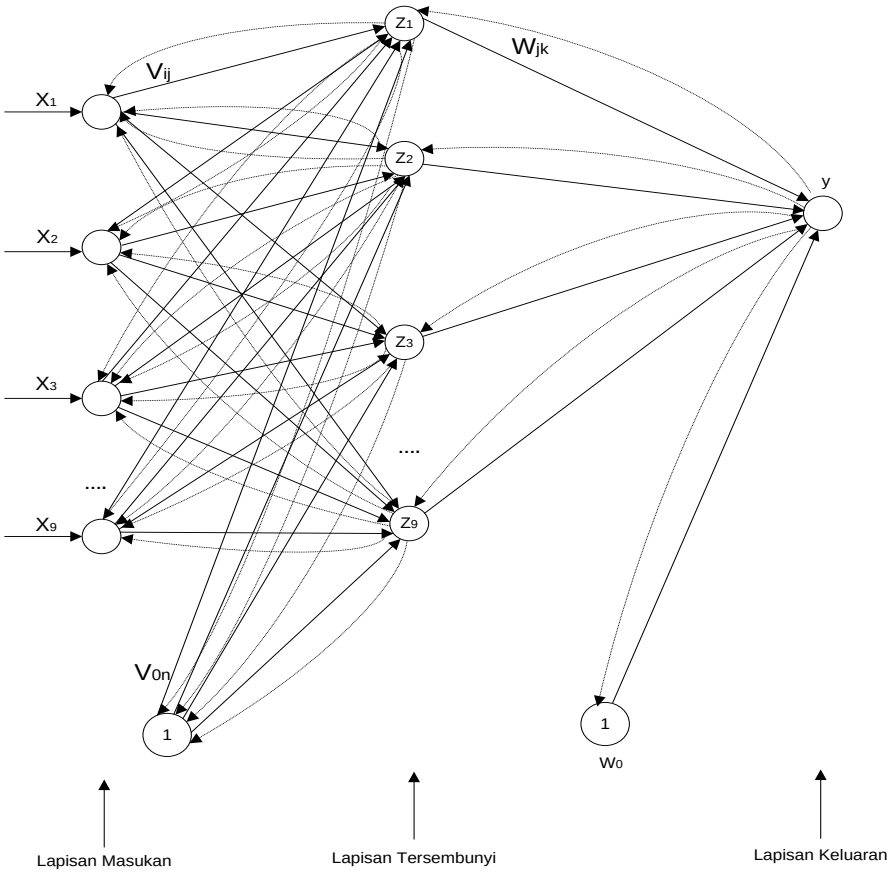
Sistem akan memberikan nilai random pada setiap bobot di lapisan masukan dan lapisan keluaran. Secara umum alur proses JST *backpropagation* dapat dilihat pada gambar 3.2 dibawah ini.



Gambar 3.2 Alur Proses JST *Backpropagation*

Gambar 3.2 menjelaskan bahwa data yang menjadi input jaringan akan mengalami proses pembelajaran. Proses pembelajaran sendiri diawali dari proses *feedforward*, yang kemudian dilanjutkan dengan proses backpropagation. Setelah proses *backpropagation*, akan dicek apakah target *error* teleh dicapai, jika target *error* telah dicapai, maka proses pmblajaran selesai, yang menghasilkan koreksi dari bobot jaringan. Jika tidak maka akan kembali ke proses *feedforward* sebagaimana dijelaskan di atas sampai mencapai nilai *epoch* maksimum.

Arsitektur jaringan *backpropagation* seperti terlihat dalam gambar 3.3 di bawah ini.



Gambar 3.3 Arsitektur Jaringan Backpropagation

Gambar 3.3 merupakan arsitektur jaringan syaraf tiruan Backpropagation dengan 3 lapisan (Layer), yaitu 1 lapisan input, 1 lapisan hidden, 1 lapisan output. Untuk lapisan hidden bisa lebih dari 1 lapisan, tidak ada aturan yang baku untuk menentukan banyaknya lapisan tersebut. Lapisan Input berfungsi untuk menerima sinyal input eksternal dan selanjutnya output dari lapisan input ini (setelah dimasukkan kedalam fungsi aktifasi) digunakan oleh lapisan hidden untuk diproses.

Unit input akan dilambangkan dengan x, hidden unit dilambangkan dengan z, dan unit *output* dilambangkan dengan y. Sedangkan untuk bobot antara x dan z dilambangkan dengan v dan bobot antara z dan y dilambangkan dengan w.

3.5.1 Input Layer

Input layer berfungsi untuk menerima nilai dari ciri pola input dimana nilai dari neuron pada input layer adalah numerik. Banyaknya neuron pada input layer dipengaruhi oleh banyaknya pengambilan pola input yang akan dimasukkan ke dalam jaringan saraf tiruan.

3.5.2 Hidden Layer

Untuk menentukan jumlah hidden layer tidak ada ketentuan yang pasti, namun waktu training dan akurasi berpengaruh pada banyaknya hidden layer yang digunakan.

3.5.3 Output Layer

Banyaknya layer yang digunakan adalah satu layer. Nilai neuron output merupakan bilangan sigmoid biner yaitu 0 atau 1. Output layer terdiri dari beberapa neuron output kombinasi dari semua neuron tersebut dapat digunakan untuk mengidentifikasi dan juga *learning* sebagai output yang seharusnya.

3.6 Perancangan Sistem

Perancangan sistem merupakan bagian dari metodologi pengembangan suatu software yang dilakukan setelah melalui tahapan analisis. Perancangan sistem yang merupakan tahapan lanjutan dari analisis sistem dimana pada

perancangan sistem digambarkan rancangan sistem yang akan dibangun sebelum melakukan pengkodean ke dalam suatu bahasa pemrograman.

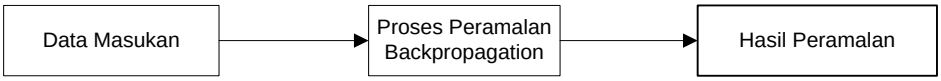
3.6.1 Perancangan Masukan

Pembuatan *database* dilakukan dengan menggunakan *Ms.Access*. Perancangan *database* dalam bahasa *Ms.Access* adalah sebagai berikut:

Nama Kolom	Ukuran kolom	Tipe Data	Keterangan
Tahun	4	text	-
Luas Lahan	double	number	-
Hasil Taksasi Tebu (ton)	double	number	-
Kapasitas Pabrik	Integer	number	-
Rendemen	double	number	-
Jumlah Penduduk	double	number	-
Jumlah kebutuhan gula	double	number	-
Jumlah pupuk	double	number	-
Jumlah hari giling	integer	number	-
Jam berhenti giling	double	number	-
Target	double	number	-

3.6.2 Perancangan Sistem Secara *Global*

Adapun perancangan sistem peramalan produksi gula secara umum dapat digambarkan seperti gambar 3.4 berikut ini :



Gambar 3.4 Perancangan Sistem Secara Global

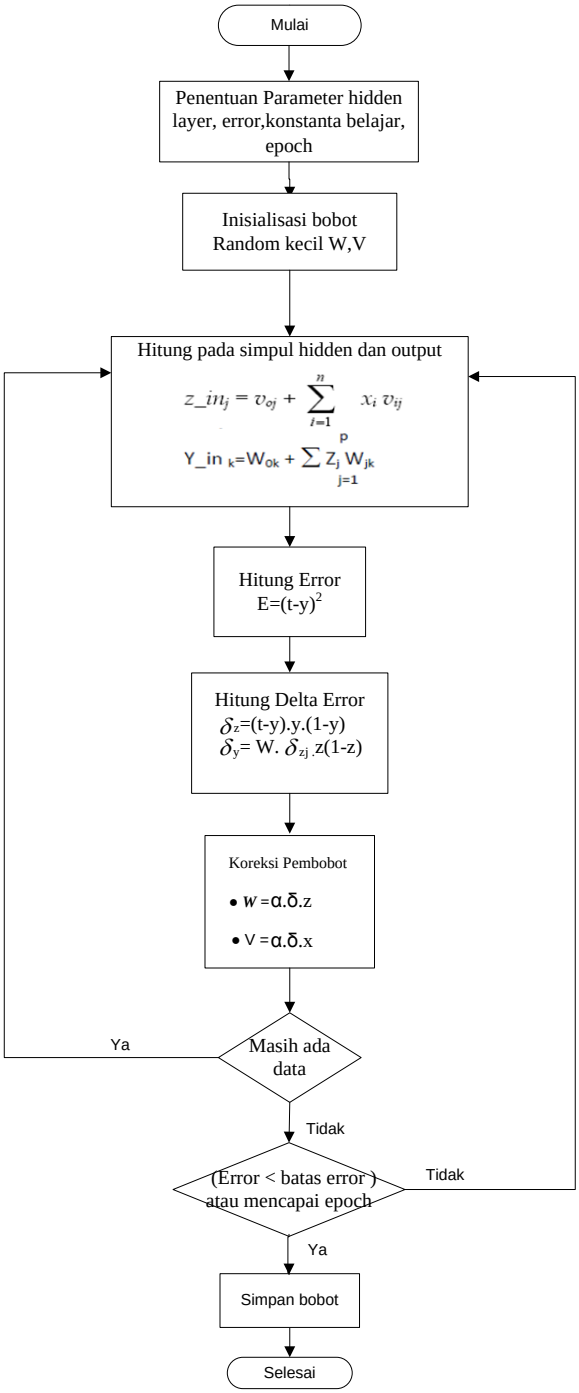
Gambar di atas menjelaskan bahwa data masukan yang terdiri dari berbagai macam variable akan diolah melalui proses peramalan backpropagation untuk memprediksi besarnya jumlah produksi gula. Setelah melalui proses peramalan maka diperoleh hasil peramalan yang bisa digunakan sebagai prediksi untuk tahun mendatang.

3.6.3 Perancangan Prosedural

Selanjutnya akan dijelaskan tentang perancangan prosedural, dimana secara umum terdapat dua prosedur , yaitu prosedur *training* (pelatihan) dan peramalan.

3.6.3.1 Prosedur *Training*

Prosedur *training* adalah prosedur untuk melakukan pembelajaran terhadap pola-pola yang akan dikenali. Proses ini dilakukan dengan menggunakan data *training*. Proses ini berhenti jika MSE lebih kecil dari *error* yang ditetapkan atau *epoch* yang telah ditentukan user telah tercapai sehingga didapatkan bobot-bobot neuron yang diharapkan. *Epoch* (iterasi maksimum) itu adalah satu set putaran vector-vektor pembelajaran. Beberapa epoch diperlukan untuk pembelajaran sebuah backpropagation sehingga kesalahan mendekati 0 (nol). Adapun gambaran alur proses *training* dijelaskan dalam gambar 3.5 berikut



Gambar 3.5 Diagram Alir Proses *Training* JST *Backpropagation*

Dari gambar di atas dapat dijelaskan bahwa pada tahap training data berupa variabel-variabel penentu jumlah produksi yang berbentuk angka, yang kemudian dimasukkan inisialisasi bobot oleh *user*, kemudian menentukan

parameter-parameter masukan yang ada seperti jumlah hidden layer, epoch, MSE, learning rate dan fungsi aktivasi yang digunakan yaitu *sigmoid biner*. Tahap selanjutnya adalah training, dan data hasil training berupa bobot.

Pada diagram alir di atas terdapat dua buah proses yaitu proses perhitungan algoritma backpropagation dan proses perhitungan bobot. Berikut adalah keterangan dari masing-masing proses.

Algorima *backpropagation* :

a. Inisialisasi bobot dan bias. Baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka disekitar 0 dan 1, atau –1(bias positif dan negatif).
b. Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai ***FALSE***.

1. Data dimasukkan ke input jaringan (*feedforward*)

a. Setiap unit input (Xi, i = 1, …,n) menerima sinyal *input* Xi dan menyebarkan sinyal tersebut pada seluruh unit dan pada *hidden units*. Perlu diketahui bahwa input Xi yang dipakai disini adalah *input training* data yang sudah diskalakan. Pertama, *input* yang mungkin dipakai dalam sistem dicari nilai terendah dan tertingginya. Kemudian, skala yang digunakan tergantung dari fungsi aktifasinya. Jika yang dipakai adalah fungsi sigmoid Biner yang mempunyai harga terendah 0 dan harga tertinggi 1, maka nilai *input* terendah juga dianggap 0 dan harga tertinggi dianggap 1. Nila-nilai diantaranya bervariasi antara 0 dan 1.
b. Pada Setiap hidden units (Zj, j = 1, …, p), akan menjumlahkan sinyal-sinyal input yang sudah berbobot, termasuk biasnya,

$$z_{inj} = Voj + \sum_{i=1}^{n} Xi\ Vij \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari hidden unit yang bersangkutan. $Z_j = f(z_in_j)$ lalu mengirim sinyal output ini ke seluruh unit pada unit output.

c. Setiap unit output (Y_k, k=1,…,m), akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya,

$$y_{ink} = Wok + \sum_{j=1}^{p} Zj\ Wjk \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit *output* yang bersangkutan, $Y_k = f(y_in_k)$, lalu mengirim sinyal *output* ini ke seluruh unit pada unit *output*.

2. Perhitungan dan propagasi balik dari *error* yang bersangkutan

a. Setiap unit *output* (Y_k, k = 1,…,m) menerima suatu target *pattern* (*desired output*) yang sesuai dengan *input training pattern* untuk menghitung kesalahan (error) antara target dengan *output* yang dihasilkan jaringan,

$\delta_k = (t_k - y_k)\ f'(y_in_k)$…….(4)

Sebagaimana *input training* data, *output training* data t_k juga telah diskalakan menurut fungsi aktivasi yang dipakai.

Faktor δ_k ini digunakan untuk menghitung koreksi *error* (ΔW_{jk}) yang nantinya akan dipakai untuk memperbaharui W_{jk}, dimana : $\Delta W_{jk} = \alpha\ \delta_k\ Z_j$

Selain itu juga dihitung koreksi bias ΔW_{0k} yang nantinya akan dipakai untuk memperbaharui W_{0k}, dimana : $\Delta W_{0k} = \alpha\ \delta_k$

Faktor δ_k ini kemudian dikirimkan ke layer yang berada pada step berikutnya

b. Setiap hidden unit (Z_j, j = 1,…,p) menjumlah input delta (yang dikirim dari step sebelumnya) yang sudah berbobot.

$$\delta_{-inj} = \sum_{k=1}^{m} \delta k \, Wjk \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasiklkan faktor koreksi error δ_j, dimana :

$\delta_j = \delta_in_j \, f'(z_in_j)$ …….(6)

Faktor δ_j ini digunakan untuk menghitung koreksi error (ΔV_{ij}) yang nantinya akan dipakai untuk memperbaharui V_{ij}, dimana : $\Delta V_{ij} = \alpha \, \delta_j \, X_i$

Selain itu juga dihitung koreksi bias ΔV_{0j} yang nantinya akan dipakai untuk memperbaharui V_{0j}, dimana : $\Delta V_{0j} = \alpha \, \delta_j$

3. Pembaharuan (adjustment bobot dan bias)

a. Setiap unit output (Y_k, k = 1,…,m) akan memperbaharui bias dan bobotnya dari setiap hidden unit (j = 0, … ,p), W_{jk} (baru) = W_{jk} (lama) + ΔW_{jk}. Demikian pula untuk setiap hidden unit (Z_j, j = 1,,p) akan memperbaharui bias dan bobotnya dari setiap unit input (i = 0,,n), V_{ij} (baru) = V_{ij} (lama) + ΔV_{ij}

Memeriksa *Stopping Condition*

Jika Stopping Condition telah terpenuhi, maka pelatihan jaringan saraf dapat dihentikan. Ada dua cara yang bisa dipakai untuk memeriksa Stopping

Condition :

Pertama, dengan membatasi jumlah iterasi yang ingin dilakukan

Kedua, dengan membatasi error. Untuk metode *Backpropagation* dipakai metode *Mean Square Error* untuk menghitung rata-rata *error* antara *output*

yang dikehendaki pada *training* data dengan *output* yang dihasilkan dengan output yang dihasilkan oleh jaringan. Misalnya, jika *error* telah mencapai 0,01 (1%), pelatihan dihentikan.

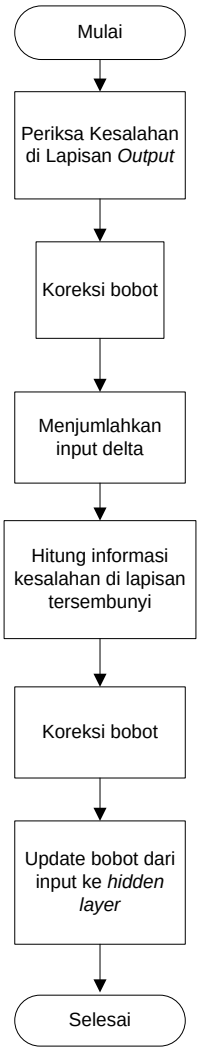
Pada step nomor 1 (poin a sampai poin c), dimana inputnya diambil dari input training set jika ingin dihitung adalah training error dan atau input test set jika yang ingin dihitung adalah test set error. Langkah ini dilakukan untuk semua data training / test yang ada.

Kemudian dicari selisih antara target *output* (t_k) dengan *output* jaringan (y_k) dan diimplemantasikan pada persamaan Mean Square Error. Jika terdapat m training data, maka :

Mean Square Error = $E = t * \{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \ldots + (t_{km} - y_{km})^2 \}$

3.6.3.2 Prosedur Peramalan

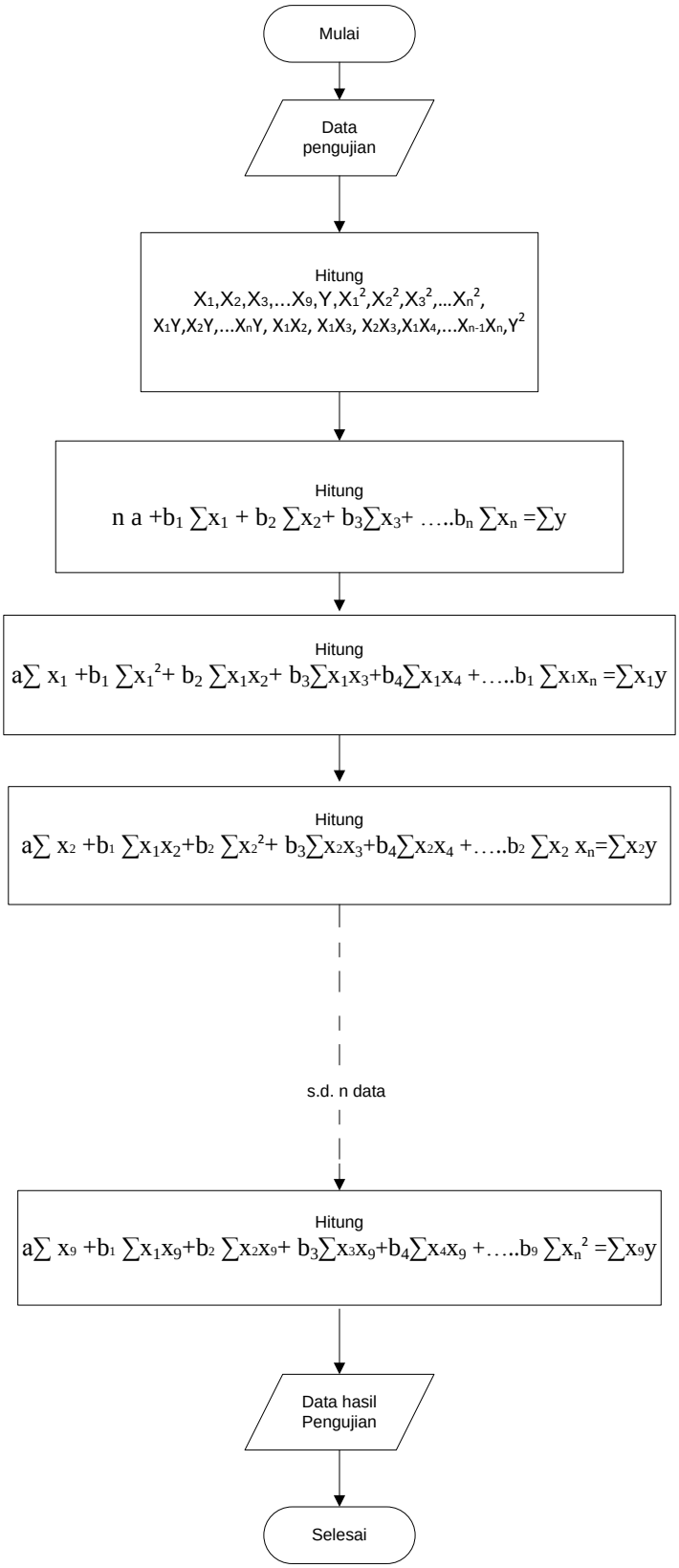
Proses ini menggunakan bobot JST yang diperoleh dari proses *training* untuk menguji data *testing* yang ada. Secara umum proses peramalan ini dapat digambarkan seperti gambar 3.6 berikut.



Gambar 3.6 Diagram Alur Proses Peramalan JST Backpropagation

3.6.3.3 Perancangan Metode *Multiple Regression*

Metode *multiple regression* dimulai dengan menginputkan data yang telah ada, kemudian memproses data yang diinputkan yang akan menghasilkan keluaran yang diinginkan. Secara umum alur proses *multiple regression* dapat dilihat pada gambar 3.7 dibawah ini.



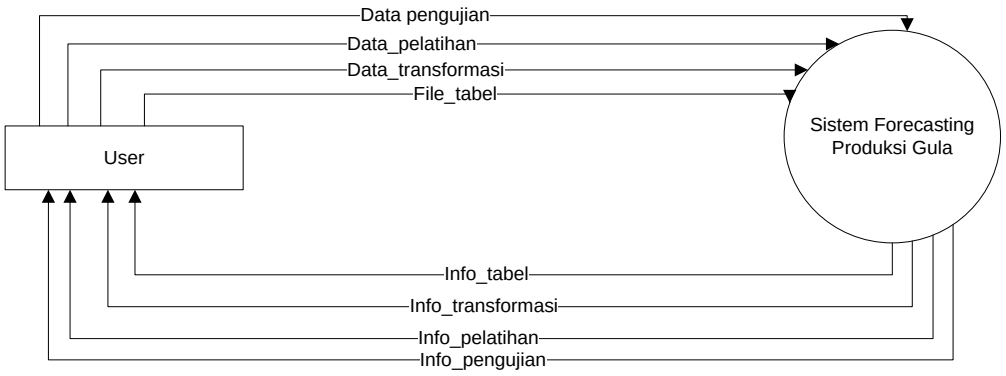
Gambar 3.7 Diagram Alur Proses Multiple Regression

3.6.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan, analisis kebutuhan fungsional dapat didefinsikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Tahapan ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan dari rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. Alat bantu yang digunakan untuk menggambarkan sistem secara umum yang akan dibangun yaitu diagram konteks dan *data flow diagram*. Untuk menjelaskan bagaimana suatu masukan diproses pada sistem maka digunakan spesifikasi proses dan kamus data untuk mengetahui aliran data yang mengalir pada sistem.

3.6.4.1 Diagram Konteks

Diagram konteks atau disebut juga dengan model sistem fundamental merepresentasikan seluruh elemen sistem sebagai sebuah bubble tunggal dengan data input output yang ditunjukan oleh anak panah yang masuk dan keluar secara berurutan.

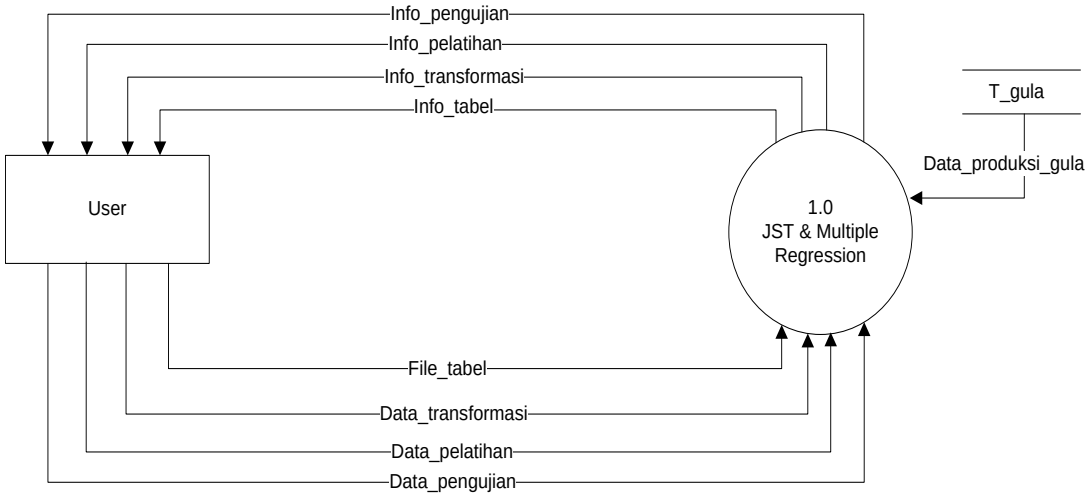


Gambar 3.8 *Diagram Konteks*

3.6.4.2 *Data Flow Diagram* (DFD)

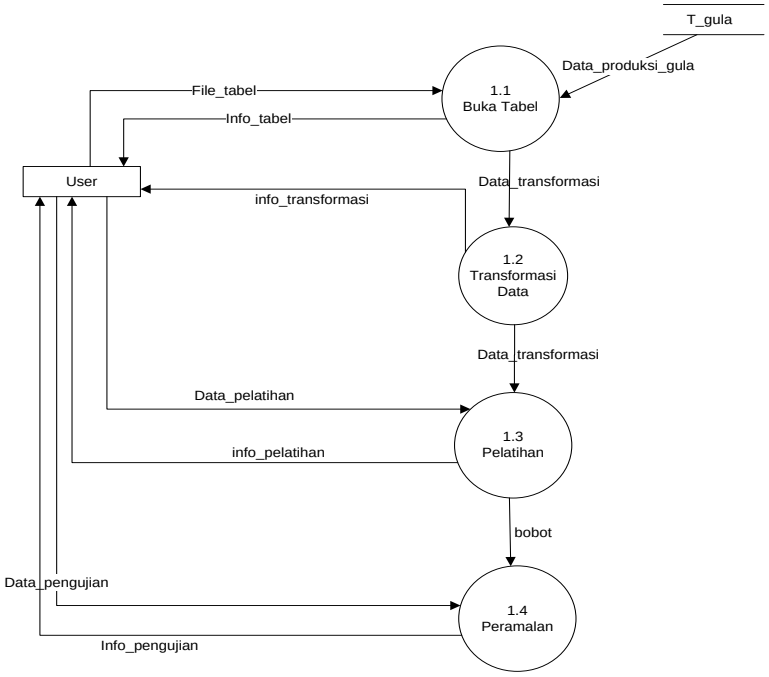
Dari diagram konteks di atas, aliran data yang ada pada sistem digambarkan dengan lebih rinci menggunakan *Data Flow Diagram*.

3.6.4.2.1 DFD *Level* 1 Sistem *Forecasting* Produksi Gula



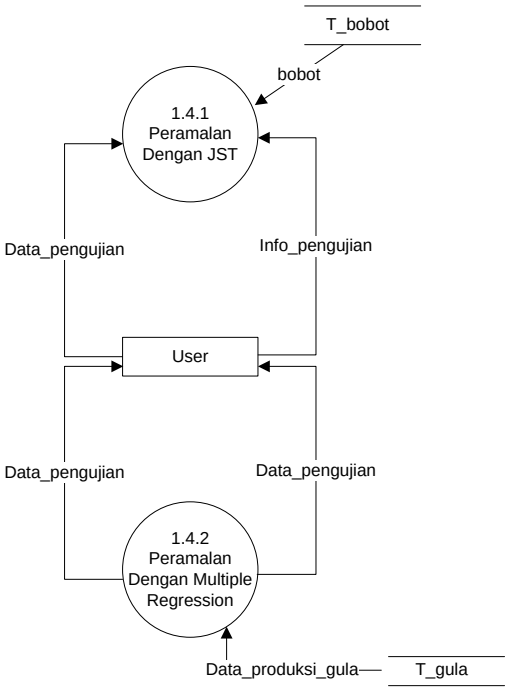
Gambar 3.9 Sistem Forecasting Produksi Gula

3.6.4.2.2 DFD *Level* 2 Proses JST & *Multiple Regression*



Gambar 3.10 DFD Level 2 Proses JST & Multiple Regression

3.6.4.2.3 DFD *Level* 3 Proses Peramalan



Gambar 3.11 DFD Level 3 Proses Peramalan

3.6.4.3 Spesifikasi Proses

Spesifikasi proses bertujuan untuk mendeskripsikan dari setiap fungsi yang disajikan pada diagram alir data. Spesifikasi proses pada alir data sistem aplikasi jaringan saraf tiruan adalah :

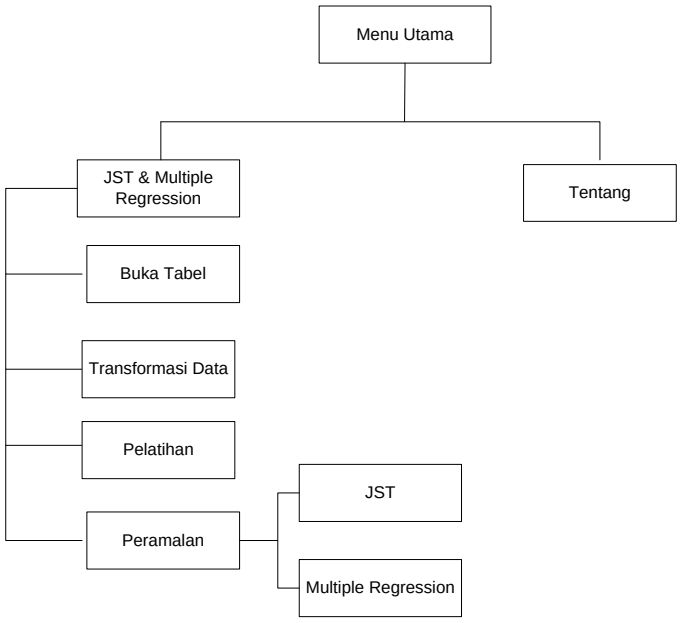
Tabel 3.1 Spesifikasi proses

No	Proses	Keterangan
1	No.Proses	1.1
	Nama proses	Buka Tabel
	Source	User
	Input	File tabel
	Output	Info tabel
	Destination	User
	Logika proses	1. user mencari file tabel yang ingin digunakan sebagai masukan ke dalam sistem. 2. Setelah file tersebut berhasil dimasukkan maka akan ada informasi mengenai isi tabel tersebut
2	No.Proses	1.2
	Nama proses	Transformasi Data
	Source	User
	Input	Data Transformasi
	Output	Info transformasi
	Destination	User
	Logika proses	1. user berhasil memasukkan file table 2. user memilih menu transformasi data 3. Sistem akan secara otomatis melakukan transformasi data sesuai dengan isi dari tabel yang digunakan sebagai masukan sistem
3	No.Proses	1.3
	Nama proses	Pelatihan
	Source	User
	Input	Data Pelatihan, data transformasi
	Output	Info pelatihan, bobot
	Destination	User
	Logika proses	1. user memilih menu pelatihan 2. user memasukkan data pelatihan 3. Data transformasi yang sudah di dapat dalam proses sebelumnya di gunakan sebagai masukan dalam proses ini . 4. Sistem akan melakukan proses pelatihan terhadap data yang telah dimasukkan oleh

		pengguna 5. Sistem akan menampilkan dan menyimpan hasil dari proses pelatihan.
4	No.Proses	1.4
	Nama proses	Peramalan
	Source	User
	Input	Data Pengujian
	Output	Info Pengujian
	destination	User
	Logika proses	1. user memilih menu Peramalan. 2. user memasukkan nilai dari setiap data pengujian. 3. Sistem akan melakukan perhitungan terhadap proses peramalan 4. Sistem akan menampilkan hasil perhitungan.

3.6.4.4 Struktur Menu

Berikut adalah struktur menu Aplikasi Jaringan Saraf Tiruan untuk forecasting produksi gula :



Gambar 3.12 Struktur menu aplikasi JST

3.6.4.5 Perancangan Arsitektur

Perancangan arsitektur merupakan perancangan yang dibuat sebelum program aplikasi dibuat. Dengan perancangan arsitektur akan mempermudah proses pembangunan aplikasi jaringan saraf tiruan

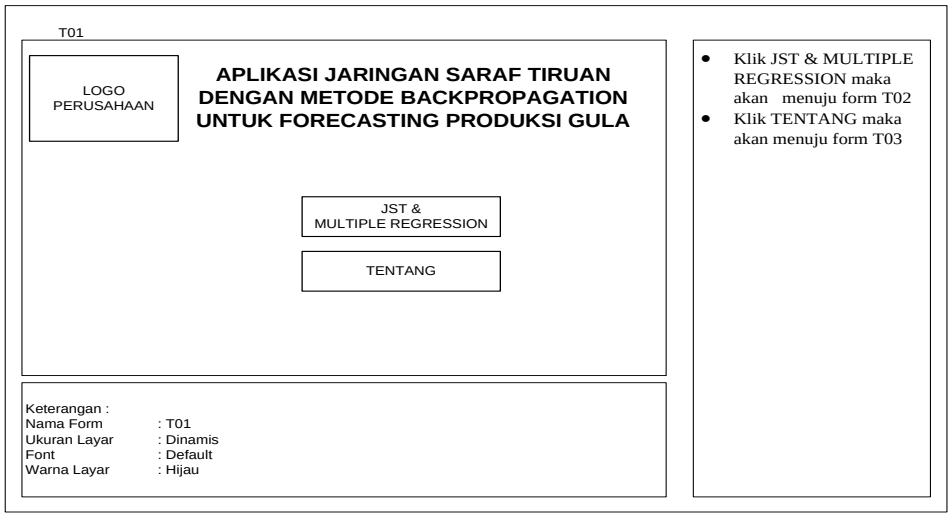
3.6.4.5.1 Perancangan Antar Muka

Interface atau antar muka merupakan tampilan dari suatu program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user.* Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*.

Perancangan *interface* untuk aplikasi Jaringan Saraf Tiruan adalah sebagai berikut :

1. *Form* Menu Utama

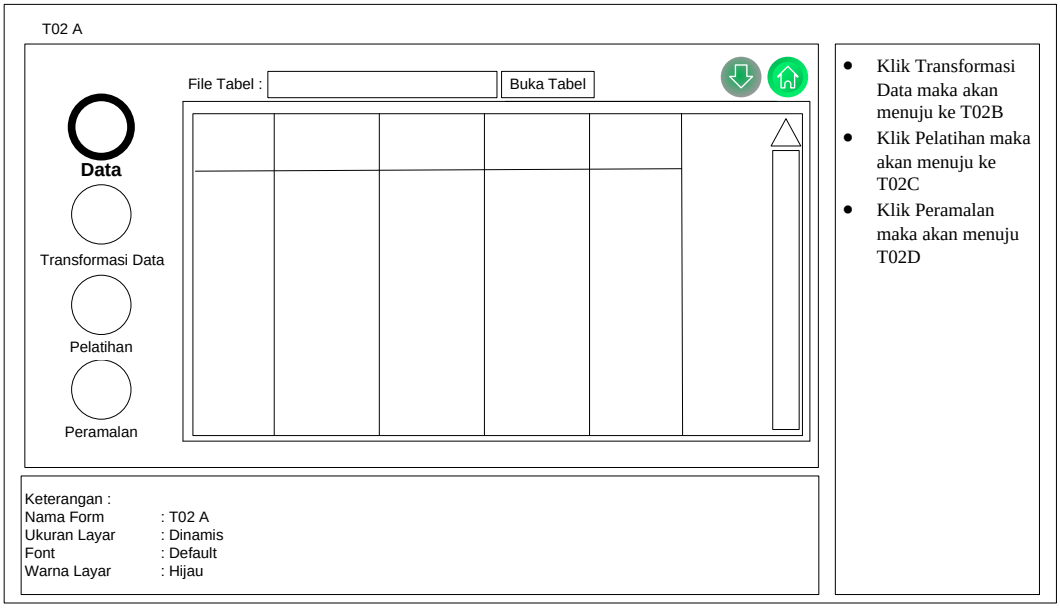
   Layar ini berfungsi menampilkan menu utama yang terdiri dari JST *Backpropagation*, *Multiple Regression* dan tentang.



Gambar 3.13 Form Menu Utama

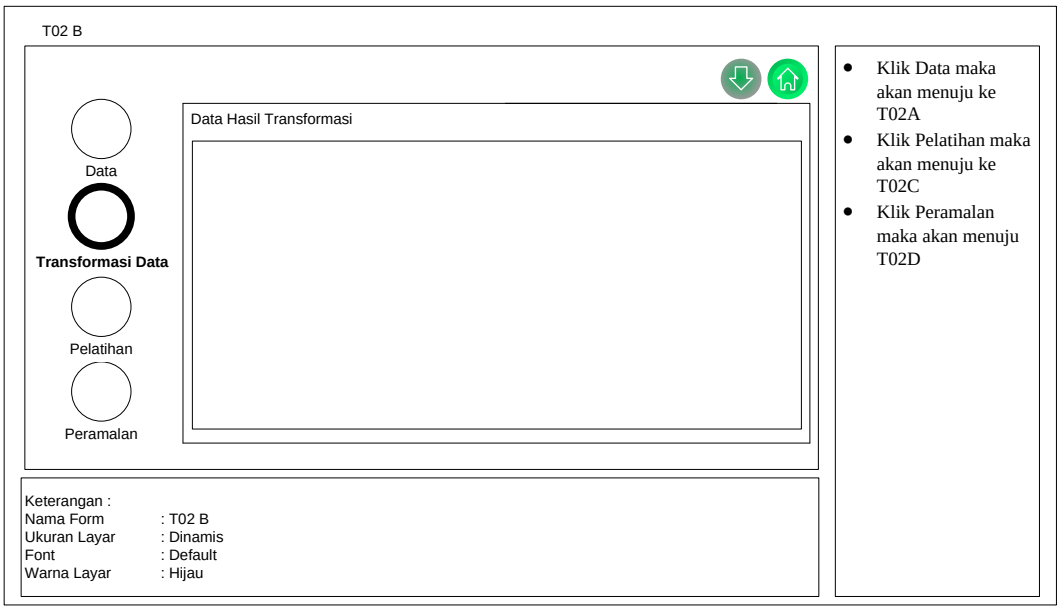
2. *Form* JST *Backpropagation* dan *Multiple Regression*

a. *Form* Data



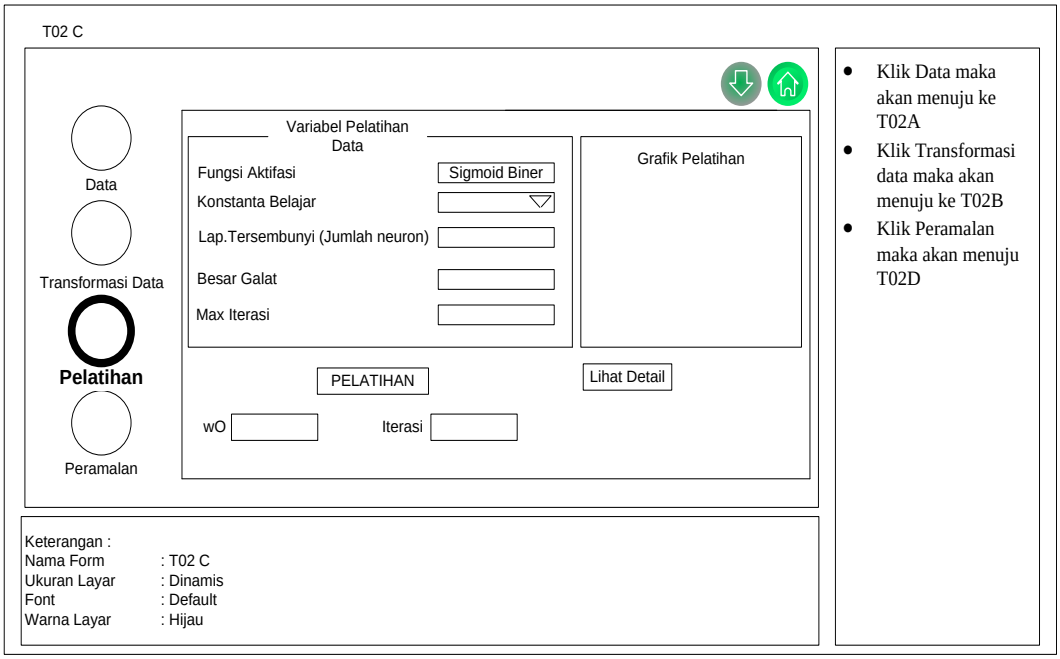
Gambar 3.14 Form Data

b. *Form* Transformasi Data



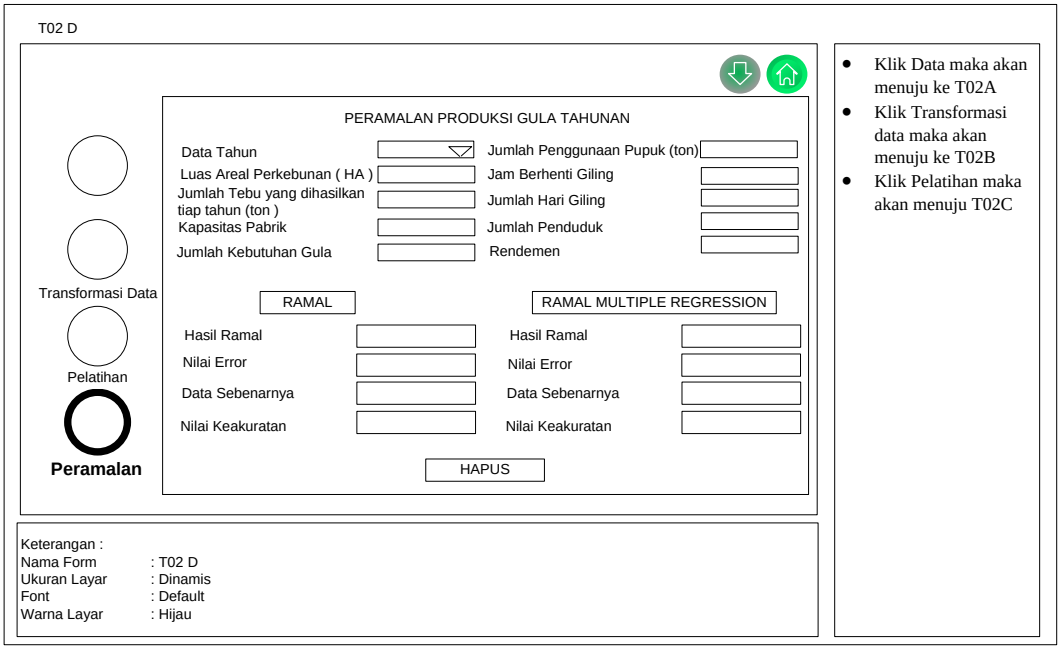
Gambar 3.15 Form Transformasi Data

c. *Form* Pelatihan



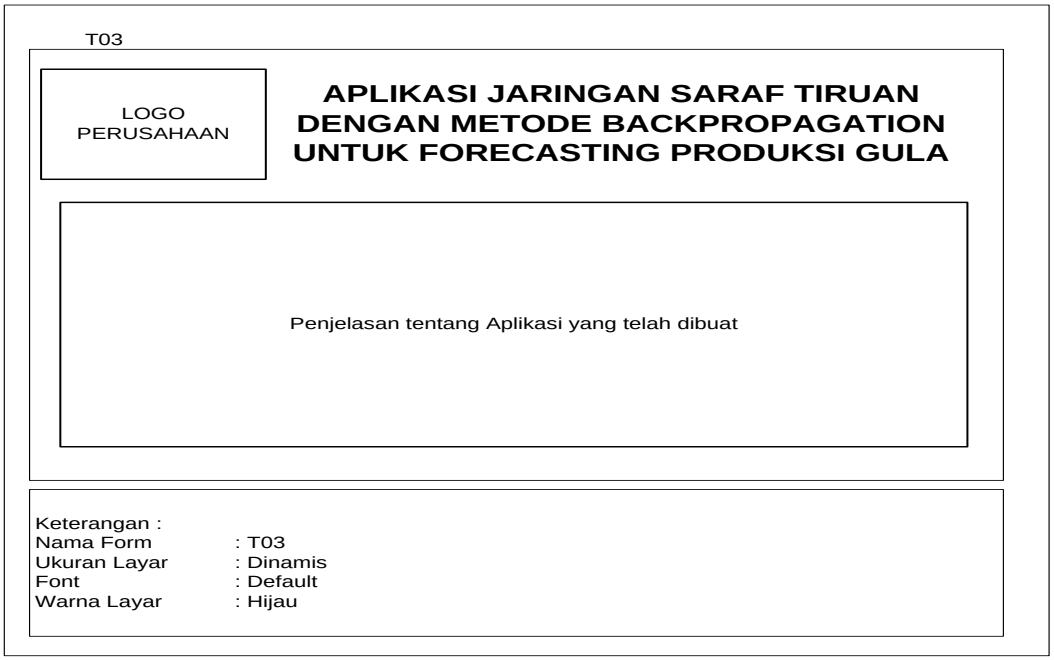
Gambar 3.16 Form Pelatihan

d. *Form* Peramalan



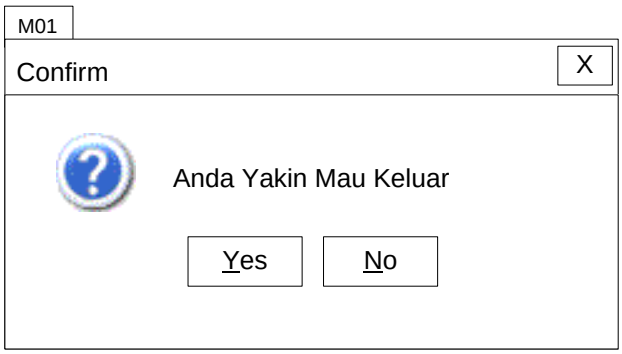
Gambar 3.17 Form Peramalan

3. Form Tentang

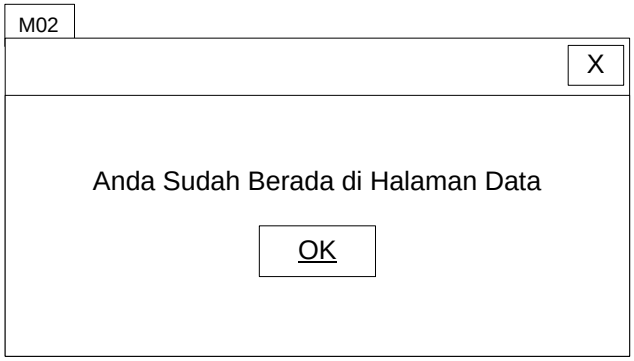


Gambar 3.18 Form Tentang

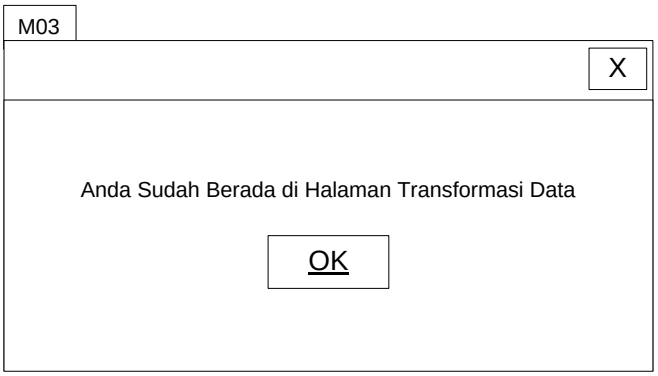
3.6.4.5.2 Perancangan Pesan



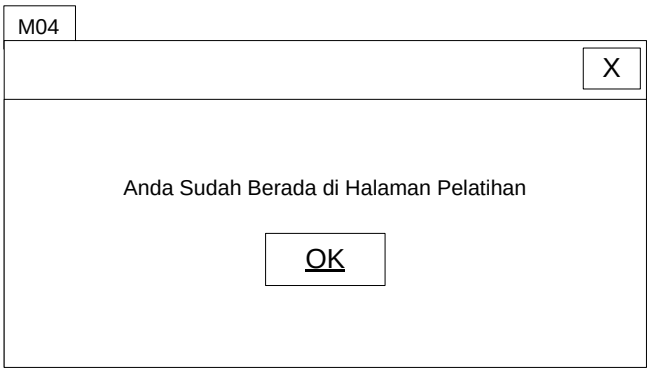
Gambar 3.19 Pesan Keluar



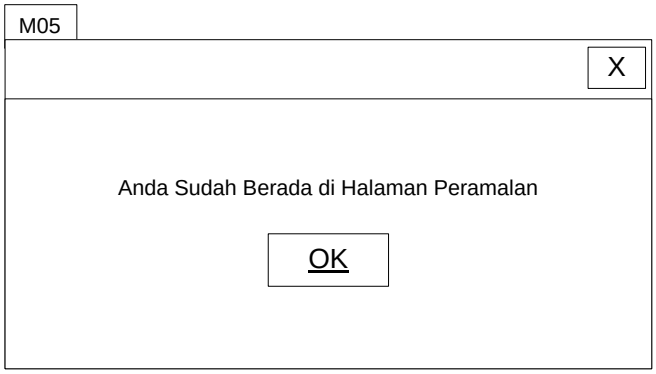
Gambar 3.20 Pesan Keterangan Halaman Data



Gambar 3.21 Pesan Keterangan Halaman Transformasi Data

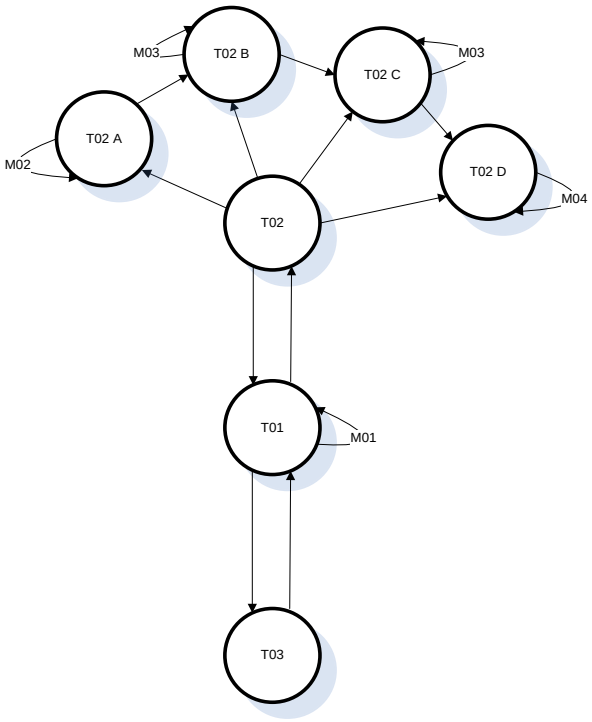


Gambar 3.22 Pesan Keterangan Halaman Pelatihan



Gambar 3.23 Pesan Keterangan Halaman Peramalan

3.6.4.5.3 Jaringan Semantik



Gambar 3.24 Jaringan Semantik Form Transformasi Data