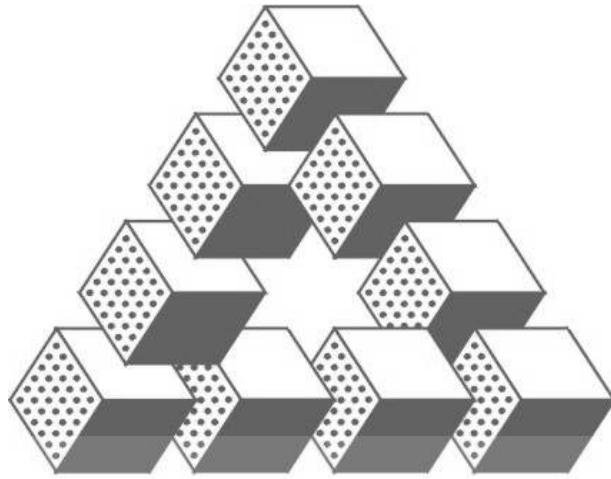


**PREDIKSI TINGKAT PENDAPATAN PETANI  
DENGAN PENERAPAN  
SISTEM NEURO-FUZZY STRUKTUR MOD\_ANFIS**



**STIKOM**  
Dinamika

Oleh :

Nama : Vendy Renato  
NIM : 01.41010.0250  
Program : S1 (Strata Satu)  
Jurusan : Sistem Informasi

**SEKOLAH TINGGI  
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER  
SURABAYA**

**2006**

**PREDIKSI TINGKAT PENDAPATAN PETANI  
DENGAN PENERAPAN  
SISTEM NEURO-FUZZY STRUKTUR MOD\_ANFIS  
SKRIPSI**

Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Komputer



Oleh :  
Nama : Vendy Renato  
NIM : 01.41010.0250  
Program : S1 (Strata Satu)  
Jurusan : Sistem Informasi

**SEKOLAH TINGGI  
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER  
SURABAYA  
2006**

**PREDIKSI TINGKAT PENDAPATAN PETANI  
DENGAN PENERAPAN  
SISTEM NEURO-FUZZY STRUKTUR MOD\_ANFIS**

Disusun Oleh :

**Nama : Vendy Renato**

**Nim : 01.41010.0250**

Surabaya, Agustus 2006

Telah diperiksa, diuji dan disetujui:

Pembimbing I

**Basuki Rahmat, S.Si, MT**  
NIDN. 0723076902

Pembimbing II

**Moch. Arifin, S.Pd, M.Si**  
NIDN. 0717106501

Mengetahui :

Wakil Ketua bidang Akademik

**Drs. Antok Supiyanto, M.MT**  
NIDN. 0726106201

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER SURABAYA



UNIVERSITAS  
**Dinamika**

*Segala sesuatu yang dimulai  
dengan ketulusan hati,  
maka yang akan didapat  
adalah kebahagiaan*



*Ku persembahkan kepada*

*Allah SWT*

*Ayah & Bunda tercinta*

*Adik-adik tercinta*

*Beserta semua orang yang sangat mendukung*

UNIVERSITAS  
Dinamika

## KATA PENGANTAR

Puji syukur atas karunia dan anugerah Allah SWT, sehingga penulis dapat menyelesaikan Tugas Akhir yang merupakan persyaratan dalam menyelesaikan Program Studi Strata Satu di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya.

Tugas Akhir ini membahas tentang *Prediksi Tingkat Pendapatan Petani dengan Neuro-Fuzzy struktur Mod ANFIS*. Sistem ini diharapkan akan mampu memprediksi pendapatan petani di masa yang akan datang, sehingga mampu mengatasi permasalahan yang berkaitan dengan produktivitas pertanian .

Pada kesempatan ini, penulis menyampaikan rasa penghargaan dan terima kasih kepada yang terhormat :

1. Bapak Basuki Rahmat, S.Si, M.T sebagai dosen pembimbing I atas segala arahan dan bimbingannya.
2. Bapak Moch. Arifin, S.Pd, M.si sebagai dosen pembimbing II atas segala bimbingan dan dukungannya.
3. Ayah, Ibu, Adik Angky, Inne atas segala bantuan, dukungan moral, doa, dan materi selama pengerjaan Tugas Akhir ini.
4. Bapak Wahyu untuk bantuannya dalam memperjuangkan proposal Tugas Akhir.
5. Bayu, Wawan, Didik, Alfi, Anger, Heru, Imam, Andi, Habibi, Modjo, Adam, Aris, Windi, Dian, Anita, dan Drian atas segala bantuan dan ide-ide yang diberikan selama proses penyusunan Tugas Akhir ini.

6. Rekan-rekan mahasiswa khususnya angkatan 2001 yang tidak dapat kami sebutkan satu-persatu atas semua bantuan, kekompakan dan dukungan moralnya.

Dalam menyusun Tugas Akhir ini, Semoga Allah SWT memberikan pahala yang setimpal kepada semua pihak yang telah memberikan bantuan, bimbingan ataupun nasehat-nasehat.

Penulis menyadari bahwa masih banyak kekurangan pada sistem ini. Namun penulis berharap semoga sistem ini dapat ikut menunjang perkembangan ilmu pengetahuan, khususnya bidang Teknologi Informasi.



Surabaya, Juli 2006

Penulis

## DAFTAR ISI

	Halaman
ABSTRAKSI .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR TABEL .....	ix
DAFTAR GAMBAR .....	x
DAFTAR LAMPIRAN .....	xii
<b>BAB I    PENDAHULUAN</b>	
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Pembatasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
<b>BAB II   LANDASAN TEORI</b>	
2.1 Teori Fuzzy .....	5
2.2 Sistem Neuro-fuzzy dengan Algoritma belajar yang telah dimodifikasi .....	7
2.3 Jaringan Saraf Tiruan (Artificial Neural Network).....	13
2.4 Teknik Peramalan .....	16
2.5 Prediksi tingkat pendapatan petani .....	21
<b>BAB III   PERANCANGAN SISTEM</b>	
3.1 Analisa Permasalahan .....	22



3.2 Model Pengembangan .....	22
3.3 Perancangan Desain Arsitektur .....	27
3.4 Perancangan Sistem .....	28
3.5 Desain Input/Output .....	36
3.6 Perancangan Data .....	42

#### BAB IV IMPLEMENTASI DAN EVALUASI

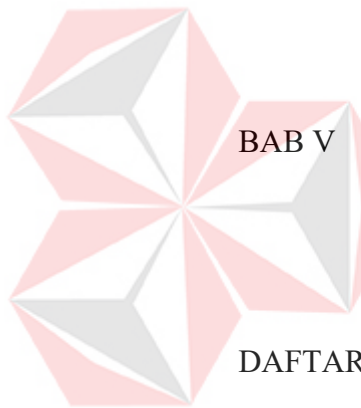
4.1 Instalasi Program .....	43
4.2 Implementasi Sistem .....	44
4.3 Pengujian Sistem .....	51
4.4 Hasil Prediksi .....	58
4.5 Evaluasi Sistem .....	59

#### BAB V PENUTUP

5.1 Kesimpulan .....	61
5.2 Saran .....	61

DAFTAR PUSTAKA .....	62
----------------------	----

LAMPIRAN .....	63
----------------	----



UNIVERSITAS  
Dinamika

## DAFTAR TABEL

	Halaman
Tabel 3.1. Data Harga Petani.....	35
Tabel 3.2. Data Pertanian .....	35
Tabel 3.3. Detil Belajar .....	36
Tabel 3.4. Uji Coba .....	36
Tabel 4.1. Hasil Proses Belajar .....	54
Tabel 4.2. Hasil Proses Validasi .....	55
Tabel 4.3. Kesalahan prediksi uji coba sistem .....	58
Tabel 4.4. Perbandingan hasil prediksi .....	59



UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur Mod_ANFIS.....	8
Gambar 2.2 Aturan koreksi kesalahan algoritma EBP modifikasi.....	11
Gambar 2.3 Ilustrasi Jaringan Saraf Tiruan .....	14
Gambar 2.4 Jaringan Saraf Single Layer .....	15
Gambar 2.5 Jaringan Saraf Multi Layer.....	16
Gambar 2.6 Pola Data Horizontal .....	18
Gambar 2.7 Pola Data Musiman .....	18
Gambar 2.8 Pola Data Siklis .....	19
Gambar 2.9 Pola Data Trend.....	19
Gambar 3.1 Alur Proses Pembelajaran .....	24
Gambar 3.2 Sub Proses Alur Tahap Maju .....	25
Gambar 3.3 Alur proses Uji Coba dan Validasi .....	26
Gambar 3.4 Arsitektur Mod_ANFIS tahap maju .....	27
Gambar 3.5 Arsitektur Mod_ANFIS tahap mundur.....	28
Gambar 3.6 Sistem flow prediksi tingkat pendapatan petani.....	29
Gambar 3.7 Context Diagram .....	30
Gambar 3.8 DFD Level 1 .....	31
Gambar 3.9 DFD level 2 Proses Maintenance Data .....	32
Gambar 3.10 CDM .....	33
Gambar 3.11 PDM .....	34
Gambar 3.12 Tampilan desain Form Data .....	38
Gambar 3.13 Tampilan desain Form Proses Belajar .....	39

Gambar 3.14 Tampilan desain form Uji Coba dan Validasi .....	40
Gambar 3.15 Tampilan desain form Detil.....	41
Gambar 3.16 Tampilan desain form Report.....	42
Gambar 4.1 Form Menu Utama .....	52
Gambar 4.2 Form Data .....	53
Gambar 4.3 Form Belajar .....	54
Gambar 4.4 Form Validasi .....	55
Gambar 4.5 Pemilihan Report.....	56
Gambar 4.6 Preview Report .....	56
Gambar 4.7 Form Uji Coba.....	57



UNIVERSITAS  
Dinamika

## DAFTAR LAMPIRAN

	Halaman
Lampiran 2. Source Code .....	63
Lampiran 3. Data Produksi Pertanian .....	85



UNIVERSITAS  
**Dinamika**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Sektor pertanian memberikan peran yang penting dalam pembangunan perekonomian nasional. Pentingnya sektor pertanian terhadap perekonomian Indonesia ini dapat dilihat dari aspek: penyediaan pangan, kontribusinya terhadap Produk Domestik Bruto (PDB), penyediaan lapangan kerja, dan sumbangannya terhadap perolehan devisa negara melalui ekspor. Disamping menyediakan bahan pangan bagi masyarakat, sektor pertanian juga menyediakan bahan baku untuk industri dan lapangan kerja bagi puluhan juta penduduk. Sektor pertanian khususnya agroindustri atau industri yang berbasis pertanian telah memberikan kontribusi yang berarti bagi pertumbuhan ekonomi nasional. Dengan demikian, penempatan sektor pertanian sebagai prioritas dalam kebijakan pembangunan pemerintahan sekarang merupakan strategi yang tepat dalam rangka peningkatan kesejahteraan rakyat.

Tingkat pendapatan petani yang tidak menentu bergantung pada hasil panen, sedikit banyak berdampak pada perubahan harga pokok hasil pertanian. Meski diakui banyak tantangan yang dihadapi petani dalam berusaha tani baik menyangkut lahan, serana pertanian hingga pada penjualan hasil panen. Apabila tingkat pendapatan petani dapat diprediksikan sejak dini maka dapat direncanakan berbagai anggaran oleh petani untuk memperoleh hasil panen yang lebih baik.

Berdasarkan ulasan diatas maka diperlukan suatu sistem yang dapat memprediksi tingkat pendapatan petani. Dimana sistem yang dikembangkan

merupakan teori dari logika fuzzy yang aturannya (*rule*) dibuat berdasarkan fungsi keanggotaan (*membership function*). Keandalan fuzzy ditentukan dari sistem keanggotaan yang mewakili sistem riil. Maka dari itu muncul ide untuk memilih parameter fungsi keanggotaan yang dapat ditentukan dengan bantuan *Artificial Neural Network* (ANN) dengan menggunakan aturan pembelajaran berdasarkan data latih. Sistem fuzzy dimana penentuan parameter dari fungsi keanggotaan dengan cara memanfaatkan arsitektur Jaringan Syaraf Tiruan dikenal dengan nama sistem *Neuro-Fuzzy*.

Dalam Tugas Akhir ini dikembangkan sistem neuro-fuzzy dengan struktur Modified Adaptive Neuro-fuzzy Inference Sistem (Mod\_ANFIS) yang merupakan modifikasi dari ANFIS standar untuk memprediksi tingkat pendapatan petani.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas dapat dibuat perumusan masalah sebagai berikut:

1. Bagaimana membuat suatu sistem untuk memprediksi tingkat pendapatan petani.
2. Bagaimana merancang model sistem Neuro-Fuzzy struktur Mod-ANFIS untuk memprediksi tingkat pendapatan petani.

## 1.3 Pembatasan Masalah

Batasan masalah dari sistem yang dibahas adalah sebagai berikut:

1. Penyiapan data latih berupa data pendapatan asli petani pada periode Januari 1989 sampai dengan Desember 2005 yang diperoleh di Badan Pusat Statistik.
2. Memprediksi jenis tanaman padi dan palawija.

3. Data pertanian dengan tiga kali masa panen per tahun, dengan ukuran perhitungan per hektar.
4. Menggunakan metode Neuro-Fuzzy struktur Mod\_ANFIS.

#### 1.4 Tujuan

Sesuai dengan permasalahan yang ada maka tujuan dibuatnya sistem ini adalah untuk mendapatkan data yang lebih akurat tentang prediksi tingkat pendapatan petani dalam periode tertentu.

#### 1.5 Sistematika Penulisan

Sistematika penulisan pada laporan ini adalah sebagai berikut:

##### **BAB I : PENDAHULUAN**

Pada bab ini membahas tentang latar belakang masalah dan penjelasan permasalahan secara umum, perumusan masalah serta batasan masalah yang dibuat, tujuan dari pembuatan tugas akhir dan sistematika penulisan buku ini.

##### **BAB II : LANDASAN TEORI**

Pada bab ini membahas secara singkat teori-teori yang berhubungan dan mendukung dalam pembuatan tugas akhir ini.

##### **BAB III : PERANCANGAN SISTEM**

Pada bab ini membahas tentang perancangan sistem, analisa sistem, Diagram Alir, *System Flow*, *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), struktur *database* yang digunakan dalam pembuatan aplikasi serta desain *input* dan *output*.



UNIVERSITAS  
Dinamika



**BAB IV : IMPLEMENTASI DAN EVALUASI**

Pada bab ini membahas tentang implementasi dari aplikasi yang dibuat secara keseluruhan. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui aplikasi tersebut telah dapat menyelesaikan permasalahan yang dihadapi sesuai dengan yang diharapkan.

**BAB V : PENUTUP**

Pada bab ini berisikan kesimpulan dari sistem dan saran untuk pengembangan sistem.



UNIVERSITAS  
**Dinamika**

## BAB II

### LANDASAN TEORI

#### 2.1 Teori Fuzzy

##### 2.1.1 Teori Himpunan Fuzzy

Teori himpunan fuzzy diperkenalkan oleh Lotfi A. Zadeh (1965). Teori ini didasarkan pada logika fuzzy yaitu terdapat nilai logika antara 0 dan 1 yang menyatakan tingkat kebenaran. Ia berpendapat bahwa logika benar dan salah dari logika Boolean tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Untuk mengatasi masalah gradasi yang tidak terhingga tersebut, Zadeh mengembangkan suatu himpunan Fuzzy. Tidak seperti Logika Boolean, logika Fuzzy mempunyai nilai kontinu. Fuzzy dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang bersamaan.

Misalkan  $V$  adalah kumpulan objek yang secara umum dinyatakan dengan  $\{v\}$ , yang bisa berharga diskrit atau kontinu.  $V$  disebut semesta pembicaraan (*universe of discourse*), dan  $v$  mewakili elemen-elemen  $V$ . Suatu himpunan fuzzy  $A$  dalam semesta pembicaraan  $V$  dapat dinyatakan oleh suatu fungsi keanggotaan  $\mu_A$  (*membership function*) yang mewakili nilai dalam interval nilai logika  $[0,1]$  untuk setiap  $v$  dalam  $V$  dan dinyatakan sebagai:

$$\mu_A = V \rightarrow [0,1] \quad (2.1)$$

Semua elemen  $v$  dalam  $V$  memberikan nilai  $\mu_A > 0$  disebut sebagai penyokong (*support*) dari himpunan fuzzy yang bersangkutan, jika  $\mu_A = 0.5$  maka

v disebut sebagai titik silang (*crossover*) dan himpunan fuzzy dimana penyokongnya bernilai 1.0 disebut sebagai fuzzy tunggal (*singleton*).

Pendekatan logika Fuzzy secara garis besar diimplementasikan dalam tiga tahapan yang dapat dijelaskan sebagai berikut :

1. Tahap Pengaburan (Fuzzyfication), yaitu pemetaan dari masukan tegas ke himpunan kabur.
2. Tahap Inferensi, yaitu pembangkitan aturan kabur.
3. Tahap Penegasan, yaitu tranformasi keluaran dari nilai kabur ke nilai tegas.

### 2.1.2 Fungsi Keanggotaan.

Fungsi keanggotaan yang sering digunakan adalah sebagai berikut:

1. Fungsi keanggotaan segitiga

Fungsi keanggotaan yang mempunyai parameter a,b dan c dengan formulasi

$$\text{segitiga}(x;a,b,c) = \max\left[\min\left(\frac{x-b}{b-a}, \frac{c-x}{c-b}\right), 0\right] \quad (2.2)$$

2. Fungsi keanggotaan trapesium

Fungsi keanggotaan yang mempunyai parameter a,b,c dan d dengan formulasi

$$\text{trapesium}(x;a,b,c,d) = \max\left[\min\left(\frac{x-b}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right] \quad (2.3)$$

3. Fungsi keanggotaan gaussian

Fungsi yang mempunyai parameter a,  $\sigma$  dengan formulasi

$$\text{gaussian}(x; \sigma, a) = \exp\left[-\frac{1}{2}\left(\frac{x-a}{\sigma}\right)^2\right] \quad (2.4)$$

4. Fungsi keanggotaan bell

Fungsi keanggotaan yang mempunyai parameter a,b,c dengan formulasi *bell*

$$\mu(x;a,b,c) = \frac{1}{\left(1 + \left|\frac{(x-c)^2}{a}\right|^{2b}\right)} \quad (2.5)$$

dengan  $b$  positif. Jika  $b$  negatif fungsi keanggotaan menjadi fungsi keanggotaan *bell* terbalik.

### 2.1.3 Basis Aturan

Basis aturan merupakan inti dari sistem fuzzy karena pada bagian ini terdapat sekumpulan aturan. Aturan *If-Then* Fuzzy atau *fuzzy conditional statements* adalah ekspresi dari bentuk IF A THEN B, dimana A dan B adalah label dari *Fuzzy Sets* menurut L.A Zadeh (1965) ditandai sesuai dengan *membership functions*. Dalam bentuk ringkasnya, aturan *if-then* fuzzy biasanya digunakan untuk menangkap mode yang tidak jelas dari pemikiran yang menjalankan peran penting dalam kemampuan manusia untuk membuat suatu keputusan dalam lingkungan yang tidak pasti dan tidak tepat.

## 2.2 Sistem Neuro-fuzzy dengan Algoritma Belajar yang telah Dimodifikasi

### (Modified ANFIS)

Neuro-fuzzy dengan struktur Mod\_ANFIS memiliki kelebihan dibandingkan ANFIS standar yaitu adanya proses yang dapat meminimalkan jaringan sehingga dapat mempercepat waktu proses. Pada ANFIS standar proses belajar yang digunakan melalui dua tahap, yaitu tahap maju dan tahap mundur (J.S Rojer Jang, 1997)

Secara garis besar proses belajar antara ANFIS standar dengan Mod\_ANFIS hampir sama yaitu dibagi menjadi dua tahap, tahap maju dan

mundur serta terdiri dari lima lapisan. Perbedaannya adanya aturan koreksi kesalahan dari *error backpropagation* (EBP).

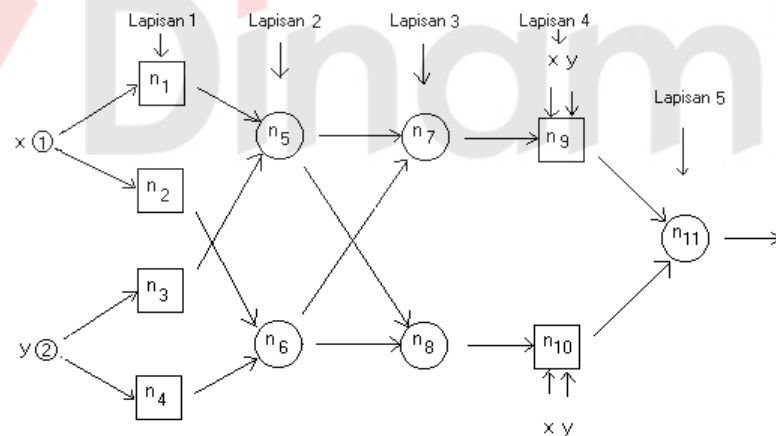
### 2.2.1 Aturan Belajar Mod\_ANFIS

Pada sistem Mod\_ANFIS penekanannya adanya modifikasi aturan koreksi kesalahan dari EBP. Algoritma belajar dari Mod\_ANFIS menggunakan dua tahap, yaitu Tahap Maju dan Tahap Mundur:

#### A. Tahap Maju

Tahap maju menggunakan mekanisme inferensi fuzzy dengan struktur ANFIS dan menggunakan metode *Least-Square Estimator* (LSE).

Arsitektur pada Mod\_ANFIS digambarkan dengan menggunakan dua masukan dan satu keluaran, dengan notasi sebagai berikut :



Gambar 2.1 Struktur Mod\_ANFIS

Mekanisme pada strukur Mod\_ANFIS dapat dijabarkan sebagai berikut:

#### 1. Lapisan 1

Mendefinisikan parameter fungsi keanggotaan ( $a1..a4$ ,  $b1..b4$ ,  $c1..c4$ ), kemudian mengimplementasikan fungsi keanggotaan. Pada lapisan ini (dalam

tugas akhir ini dipilih fungsi bell). Fungsi bell dapat menangani permasalahan yang kompleks dengan lebih baik (J.S Roger Jang, 1997). Dengan demikian keluaran dari simpul di lapisan ini merupakan fungsi bell. Untuk semua keluaran simpul pada tahap maju diberi simbol 'a', sehingga pada lapisan 1 diperoleh keluaran simpul  $n1a$  sampai dengan  $n4a$ . Tanda a untuk membedakan dengan nilai keluaran simpul yang baru yang diberi simbol 'b' (setelah dikoreksi). Pada lapisan ini bersifat adaptif karena adanya perubahan parameter. Untuk selanjutnya parameter-parameter ini disebut parameter premis.

## 2. Lapisan 2

Pada lapisan ini jika diterapkan logika fuzzy AND sebagai fungsi simpul, maka simpul keluaran yang dihasilkan adalah :

$$\begin{aligned} n5a &= \min(n1a, n3a) \\ n6a &= \min(n2a, n4a) \end{aligned} \quad (2.6)$$

## 3. Lapisan 3

Pada lapisan yang ini dilakukan normalisasi dari sinyal yang masuk, bentuk normalisasinya sebagai berikut

Misalkan  $ntot\_a = n5a + n6a$ , maka diperoleh

$$\begin{aligned} n7a &= n5a / ntot\_a \\ n8a &= n6a / ntot\_a \end{aligned} \quad (2.7)$$

## 4. Lapisan 4

Pada lapisan ini juga bersifat adaptif karena terdapat perubahan parameter. Dari sinyal yang masuk pada lapisan ini diperoleh matriks A, untuk Mod\_ANFIS matriks A dituliskan sebagai berikut:

$$A = [(n7a \ x) \ (n7a \ y) \ n7a \ (n8a \ x) \ (n8a \ y) \ n8a] \quad (2.8)$$

kemudian dengan metode *Least Squares Estimator* (LSE) diperoleh parameter-parameter konsekwen ( $p_1, q_1, r_1, p_2, q_2, r_2$ ). Di sini dituliskan:

$$\theta = [A^T A]^{-1} A^T U \quad (2.9)$$

dimana  $U$  sebagai keluaran pengontrol yang diinginkan, sehingga diperoleh parameter  $\theta = [p_1 \ q_1 \ r_1 \ p_2 \ q_2 \ r_2]^T$  dan

$$\begin{aligned} f_1 &= p_1 x + q_1 y + r_1 \\ f_2 &= p_2 x + q_2 y + r_2 \end{aligned} \quad (2.10)$$

Dengan demikian diperoleh keluaran simpul  $n_9$  dan  $n_{10}$ , yaitu:

$$\begin{aligned} n_{9a} &= n_{7a} f_1 \\ n_{10a} &= n_{8a} f_2 \end{aligned} \quad (2.11)$$

## 5. Lapisan 5

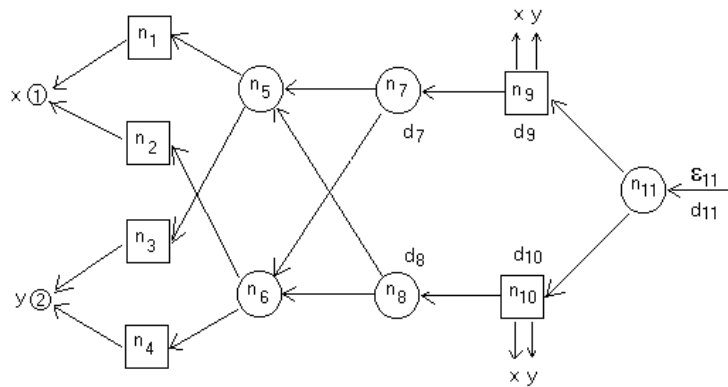
Pada lapisan terakhir ini selanjutnya keluar dari jaringan, outputnya berupa penjumlahan semua sinyal yang masuk, yaitu

$$n_{11a} = n_{9a} + n_{10a} \quad (2.12)$$

Tahap maju dari sistem Mod\_ANFIS telah selesai dilakukan, selanjutnya akan dilakukan proses ke tahap mundur.

## B. Tahap Mundur

Setelah semua proses pada tahap maju selesai dan diperoleh keluaran dari semua sinyal, kemudian *error* keluaran jaringan ini dipropagasibalik menggunakan aturan koreksi kesalahan yang baru yaitu dengan menggunakan algoritma EBP yang telah dimodifikasi. Selanjutnya proses belajar propagasibalik pada Mod\_ANFIS dapat digambarkan sebagai berikut:



Gambar 2.2. Aturan koreksi kesalahan menggunakan algoritma EBP yang telah dimodifikasi

Dari Gambar 1.9,  $\varepsilon_{11}$  diperoleh dari hasil turunan selisih keluaran jaringan dan keluaran yang diharapkan, jumlah kuadrat kesalahan dapat dijabarkan sebagai berikut:

$$Ep = \sum_{k=1}^{N(\ell)} (d_k^p - x_{L,k}^p)^2 \quad (2.13)$$

untuk keluaran sistem Mod\_ANFIS ini,  $Ep = \varepsilon_{11}$ ,  $x_i$  pada lapisan keluaran diberi notasi  $n_{11}$ , dan keluaran yang diharapkan  $d_k$  dituliskan sebagai  $U$ . Sehingga diperoleh hasil turunannya

$$\varepsilon_{11} = -2(U - n_{11a}) \quad (2.14)$$

Selanjutnya didefinisikan nilai  $d_{11}$  sebagai berikut:

$$d_{11} = -\varepsilon_{11}/2 = U - n_{11a} \quad (2.15)$$

Sehingga keluaran simpul  $n_{11}$  yang baru menjadi:

$$n_{11b} = n_{11a} + d_{11} \quad (2.16)$$

Padahal

$$n_{11b} = n_{9b} + n_{10b} \quad \text{karena} \quad n_{11a} = n_{9a} + n_{10a}$$

Jika didefinisikan  $n_{9b} = n_{9a} + d_9$  dan  $n_{10b} = n_{10a} + d_{10}$ , maka diperoleh

$$d_{11} = d_9 + d_{10}$$



Jika ruas kiri dikalikan dengan  $(f_1 + f_2)/(f_1 + f_2)$ , maka diperoleh:

$$\frac{d_{11} f_1}{f_1 + f_2} + \frac{d_{11} f_2}{f_1 + f_2} = d_9 + d_{10} \quad (2.17)$$

Dari sini bisa diasumsikan bahwa

$$d_9 = \frac{d_{11} f_1}{f_1 + f_2} \quad \text{dan} \quad d_{10} = \frac{d_{11} f_2}{f_1 + f_2} \quad (2.18)$$

Karena  $n_{9a} = n_{7a} f_1$  dan  $n_{10a} = n_{8a} f_2$ , maka  $n_{9b} = n_{7b} f_1$  dan  $n_{10b} = n_{8b} f_2$

atau sama dengan

$$n_{9a} + d_9 = (n_{7a} + d_7) f_1, \text{ dan}$$

$$n_{10a} + d_{10} = (n_{8a} + d_8) f_2$$

Sehingga diperoleh

$$\begin{aligned} d_8 &= d_{10} / f_2 \\ d_7 &= d_9 / f_1 \end{aligned} \quad (2.19)$$

Selanjutnya, ambil nilai  $ntot\_a$  hasil proses tahap maju, dan dituliskan  $ntot$  baru atau  $ntot\_b$  sebagai berikut

$$ntot\_b = ntot\_a + d\_tot; \quad (2.20)$$

Dimana nilai  $d\_tot$  bisa dipilih nilai sembarang, dipilih yang sesuai melalui uji coba sistem, misal jika dipilih  $d\_tot = 0$ , berarti  $ntot\_b = ntot\_a$ . Selanjutnya sesuai dengan persamaan (2.1) dituliskan nilai bayangan keluaran simpul yang baru di lapisan 2 yaitu

$$\begin{aligned} n_{5b1} &= (n_{7a} + d_7) ntot\_b \\ n_{6b1} &= (n_{8a} + d_8) ntot\_b \end{aligned} \quad (2.21)$$

Proses selanjutnya berada pada lapisan pertama tahap mundur, disini diterapkan fungsi turunan keanggotaan Bell.

$$Y = \text{Bell}(x, a, b, c)$$

$$Y = n5b1 \quad \text{untuk} \quad n1b \text{ dan } n3b$$

$$Y = n6b1 \quad \text{untuk} \quad n2b \text{ dan } n4b$$

Sehingga diperoleh persamaan parameter premis yang baru, yaitu:

$$a_{\text{baru}} = a_{\text{lama}} + \frac{dy}{da} \quad (2.22)$$

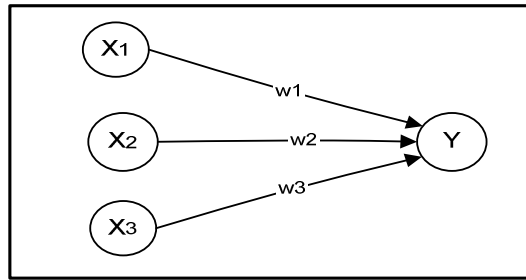
$$b_{\text{baru}} = b_{\text{lama}} + \frac{dy}{db} \quad (2.23)$$

$$c_{\text{baru}} = c_{\text{lama}} + \frac{dy}{dc} \quad (2.24)$$

Fungsi turunan bell yang dipakai adalah parameter  $a$ ,  $b$  dan  $c$ . Untuk  $x$  atau inputan tidak dipergunakan karena yang dipakai sebagai acuan adalah perubahan pada parameter premis (lapisan ke-1). Parameter-parameter yang telah diperbaharui selanjutnya digunakan untuk proses Mod\_ANFIS yang baru, sampai didapat *error* yang minimum dari beberapa proses yang dilakukan.

## 2.1 Jaringan Saraf Tiruan (Artificial Neural Network)

Jaringan Saraf Tiruan atau *Artificial Neural network* adalah model pemrosesan komputasi yang diinspirasi dari cara kerja sistem jaringan saraf manusia (Setiawan, Kuswara, 2003). Pada jaringan saraf tiruan juga terdapat istilah *neuron* atau sering disebut *node*. Setiap neuron akan terhubung dengan neuron yang lain melalui layer dengan bobot tertentu. Sedangkan setiap neuron mempunyai *internal state* yang disebut aktivasi. Aktivasi tersebut merupakan fungsi dari input yang diterima. Suatu neuron akan mengirimkan sinyal ke neuron-neuron yang lain.



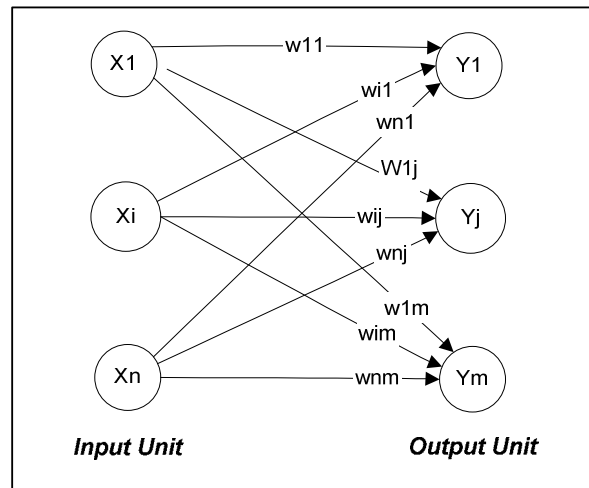
Gambar 2.3. Ilustrasi Jaringan Saraf Tiruan

Misalnya sebuah neuron Y pada Gambar 2.3 menerima input dari neuron X1, X2, X3. Aktivasi atau output sinyal adalah  $f(y)$ . Bobot yang menghubungkan neuron X1, X2, X3 ke neuron Y adalah  $w_1$ ,  $w_2$ ,  $w_3$ . Maka input jaringan  $y_{in} = \sum X_i w_i$  pada neuron Y adalah jumlah dari sinyal bobot dan masing-masing neuron X1, X2, X3, yaitu diantaranya dengan fungsi aktivasi

$$f(y) = \frac{1}{1 + \exp(-y_{in})} \quad (2.25)$$

### 2.3.1 Arsitektur Jaringan Saraf Tiruan

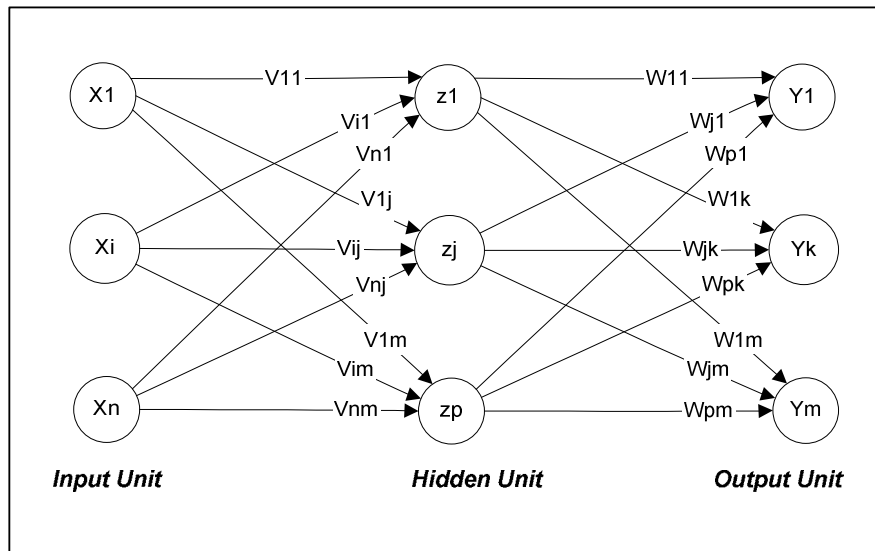
Pada Jaringan Saraf Tiruan, neuron-neuron tersusun dalam layer. Neuron yang terletak pada layer yang sama biasanya mempunyai hubungan yang sama antara satu dengan lainnya. Jaringan saraf dapat di klasifikasikan menjadi dua jenis yaitu *single layer* dan *multi layer* (Setiawan, Kuswara, 2003).



Gambar 2.4. Jaringan Saraf *Single Layer*

Dalam jaringan *single layer*, neuron-neuron dapat dikelompokkan menjadi dua bagian yaitu *input* dan *output* seperti pada Gambar 2.4. Unit-unit input akan menerima masukan dari luar, sedangkan unit-unit output akan mengeluarkan respon sesuai dengan masukannya.

Sedangkan dalam jaringan *multilayer* seperti pada Gambar 2.5, selain ada unit-unit input dan output, juga terdapat unit-unit yang tersembunyi (*hidden unit*). Jumlah hidden unit menyesuaikan dengan kebutuhan yang ada. Semakin kompleks jaringan, *hidden unit* yang dibutuhkan juga semakin banyak, demikian juga dengan jumlah layernya. Jaringan *multilayer* sering digunakan untuk persoalan yang lebih rumit karena pelatihan untuk hal yang kompleks akan lebih berhasil jika menggunakan jaringan *multilayer* (Setiawan, Kuswara, 2003)



Gambar 2.5. Jaringan Saraf *Multi Layer*

## 2.4 Teknik Peramalan

Teknik peramalan dapat diaplikasikan pada dunia bisnis yang berorientasi pada waktu yang akan datang, baik pada bidang keuangan, sumber daya manusia, produksi dan pemasaran. Situasi dan kondisi pada waktu yang akan datang tidaklah dapat diperkirakan secara pasti, sehingga kita perlu adanya usaha untuk meminimalkan ketidakpastian itu dengan cara menggunakan metode atau teknik peramalan tertentu (Aritonang R. Lerbin R., 2002).

Data merupakan salah satu unsur utama yang tidak dapat dipisahkan dari kegiatan peramalan. Tipe pola suatu data dapat mempengaruhi hasil peramalan. Berdasarkan dimensi waktunya, data dibedakan menjadi data *time series* (runtut waktu) dan data *cross sectional*. Data *time series* (runtut waktu) merupakan data yang diperoleh dari waktu ke waktu berikutnya selama kurun waktu tertentu. Misalnya data harga yang didapat petani selama dua belas bulan dalam jangka waktu satu tahun. Data *cross sectional* merupakan data yang dikumpulkan pada satu waktu tertentu, tanpa memiliki variasi dimensi waktu. Misalnya, data

mengenai penjualan beberapa perusahaan pada tahun 1998. Dalam konteks peramalan data yang lebih relevan adalah data runtut waktu.

Berdasarkan metode yang diterapkan, peramalan dapat dibagi menjadi dua kategori, yaitu:

1. Metode kualitatif

Metode kualitatif merupakan metode yang lebih didasarkan pada intuisi dan penilaian orang yang melakukan peramalan daripada pengolahan dan analisa data historis yang telah tersedia.

2. Metode kuantitatif

Metode kuantitatif didasarkan pada manipulasi atas data yang tersedia secara memadai tanpa adanya penilaian subjektif dari orang yang melakukan peramalan. Peramalan dengan metode kuantitatif dapat diterapkan bila terdapat tiga kondisi (Makridakis, 1993), yaitu:

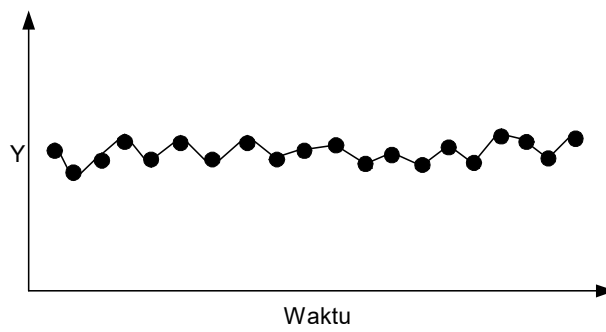
1. Adanya informasi keadaan di masa lalu
2. Informasi tersebut dapat dikuantitatifkan dalam bentuk data numerik
3. Dapat diasumsikan bahwa beberapa aspek pola masa lalu akan terus berlanjut di masa yang akan datang.

Pola data dapat dibedakan menjadi empat jenis siklis dan *trend* (Makridakis, 1993), yaitu:

1. **Pola horizontal**

Pola horizontal (H) terjadi bilamana nilai data berfluktuasi disekitar nilai rata-rata yang konstan. (Deret seperti itu "*stationer*" terhadap nilai rata-ratanya).

Gambar 2.6 menunjukkan pola khas dari data horisontal atau stasioner.

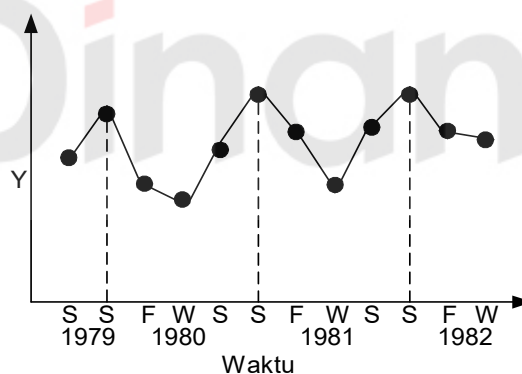


Gambar 2.6. Pola Data Horizontal

## 2. Pola musiman

Pola musiman (S) terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu). Untuk pola musiman kuartal, datanya dapat diilustrasikan pada Gambar

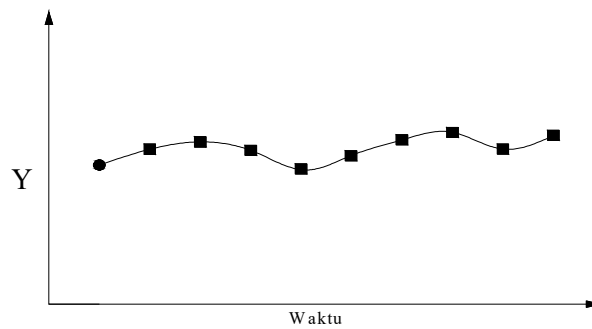
2.7.



Gambar 2.7. Pola Data Musiman

## 3. Pola siklis (C)

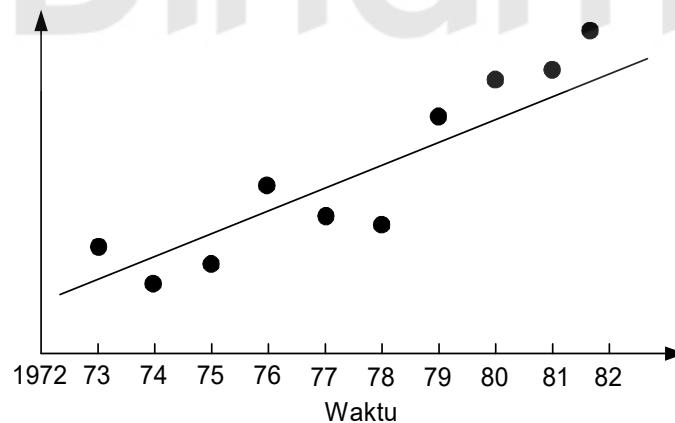
Pola Siklis (C) terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Penjualan produk seperti mobil, baja, dan peralatan utama lainnya menunjukkan jenis pola ini seperti ditunjukkan pada Gambar 2.8.



Gambar 2.8. Pola Data Siklis

#### 4. Pola *trend* (T)

Pola *trend* terjadi bilamana terdapat banyak kenaikan atau penurunan sekuler jangka panjang dalam data. Penjualan banyak perusahaan, produk bruto nasional (GNP) dan berbagai indikator bisnis atau ekonomi lainnya mengikuti suatu *trend* selama perubahannya sepanjang waktu. Gambar 2.9 menunjukkan salah satu pola data *trend*.



Gambar 2.9. Pola Data Trend

##### 2.4.1 Ukuran ketepatan metode peramalan

Dalam pemodelan deret berkala, sebagian data yang diketahui dapat digunakan untuk meramalkan data berikutnya. Nilai kesalahan peramalan yaitu



selisih antara data peramalan terhadap data aktual. Nilai kesalahan yang terjadi merupakan suatu data penting untuk menilai ketepatan suatu metode peramalan (Makridakis, 1993). Dari hasil peramalan akan diperoleh metode mana yang memperoleh kesalahan terkecil, sehingga nilai peramalan dapat dipakai sebagai acuan dalam menentukan kebutuhan-kebutuhan di masa yang akan datang. Terdapat beberapa ukuran kesalahan dalam peramalan antara lain:

### 1. Root Mean Squared Error

Kesalahan rata-rata akar kuadrat atau *Root Mean Squared Errors* (RMSE) dengan persamaan:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (X_t - F_t)^2}{n}}$$

*RMSE* adalah metode untuk mengevaluasi teknik peramalan. Akar kuadrat dari hasil masing-masing kesalahan (selisih data aktual dengan data peramalan dikuadratkan kemudian dijumlahkan kemudian dibagi dengan jumlah data).

### 2. Mean Absolute Percentage Error

Nilai tengah kesalahan presentase absolute atau *Mean Absolute Percentage Error* (MAPE), dengan persamaan:

$$MAPE = \frac{\sum_{t=1}^n \frac{|X_t - F_t|}{X_t}}{n} 100\%$$

MAPE merupakan presentase yang dihitung dari nilai absolut kesalahan di masing-masing periode dan dibagi dengan nilai data aktual periode tersebut kemudian dicari rata-rata kesalahan.

Keterangan:

$X_t$  = nilai aktual periode t

$F_t$  = nilai peramalan pada periode t

$X_t - F_t$  = nilai kesalahan peramalan (error)

n = jumlah data

## 2.5 Prediksi Tingkat Pendapatan Petani

Sesuai dengan teori ekonomi pertanian, ada beberapa hal yang menentukan tingkat pendapatan petani, yaitu pertama nasib petani hanya dapat diperbaiki melalui kenaikan harga (price) produk-produk pertanian. Kenaikan harga produk-produk pertanian berpengaruh langsung terhadap pendapatan petani yang secara otomatis juga meningkat.

Alasan kedua, peningkatan penawaran (supply) produk-produk pertanian akan menurunkan harga dan pendapatan total (total revenue) petani. Alasan ketiga, menyangkut ekonomi makro, kenaikan harga produk pertanian akan menyebabkan penurunan daya beli masyarakat serta memicu inflasi sehingga pada akhirnya pendapatan riil petani itu sendiri turun. Pendapatan petani bergantung pada tingkat produktivitas hasil-hasil pertanian yang dihasilkannya. Harga yang diterima petani dari hasil-hasil pertaniannya juga berubah-ubah. Pengeluaran-pengeluaran yang ada pun harus diperhitungkan dengan matang, sehingga diharapkan memperoleh keuntungan bersih yang optimal. Meskipun nilai produksi pada hasil panen itu tinggi, namun pengeluaran juga sangat besar, sehingga laba bersih petani tidak maksimal, tentunya juga bukan sesuatu yang optimal.

## BAB III

### PERANCANGAN SISTEM

#### 3.1 Analisa Permasalahan

Sebelum revolusi hijau dijalankan, petani terdahulu melakukan sistem pertanian mendekati sistem alami, yaitu bersifat daur ulang sisa panen tanpa input senyawa anorganik. Pada saat itu belum ada upaya pengendalian hama penyakit/gulma dengan menggunakan bahan kimia, serta modal yang minim sehingga produktivitasnya relatif masih rendah. Hal ini berpengaruh pada tingkat pendapatan para petani yang dapat dikatakan rendah. Apabila tingkat pendapatan petani dapat diprediksikan sejak dini maka dapat direncanakan berbagai anggaran oleh petani untuk memperoleh hasil panen yang lebih baik.

Mengingat pentingnya nilai pendapatan petani maka diperlukan adanya sistem yang mampu memprediksikan tingkat pendapatan petani dan tingkat produktivitas hasil panen berdasarkan data tahun-tahun sebelumnya. Selanjutnya, hasil keluaran jaringan dan selisih jaringan akan dibandingkan dengan hasil prediksi yang telah dilakukan oleh Deptan. Dengan menggunakan salah satu ukuran ketepatan peramalan, keluaran jaringan dan hasil prediksi akan dapat dilakukan perbandingan. Data *input* dan *output* tersebut dihubungkan suatu fungsi yang akan dibangkitkan dengan Neuro-Fuzzy. Dari penyesuaian antara *input* dan *output* akan menghasilkan suatu fungsi yang dapat memetakan *input* ke *output*. Apabila *error* (kesalahan) yang dihasilkan oleh fungsi kecil maka fungsi yang didapat sudah optimal.

### 3.2 Model Pengembangan

Tugas Akhir ini untuk mengembangkan sebuah sistem yang dapat memprediksi tingkat pendapatan petani dengan menggunakan sistem Neuro-Fuzzy dengan struktur Mod\_ANFIS.

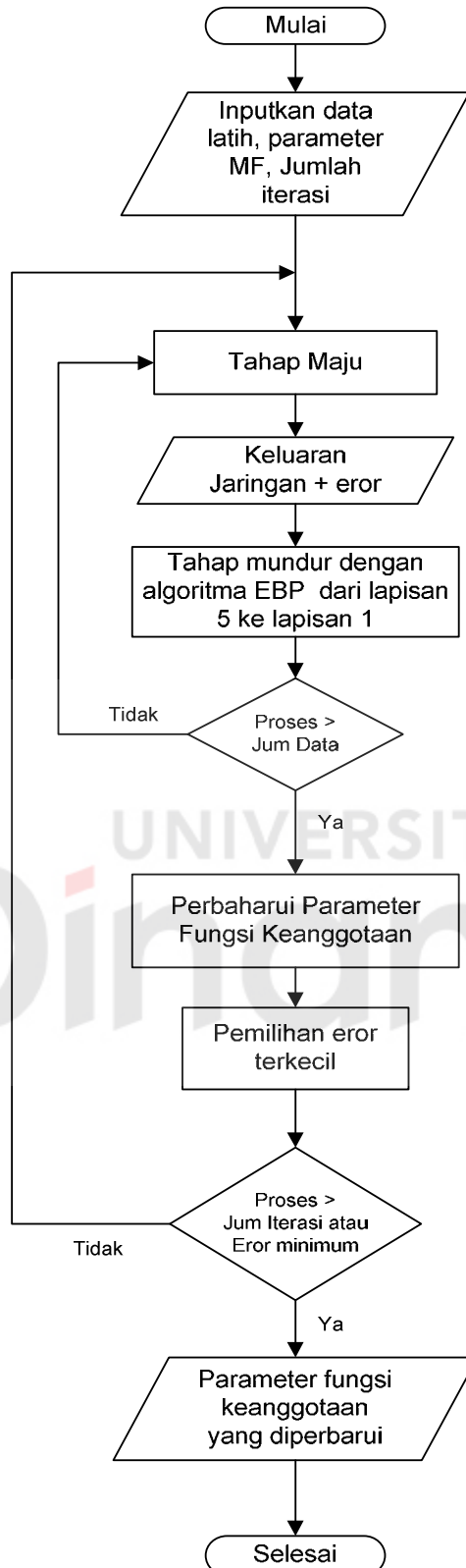
Alur model pengembangan sistem dibagi menjadi dua, yaitu:

1. Alur Proses Pembelajaran.

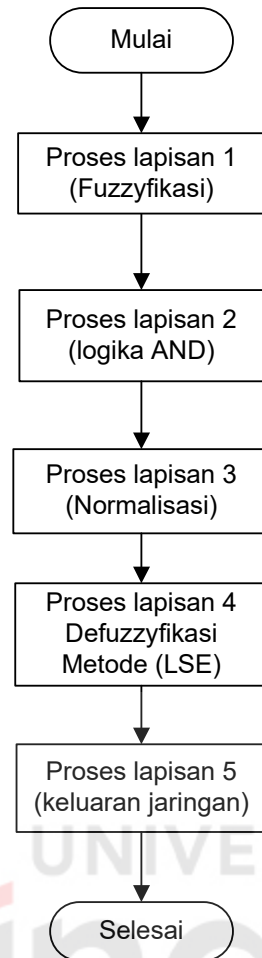
Alur Proses Pembelajaran digunakan untuk memperoleh parameter premis yang baru (parameter premis yang telah diperbaharui). Di dalam alur proses pembelajaran terdapat sub Proses Alur Maju yang terdiri dari lapisan 1 sampai dengan lapisan 5. Alur Proses Pembelajaran dapat dilihat pada Gambar 3.1 sedangkan untuk sub proses Alur Maju dapat dilihat pada Gambar 3.2.

2. Alur Proses Uji Coba.

Alur Proses Uji Coba hanya terdapat proses maju (lapisan 1 sampai dengan lapisan 5) tetapi parameter premis yang digunakan adalah parameter premis hasil belajar. Alur Proses Uji Coba dapat dilihat pada Gambar 3.3.

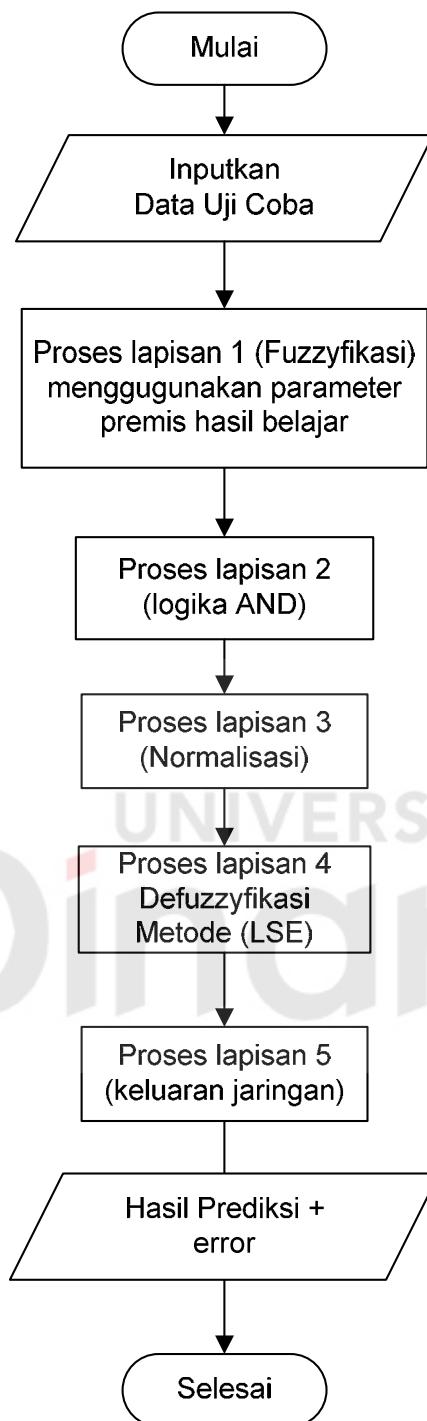


Gambar 3.1. Alur Proses Pembelajaran



Gambar 3.2. Sub Proses Alur Tahap Maju

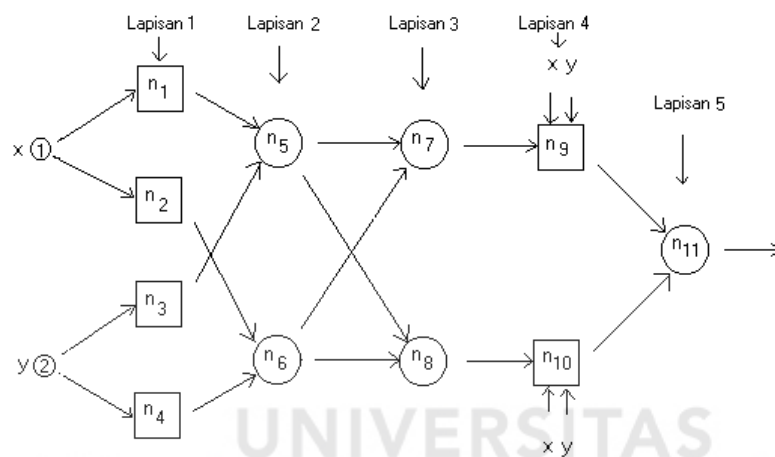
Sub Proses Alur Tahap Maju seperti pada Gambar 3.2 terdiri dari lapisan 1 sampai dengan lapisan 5. Lapisan 1 terdapat proses fuzzyfikasi, yaitu mengubah masukan nilai *crips* (nilai aktual) menjadi nilai fuzzy dengan menggunakan fungsi keanggotaan bell. Lapisan 2 penerapan logika fuzzy AND yaitu memilih node yang terkecil dari node yang masuk. Lapisan 3 bentuk normalisasi. Lapisan 4 Defuzzyfikasi dengan menggunakan metode LSE. Lapisan terakhir tahap maju merupakan keluaran jaringan.



Gambar 3.3. Alur Proses Uji Coba dan Validasi

### 3.3 Perancangan Desain Arsitektur

Sistem Neuro-Fuzzy dengan struktur Mod\_ANFIS yang dirancang disesuaikan dengan format data latih, yaitu jaringan dengan 2 masukan dan 1 keluaran. Perancangan arsitektur jaringan seperti yang terlihat pada Gambar 3.4.

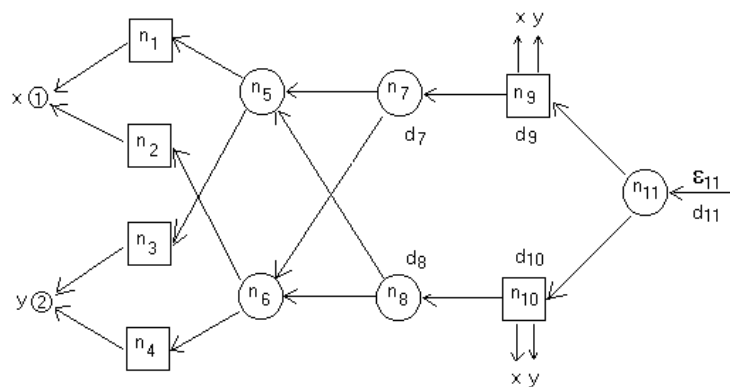


Gambar 3.4. Arsitektur Mod\_ANFIS tahap maju

Keterangan:

1. Lapisan 1 proses Fuzzyfikasi.
2. Lapisan 2 penerapan logika AND.
3. Lapisan 3 proses Normalisasi.
4. Lapisan 4 proses Defuzzifikasi dengan metode LSE
5. Lapisan 5 Keluaran jaringan





Gambar 3.5. Arsitektur Mod\_ANFIS tahap mundur

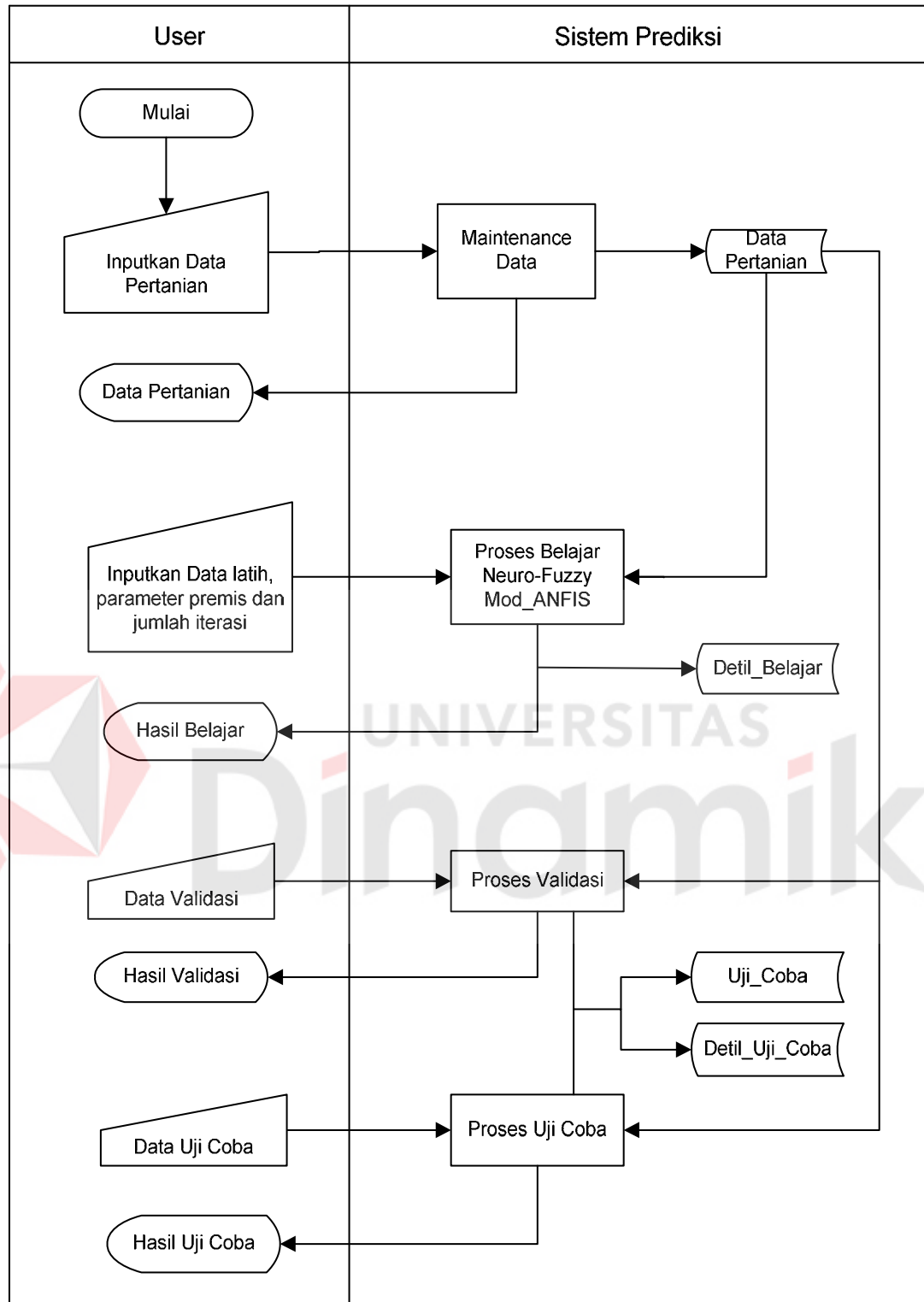
### 3.4 Perancangan Sistem

Sebelum membuat program aplikasi, terlebih dahulu dilakukan proses perancangan sistem. Hal ini dilakukan supaya aplikasi yang dibuat dapat berfungsi sesuai dengan yang diharapkan sehingga mampu menghasilkan prediksi tingkat pendapatan petani yang akurat.

Dalam perancangan sistem ini ada beberapa tahapan-tahapan yang harus dilakukan. Adapun tahapan-tahapan dalam perancangan sistem yang dilakukan adalah pembuatan *System Flow*, *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, dan *Struktur Database*.

#### 3.4.1 System Flow

*System Flow* menunjukkan jalannya program aplikasi secara garis besar. *System flow* sangat membantu dalam pembuatan suatu program aplikasi. Karena selain menunjukkan jalannya program aplikasi dan pengguna, *system flow* juga memperlihatkan *database* yang dibutuhkan oleh aplikasi. *System Flow* yang dibuat dalam aplikasi ini pada gambar 3.6 sebagai berikut:

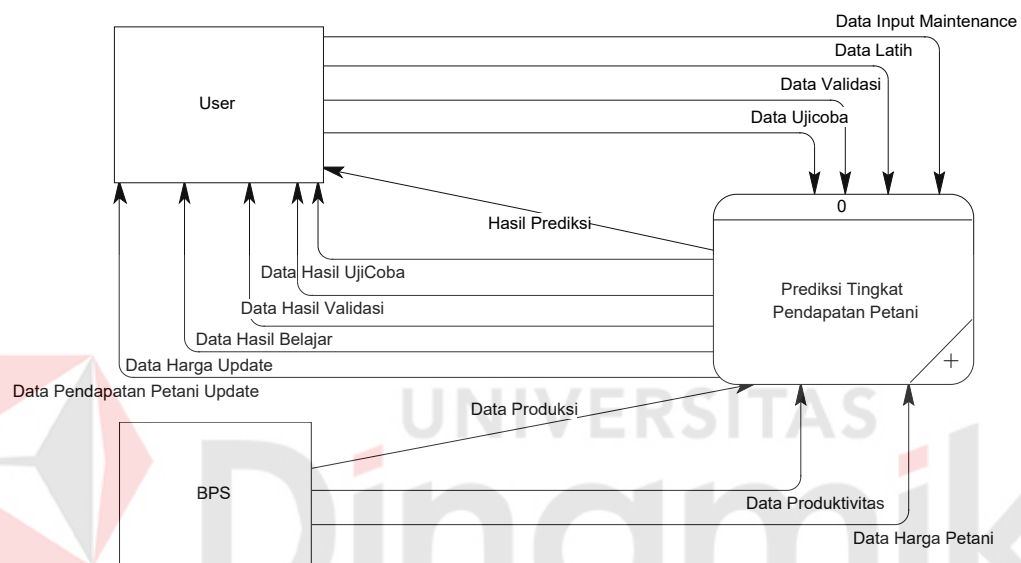


Gambar 3.6. Sistem flow prediksi tingkat pendapatan petani

### 3.4.2 Data Flow Diagram

*Data Flow Diagram* (DFD) menurut aplikasi Penerapan Neuro-Fuzzy Mod\_ANFIS Untuk Prediksi Tingkat Pendapatan Petani terdiri atas *Context Diagram*, DFD level 1 dan DFD Level 2 proses maintenance data.

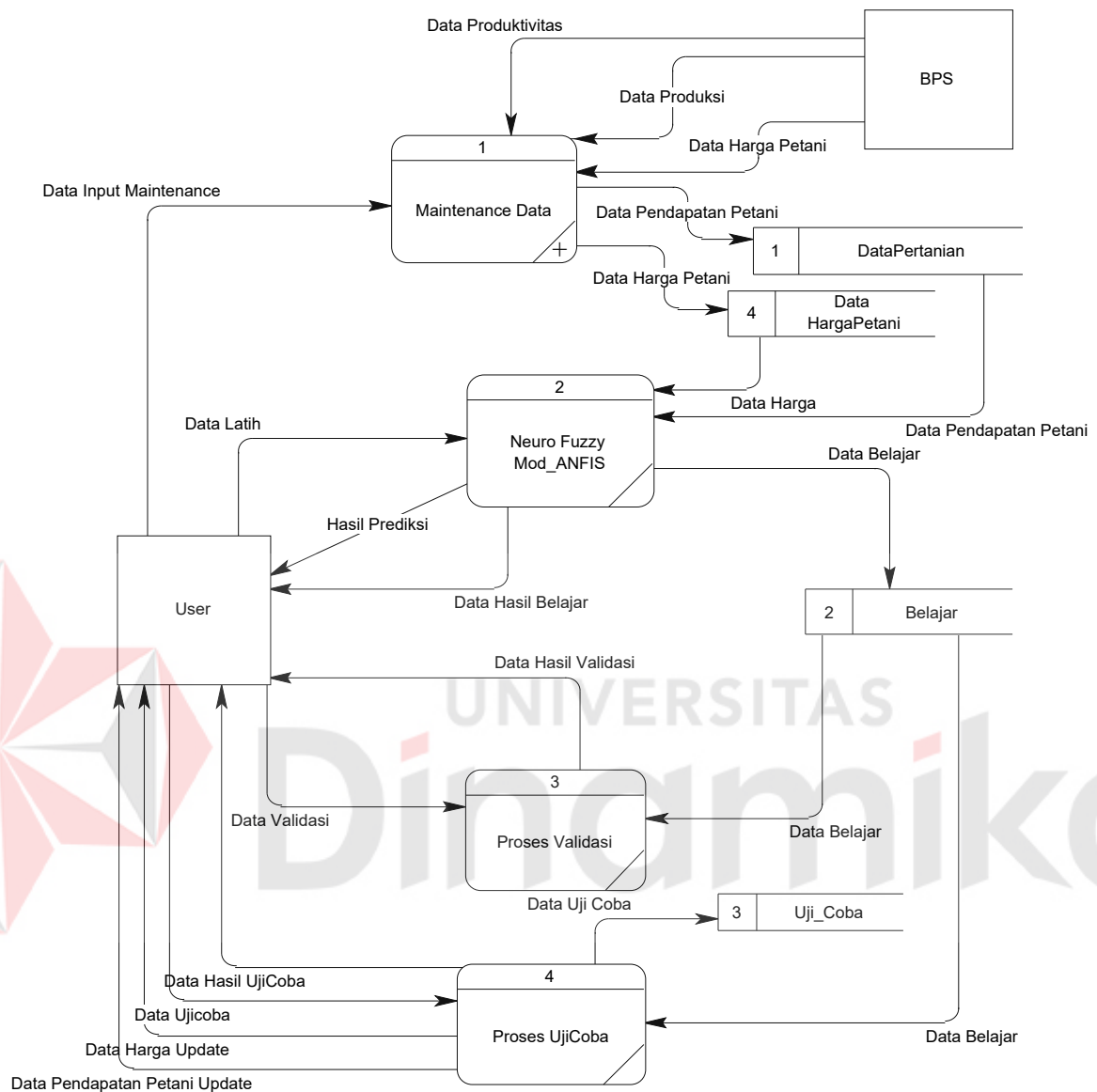
#### A. Context Diagram



Gambar 3.7. Context Diagram

Gambar 3.7 menunjukkan *context diagram*. *Context diagram* merupakan level paling awal dari suatu DFD. Dalam *context diagram* terlihat *entity* yang berperan dalam program aplikasi ini, yaitu *user*. *User* bersifat *universal*, jadi selain sebagai *user* juga sebagai *admin*. Sehingga *user* memegang kendali penuh terhadap sistem. *User* memberikan data ke sistem berupa data latih, data pembelajaran, data validasi beserta data uji coba. Sistem akan memberikan data hasil belajar, data hasil prediksi, data harga update, data pendapatan update dan data validasi kepada *user*.

## B. DFD Level 1

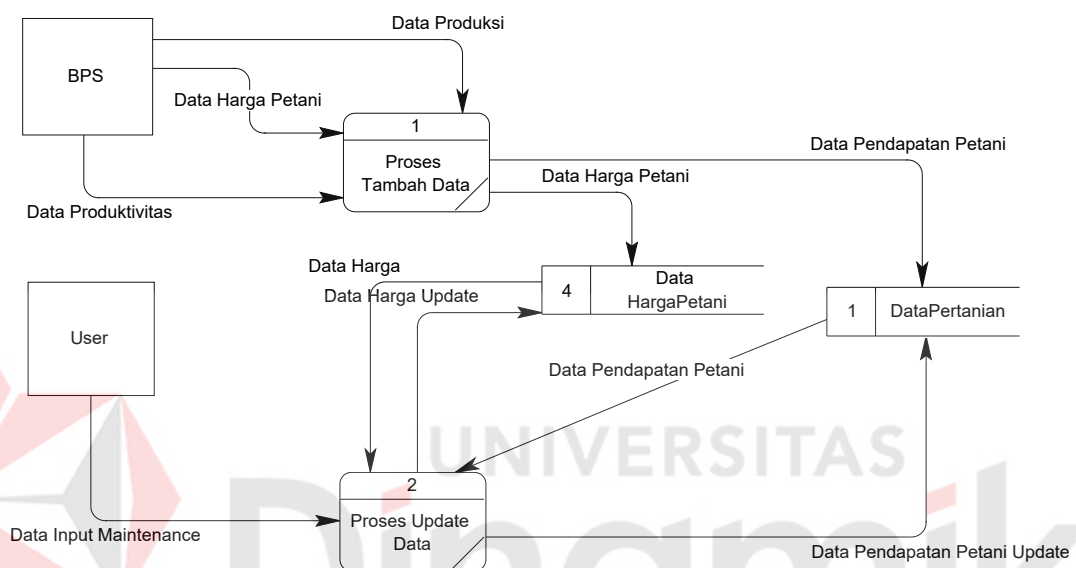


Gambar 3.8. DFD Level 1

Pada DFD Level 1 terdapat proses Maintenance Data, proses Neuro-Fuzzy Mod\_ANFIS, Proses Validasi dan Proses Uji Coba. Dalam proses maintenace data, *user* menginputkan data input maintenance, kemudian sistem akan menyimpan data hasil maintenance. *User* memilih data latihan untuk dilakukan proses belajar Neuro-Fuzzy Mod\_ANFIS dan menghasilkan output berupa data

hasil belajar. Selanjutnya dilakukan proses validasi dengan cara memberikan data latih dan data hasil belajar untuk *output*-nya berupa data hasil validasi. Terakhir adalah proses Uji Coba, user memilih data uji coba yang akan menghasilkan output berupa data data hasil uji coba.

### C. DFD Level 2 Proses Maintenance Data



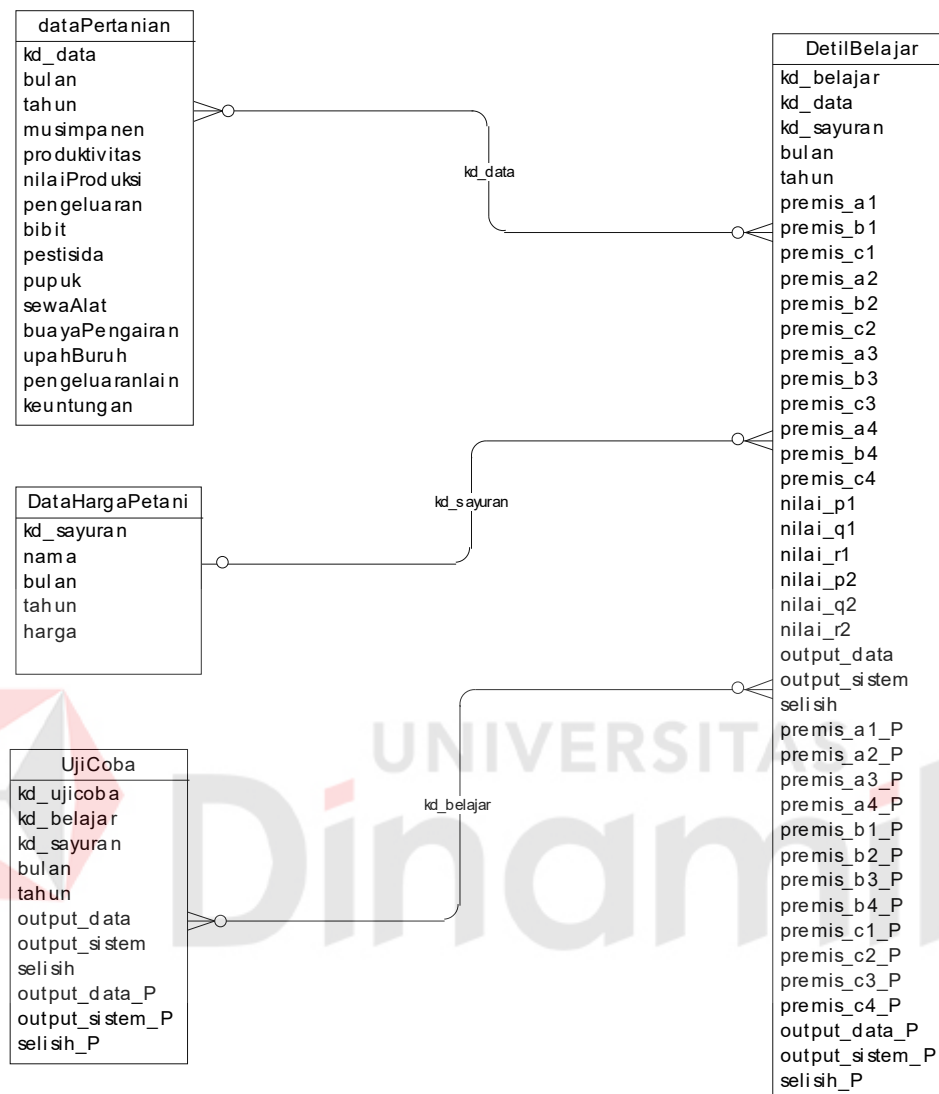
Gambar 3.9. DFD Level 2 Proses Maintenance Data

Pada DFD level 2 Proses Maintenance Data terdapat proses Tambah Data dan proses untuk meng-*update* data.

### 3.4.3 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) merupakan suatu desain sistem yang digunakan untuk merepresentasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan sistem dalam pemrosesan *database*.

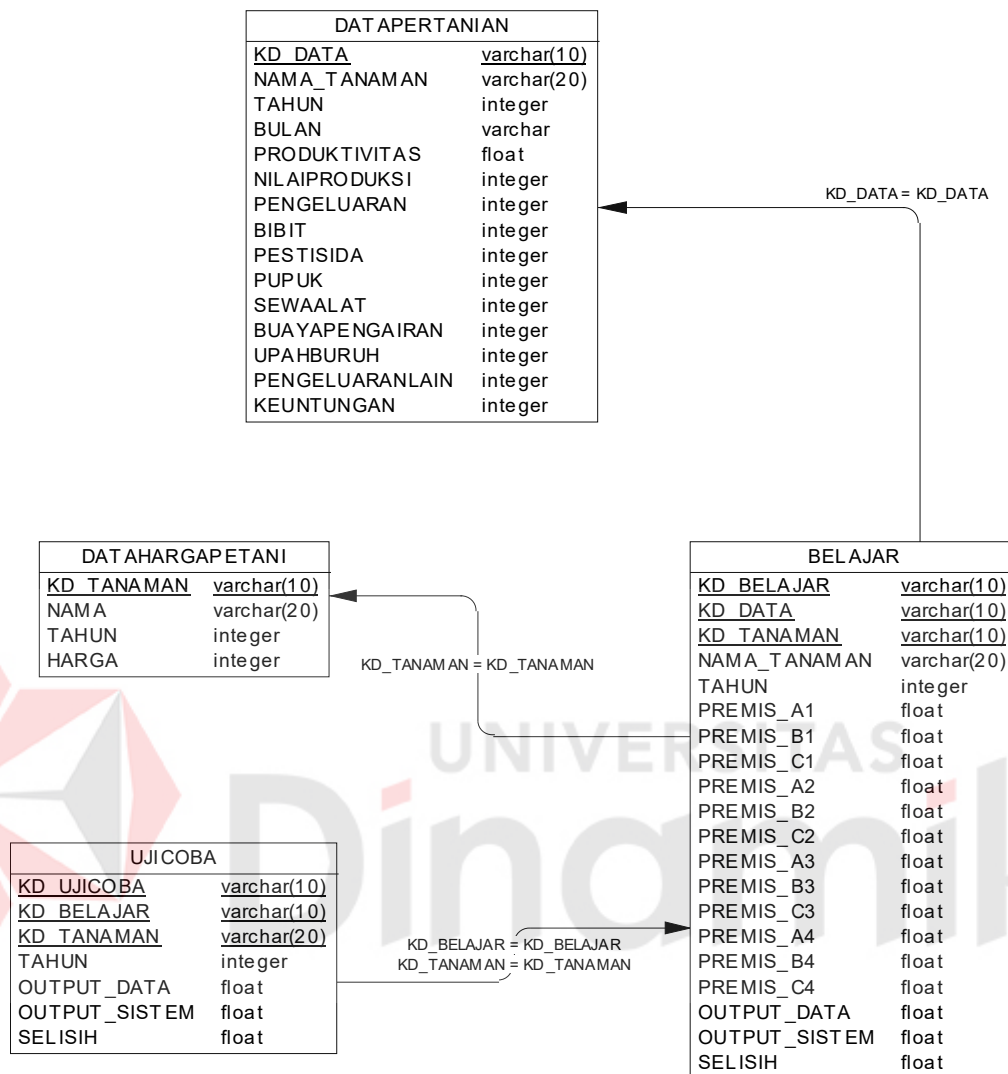
### A. Conceptual Data Model (CDM)



Gambar 3.10. CDM

ERD juga menunjukkan hubungan (relasi) antar tabel. ERD terdiri atas *Conceptual Data Model* (CDM), pada gambar 3.10 dan *Physical Data Model* (PDM) pada gambar 3.11.

## B. Physical Data Model (PDM)



Gambar 3.11. PDM

### 3.4.4 Struktur Database

Tabel-tabel yang digunakan dalam aplikasi ini antara lain:

- Nama Tabel** : DATAHARGAPETANI

**Primary Key** : KD\_TANAMAN

**Foreign Key** : -

**Fungsi** : Untuk menyimpan data harga yang didapat petani.

Tabel 3.1. Tabel datahargapetani

No	Field	Type	Length	Key
1.	KD TANAMAN	Varchar	3	PK
2.	NAMA	Varchar	30	
3.	BULAN	Varchar	30	
4.	TAHUN	Integer		
5.	HARGA	Integer		

2. Nama Tabel : DATAPERTANIAN
- Primary Key : KD\_DATA
- Foreign Key : -
- Fungsi : Untuk menyimpan data produksi

Tabel 3.2. Tabel datapertanian

No	Field	Type	Length	Key
1.	KD_DATA	Varchar	6	PK
2.	NAMA TANAMAN	Varchar	20	
3.	TAHUN	Integer		
4.	BULAN	Varchar	8	
5.	PRODUKTIVITAS	Integer		
6.	NILAIPRODUKSI	Integer		
7.	PENGELUARAN	Integer		
8.	BIBIT	Integer		
9.	PESTISIDA	Integer		
10.	PUPUK	Integer		
11.	SEWAALAT	Integer		
12.	BIAYAPENGAIAN	Integer		
13.	UPAHBURUH	Integer		
14.	PENGELUARANLAIN	Integer		
15.	KEUNTUNGAN	Integer		

3. Nama Tabel : DETILBELAJAR
- Primary Key : KD\_BELAJAR
- Foreign Key : KD\_DATA *references* ke  
DATA PERTANIAN (KD\_DATA)  
KD\_TANAMAN



Fungsi : Untuk menyimpan data belajar

Tabel 3.3. Tabel detilbelajar

No	Field	Type	Length	Key
1.	KD BELAJAR	Varchar	9	PK
2.	KD DATA	Varchar	9	PK,FK
3.	KD TANAMAN	Varchar	9	PK,FK
4.	PREMIS A1	Float	8	
5.	PREMIS B1	Float	8	
6.	PREMIS C1	Float	8	
7.	PREMIS A2	Float	8	
8.	PREMIS B2	Float	8	
9.	PREMIS C2	Float	8	
10.	PREMIS A3	Float	8	
11.	PREMIS B3	Float	8	
12.	PREMIS C3	Float	8	
13.	PREMIS A4	Float	8	
14.	PREMIS B4	Float	8	
15.	PREMIS C4	Float	8	
16.	Nilai F1	Float	8	
17.	Nilai F2	Float	8	
18.	OUTPUT DATA	Float	8	
19.	OUTPUT SISTEM	Float	8	
20.	OUTPUT DATA P	Float	8	
21.	OUTPUT SISTEM P	Float	8	
22.	SELISIH P	Float	8	

4. Nama Tabel : UJICOBA

Primary Key : KD\_UJI\_COBA , KD\_BELAJAR

Foreign Key : KD\_BELAJAR *references* ke  
BELAJAR (KD\_BELAJAR)

Fungsi : Untuk menyimpan data uji coba

Tabel 3.4. Tabel ujicoba

No	Field	Type	Length	Key
1.	KD UJICOBA	Varchar	9	PK
2.	KD BELAJAR	Varchar	9	PK,FK
3.	KD SAYURAN	Varchar	9	PK,FK
4.	BULAN	Varchar	8	
5.	TAHUN	Integer	8	
6.	OUTPUT DATA	Integer	8	

4	OUTPUT SISTEM	Integer	8	
4	SELISIH	Integer	8	

### 3.5 Desain Input/Output

Untuk menjalankan aplikasi Penerapan Neuro-Fuzzy Mod\_ANFIS untuk Presiksi Tingkat Pendapatan Petani ini dibutuhkan beberapa form yang digunakan sebagai sarana untuk melakukan proses prediksi.

#### 3.5.1 Form Maintenance Data

Form Data digunakan untuk meng-*input*-kan data produksi sekaligus *maintenance* data produksi. Tampilan desain dari Form Data terlihat pada Gambar

3.12 Langkah-langkah penggunaan form Data adalah sebagai berikut:

1. Pilih kategori pada combobox paling atas, untuk memilih opsi data yang akan ditambahkan atau dirubah, seperti Data Harga Petani atau Data Produksi Pertanian.
2. Pilih jenis pada combobox dibawahnya, untuk memilih jenis tanaman yang akan ditambahkan atau dirubah, seperti jenis padi atau palawija.
3. Tekan tombol 'Tampil' untuk menampilkan data yang nantinya akan di tampilkan pada grid dibawahnya dan frame di sebelah kanan.
4. Tekan tombol 'Tambah' untuk menyiapkan kursor berada pada frame kanan, untuk nantinya siap di inputkan data baru. Tombol akan berubah label menjadi 'Simpan' untuk proses simpan data.
5. Untuk mengedit terlebih dahulu klik tombol 'Edit' kemudian tekan tombol 'Save' untuk menyimpan.
6. Pembatalan semua proses dapat dilakukan dengan menekan tombol 'Batal'.

FORM MAINTENANCE DATA	
<div style="display: flex; flex-direction: column; align-items: flex-start;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <span style="margin-right: 10px;">Kategori</span> <div style="border: 1px solid black; padding: 2px 10px; flex-grow: 1;">Pilih</div> <div style="width: 20px; height: 20px; border: 1px solid black; margin-left: 5px;"></div> </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <span style="margin-right: 10px;">Jenis</span> <div style="border: 1px solid black; padding: 2px 10px; flex-grow: 1;">Pilih</div> <div style="width: 20px; height: 20px; border: 1px solid black; margin-left: 5px;"></div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <span style="margin-right: 10px;">Tanggal</span> <div style="border: 1px solid black; padding: 2px 10px; flex-grow: 1;">Pilih</div> <div style="width: 20px; height: 20px; border: 1px solid black; margin-left: 5px;"></div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; width: 100px; text-align: center;">TAMPIL</div> <div style="border: 1px solid black; width: 100%; height: 100px; margin-bottom: 10px; display: flex; align-items: center; justify-content: center;">Grid</div> </div>	<div style="border: 1px solid black; padding: 10px; width: 150px;"> <div style="display: flex; margin-bottom: 5px;"> <span style="margin-right: 10px;">Kode</span> <div style="border: 1px solid black; flex-grow: 1;"></div> </div> <div style="display: flex; margin-bottom: 5px;"> <span style="margin-right: 10px;">Nama</span> <div style="border: 1px solid black; flex-grow: 1;"></div> </div> <div style="display: flex;"> <span style="margin-right: 10px;">Nilai</span> <div style="border: 1px solid black; flex-grow: 1;"></div> </div> </div>
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px 20px; margin: 5px;">TAMBAH</div> <div style="border: 1px solid black; padding: 5px 20px; margin: 5px;">EDIT</div> <div style="border: 1px solid black; padding: 5px 20px; margin: 5px;">BATAL</div> <div style="border: 1px solid black; padding: 5px 20px; margin: 5px;">KELUAR</div> </div>	

Gambar 3.12. Tampilan desain Form Maintenance Data

### 3.5.2 Form Proses 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 dari sejumlah iterasi yang dilakukan. Tampilan desain dari form Proses Belajar terlihat pada Gambar

3.13. Langkah-langkah penggunaan form Proses Belajar adalah sebagai berikut:

1. Memilih jenis tanaman dengan cara memilih pada *combo box* jenis tanaman.
2. Meng-*input*-kan banyaknya iterasi yang diinginkan pada input box.
3. Untuk parameter premis dapat di-*input*-kan secara manual ataupun diambil dari sistem. Apabila diinginkan parameter premis dari sistem dapat dilakukan dengan menekan tombol 'Default Premis'.
7. Selanjutnya dilakukan proses belajar dengan menekan tombol 'Proses '.
8. Tombol 'Ujicoba' digunakan untuk menampilkan proses selanjutnya yaitu proses Ujicoba.

FORM PROSES BELAJAR																									
Jenis		<div style="border: 1px solid black; padding: 2px; display: inline-block;">Pilih</div>																							
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px; text-align: center;">a1</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">b1</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">c1</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">a3</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">b3</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">c3</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px; text-align: center;">a2</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">b2</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">c2</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">a4</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">b4</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">c4</td> </tr> </table>						a1	b1	c1	a3	b3	c3	a2	b2	c2	a4	b4	c4	<div style="border: 1px solid black; padding: 5px; text-align: center;">Iterasi</div>		<div style="border: 1px solid black; padding: 5px; text-align: center;">in1</div>		<div style="border: 1px solid black; padding: 5px; text-align: center;">In2</div>		<div style="border: 1px solid black; padding: 5px; text-align: center;">Out1</div>	
a1	b1	c1	a3	b3	c3																				
a2	b2	c2	a4	b4	c4																				
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">DEFAULT PREMIS</div> <div style="border: 1px solid black; padding: 2px 10px;">HAPUS</div> </div>						<div style="border: 1px solid black; padding: 5px; text-align: center;">Nilai Teta</div>																			
<div style="border: 1px solid black; padding: 5px; text-align: center;">Keluaran Jaringan</div>						<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 5px; text-align: center;">RMSE</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">MAPE</td> </tr> <tr> <td style="padding: 5px;">RMSE</td> <td style="padding: 5px;">MAPE</td> </tr> </table>				RMSE	MAPE	RMSE	MAPE												
RMSE	MAPE																								
RMSE	MAPE																								
PROSES		UJI COBA		DATA		KELUAR																			

Gambar 3.13. Tampilan desain form Proses Belajar

### 3.5.3 Form Uji Coba dan Validasi

Form Uji Coba dan Validasi digunakan untuk melakukan uji coba atau validasi dengan menggunakan parameter premis hasil belajar. Tampilan desain dari Form Uji Coba dan Validasi seperti pada Gambar 3.14. Langkah-langkah penggunaan form Proses Belajar adalah sebagai berikut:

1. Memilih jenis proses Validasi atau Ujicoba, dengan cara memilih pada *combo box* paling atas .
2. Memilih Jenis Uji Coba yang akan dilakukan.
3. Tekan Tombol ‘Proses’ untuk proses uji coba.
4. Untuk membuat laporan *User* dapat menekan tombol ‘Report’.

5. Tombol ‘Hasil’ digunakan untuk menampilkan form detil yang didalamnya terdapat proses perhitungan nilai produksi, biaya-biaya, serta keuntungan yang diraih.

**Validasi Dan Uji Coba**

Pilih

Pilih

**Parameter Premis**

Grid

**Data Aktual Dan Keluaran Sistem**

Grafik

**Error Keluaran Sistem**

Grafik

RMSE

MAPE

Proses Report Hasil Keluar

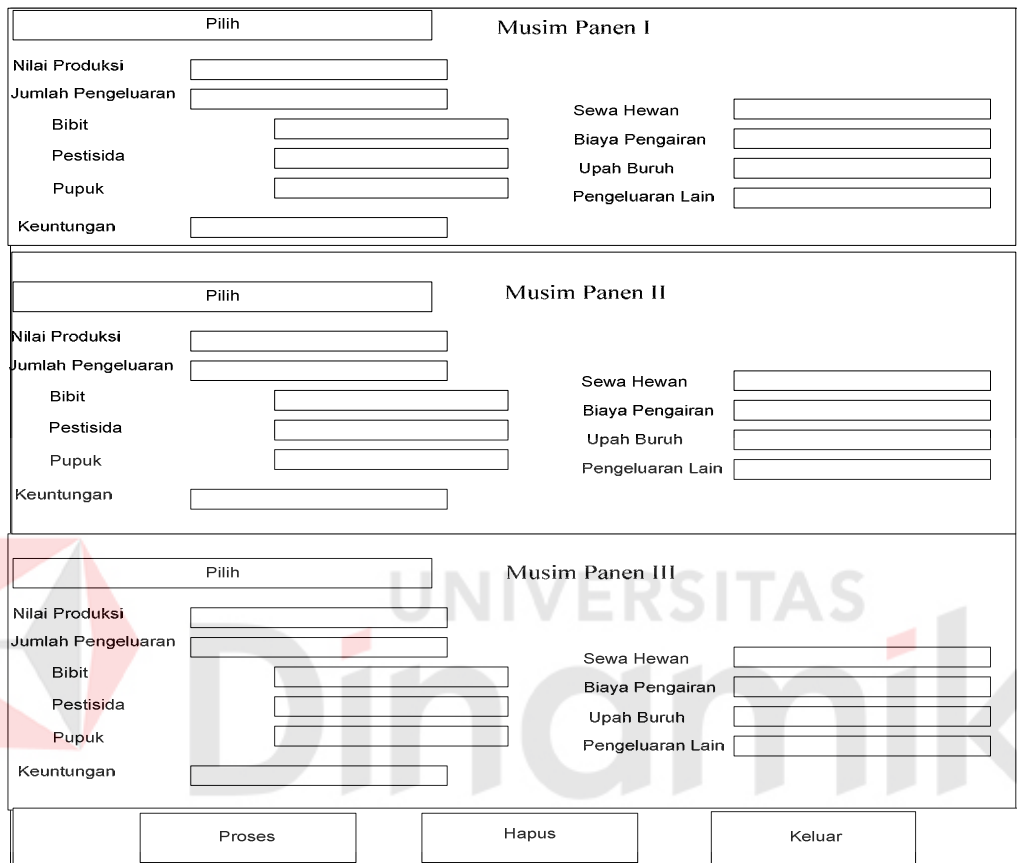
Gambar 3.14. Tampilan desain form Uji Coba dan Validasi

### 3.5.4 Form Detil

Form Detil digunakan untuk melakukan perhitungan nilai produksi, biaya-biaya, serta keuntungan yang diraih oleh petani pada sistem prediksi ini. Tampilan desain dari Form Detil seperti pada Gambar 3.15. Langkah-langkah penggunaan form Proses Belajar adalah sebagai berikut:

1. Memilih jenis tanaman yang akan di uji coba dengan cara memilih pada *combo box* tiap-tiap musim panen.

2. Tekan ‘Hapus’ apabila ingin me-reset ulang perhitungan atau mengganti jenis tanaman .
3. Tekan Tombol ‘Proses’ untuk melakukan proses penghitungan.



Pilih		Musim Panen I	
Nilai Produksi	<input type="text"/>		
Jumlah Pengeluaran	<input type="text"/>		
Bibit	<input type="text"/>	Sewa Hewan	<input type="text"/>
Pestisida	<input type="text"/>	Biaya Pengairan	<input type="text"/>
Pupuk	<input type="text"/>	Upah Buruh	<input type="text"/>
Keuntungan	<input type="text"/>	Pengeluaran Lain	<input type="text"/>

Pilih		Musim Panen II	
Nilai Produksi	<input type="text"/>		
Jumlah Pengeluaran	<input type="text"/>		
Bibit	<input type="text"/>	Sewa Hewan	<input type="text"/>
Pestisida	<input type="text"/>	Biaya Pengairan	<input type="text"/>
Pupuk	<input type="text"/>	Upah Buruh	<input type="text"/>
Keuntungan	<input type="text"/>	Pengeluaran Lain	<input type="text"/>

Pilih		Musim Panen III	
Nilai Produksi	<input type="text"/>		
Jumlah Pengeluaran	<input type="text"/>		
Bibit	<input type="text"/>	Sewa Hewan	<input type="text"/>
Pestisida	<input type="text"/>	Biaya Pengairan	<input type="text"/>
Pupuk	<input type="text"/>	Upah Buruh	<input type="text"/>
Keuntungan	<input type="text"/>	Pengeluaran Lain	<input type="text"/>

Proses	Hapus	Keluar
--------	-------	--------

Gambar 3.15. Tampilan desain form Detil

### 3.5.5 Form Report

Form Report digunakan untuk melakukan preview data hasil prediksi dan mencetak dengan printer. Tampilan desain dari Form Report seperti pada Gambar

3.16. Langkah-langkah penggunaan form Proses Report adalah sebagai berikut:

1. Memilih jenis kategori yang akan di cetak, berupa Data Produksi Pertanian atau Data Harga Petani dengan cara memilih pada *combo box* kategori posisi paling atas.

2. Memilih tombol 'Preview' untuk melihat tampilan data yang akan dicetak.
3. Tekan Tombol 'Print' untuk mencetak.

Kategori	<input style="width: 100%;" type="text" value="Pilih"/>			
Jenis	<input style="width: 100%;" type="text" value="Pilih"/>			
Tanggal	<input style="width: 100%;" type="text"/>			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px 20px;">PREVIEW</td> <td style="padding: 5px 20px;">PRINT</td> <td style="padding: 5px 20px;">KELUAR</td> </tr> </table>		PREVIEW	PRINT	KELUAR
PREVIEW	PRINT	KELUAR		

Gambar 3.16. Tampilan desain form Report

### 3.6 Perancangan Data

Susunan pasangan data latih digunakan untuk proses belajar berdasarkan data sekunder yang diperoleh dari BPS, disusun dengan format

$$[x(t-24); \dots; x(t-1); x(t); x(t+1); \dots; x(t+12)]$$

Keterangan :

$$x(t-24) = \text{input-an 1}$$

$$x(t) = \text{input-an 2}$$

$$x(t+12) = \text{output atau data aktual}$$

Data produksi yang digunakan dalam proses belajar adalah periode musim panen per empat bulan, tahun 1994 sampai tahun 1997 (*input-an 1*) tahun 1998 sampai tahun 2001 (*input-an 2*) sebagai masukan jaringan. Data tahun selebihnya hingga tahun 2005 (*input-an 3*) sebagai *output*. Data dari hasil proses belajar berupa parameter premis yang telah diperbaharui akan dipergunakan untuk melakukan proses validasi dan uji coba.

## **BAB IV**

### **IMPLEMENTASI DAN EVALUASI**

#### **4.1 Instalasi Program**

Sebelum mengimplementasikan dan menjalankan aplikasi Penerapan Neuro-Fuzzy Struktur Mod\_ANFIS Untuk Prediksi Tingkat Pendapatan Petani, terlebih dahulu komponen-komponen utama komputer yang mendukung setiap proses harus sudah terpasang.

##### **4.1.1 Kebutuhan perangkat keras**

Untuk dapat menjalankan aplikasi Penerapan Neuro-Fuzzy Struktur Mod\_ANFIS Untuk Prediksi Tingkat Pendapatan Petani ini dibutuhkan persyaratan minimal sebagai berikut:

- a. CPU Pentium II 333 Mhz atau lebih
- b. Memory minimal 32 MB
- c. Harddisk minimal 4,3 GB
- d. VGA Card 4 MB
- e. Monitor SVGA dengan resolusi 800 x 600
- f. Keyboard, mouse dan printer

##### **4.1.2 Kebutuhan perangkat lunak**

Kebutuhan perangkat lunak aplikasi Penerapan Neuro-Fuzzy Struktur Mod\_ANFIS Untuk Prediksi Tingkat Pendapatan Petani ini adalah:

- a. Microsoft Windows XP
- b. Power Designer



- d. Borland Delphi 6.0
- c. Microsoft SQL Server 7

## 4.2 Implementasi Sistem

Setelah semua komponen yang mendukung aplikasi Penerapan Neuro-Fuzzy Struktur Mod\_ANFIS Untuk Prediksi Tingkat Pendapatan Petani terpasang, proses selanjutnya bagaimana mewujudkan rancangan sistem ke dalam sebuah program Prediksi Tingkat Pendapatan Petani. Berikut ditunjukkan modul/bagian yang penting dalam implementasi sistem.

### 4.2.1 Proses Belajar

Berdasarkan alur proses belajar pada Gambar 3.1 di muka, maka proses yang terjadi dapat dijabarkan sebagai berikut:.

#### A. Inputkan Data Latih

Data yang digunakan sebagai data latih dimasukkan ke dalam variable array dengan tujuan mempermudah pengaksesan data.

```

type
    in1, in2, out1 : Array[1..12] of real;
    Input1, Input2, Output : real;
    maks_iterasi : integer;
    Tgl : TDateTime;
end;

var
    DL : array of DataLatih;

while not adoquery3.Eof do
begin
    in1[i]:=adoquery1.FieldName('harga').AsFloat;
    in2[i]:=adoquery2.FieldName('harga').AsFloat;
    out1[i]:=adoquery3.FieldName('harga').AsFloat;

    inc(i);
    Next;
end;

```

Proses selanjutnya mengolah susunan pasangan data latih beserta jumlah iterasi dimasukkan ke dalam sistem. Implementasi proses ini dalam program adalah sebagai berikut:

```
maks_iterasi:=StrToInt(edIterasi.Text);
```

Keterangan:

Sistem mengambil data belajar terdiri dari input1, input2 dan output. Kemudian ditampung dalam variable array in1, in2 dan out1.

## B. 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 dulu ditentukan parameter fungsi keanggotaan ( $a1..a4$ ,  $b1..b4$ ,  $c1..c4$ ), 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:

```
Bantu:=abs((in1[i]-(c1))/(a1));
```

```
Bantu:=exp(ln(Bantu)*2*(b1))+1;
```

Nilai parameter fungsi keanggotaan dapat diisikan secara manual atau langsung mengambil dari sistem. Nilai dari sistem diperoleh dengan menggunakan bantuan program Matlab. Untuk semua keluaran simpul pada tahap maju ini diberi simbol 'a', sehingga pada lapisan 1 ini diperoleh keluaran

simpul n1a sampai dengan 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 jika diterapkan logika fuzzy AND sebagai fungsi simpul, maka implementasinya dalam program adalah sebagai berikut:

```
n5a := Min(n1a, n3a);
n6a := Min(n2a, n4a);
```

Penerapan logika fuzzy AND pada lapisan ini adalah memilih nilai yang terkecil dari simpul-simpul 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 := n5a+n6a;
if ntot_a=0 then
begin
  n7a := 0;
  n8a := 0;
end
else
begin
  n7a :=n5a/ntot_a;
  n8a :=n6a/ntot_a;
end;
```

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

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 :

```

A :=(n7a*in1) (n7a*in2) n7a (n8a*in1) (n8a*in2) n8a]
A[i,1] := n7a*in1; A[i,2] := n7a*in2; A[i,3] := n7a;
A[i,4] := n8a*in1; A[i,5] := n8a*in2; A[i,6] := n8a;

// transpose matrik
for i:= 1 to 6 do
  begin
    for j:= 1 to 12 do
      begin
        ATranspose[i,j]:= A[j,i];
      end;
    end;
  end;

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

Invers(AHasil,7,AInvers);
for i:=1 to 6 do
  begin
    for j:=1 to 12 do
      begin
        bantu:=0;
        for k:=1 to 6 do
          begin
            bantu:=bantu + AInvers[i,k]*ATranspose[k,j];
          end;
        AHasil[i,j]:=bantu;
      end;
    end;
  end;

//Proses mencari teta= AHasil*Out1
for i:= 1 to 6 do
  begin
    bantu:=0;
    for j:= 1 to 12 do
      begin
        bantu:=bantu+ AHasil[i,j]*out1[j];
      end;
  end;

```

```

        Teta[i]:=bantu;
    end;
for i:= 1 to 6 do
    begin
        mTeta.Lines.Append(FloatToStr(Teta[i]));
    end;
    p1:=Teta[1];
    q1:=Teta[2];
    r1:=Teta[3];
    p2:=Teta[4];
    q2:=Teta[5];
    r2:=Teta[6];

```

Proses LSE ini akan menghasilkan parameter konsekuen (p1, q1, r1, p2, q2 dan r2). Kemudian hasilnya digunakan untuk mencari nilai f1 dan f2.

```

f1[i]:=(p1*in1[i])+(q1*in2[i])+r1);
f2[i]:=(p2*in1[i])+(q2*in2[i])+r2);

n9a  := n7a*f1;
n10a := n8a*f2;

```

Hasil defuzzyfikasi berupa simpul n9a dan n10a, yang didapat dengan cara mengalikan simpul n7a, n8a dengan nilai array f1 dan f2.

##### 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 :

```

N11a = n9a+n10a;
error := output-n11a;

```

Hasil dari proses lapisan ini adalah n11a yang merupakan keluaran jaringan., sekaligus dilakukan perhitungan untuk mencari besar *error*. Sampai di sini selesai tahap maju dari sistem aplikasi Penerapan Neuro-Fuzzy Struktur Mod\_ANFIS untuk Prediksi Tingkat Pendapatan Petani diteruskan ke tahap mundur.

### C. 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:

```
e11 := -2*(Output-n11a);
d11 := -e11/2;

ftot:= f1+f2;

d9  := d11*f1/ftot;
d10 := d11*f2/ftot;

d7  := d9/f1;
d8  := d10/f2;

ntot_b := ntot_a;

n5b1 := (n7a+d7)*ntot_b;
n6b1 := (n8a+d8)*ntot_b;

n1b := n5b1; n3b := n5b1;
n2b := n6b1; n4b := n6b1;
```

Proses selanjutnya perubahan parameter fungsi keanggotaan pada layer 1 tahan mundur.

Implementasi dalam program sebagai berikut:

```
da1[i]:=(2*b1_lama/a1_lama)*n1b[i]*(1-n1b[i]);
tampung_a1[i]:=a1_lama+da1[i];
if in1[i]=c1_lama then
begin
    db1[i]:=0;
    dc1[i]:=0;
    tampung_b1[i]:=b1_lama;
    tampung_c1[i]:=c1_lama;
end
else
begin
```

```

db1[i] := (-2) * ln(abs((in1[i] - c1_lama) / a1_lama)) * n1b[i] * (1 - n1b[i]);
dc1[i] := (2 * b1_lama / (in1[i] - c1_lama)) * n1b[i] * (1 - n1b[i]);
tampung_b1[i] := b1_lama + db1[i];
tampung_c1[i] := c1_lama + dc1[i];

end;

```

Perhitungan turunan fungsi keanggotaan diterapkan dari  $da1$  sampai  $dc4$ .

Hasil dari lapisan 1 tahap mundur adalah parameter premis yang telah diperbaharui. 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 berdasarkan parameter yang telah diperbaharui melalui proses belajar, untuk datanya berupa pasangan data latih.

Pada proses validasi hanya digunakan proses tahap maju dari algoritma *Neuro-Fuzzy*, tidak ada proses mundur untuk koreksi kesalahan.

Berdasarkan alur proses validasi pada Gambar 3.3 di muka, maka implementasi dalam program sebagai berikut:

```

in1[i] := adoquery1.FieldByName('produktivitas').AsFloat;
in2[i] := adoquery2.FieldByName('produktivitas').AsFloat;
out1[i] := adoquery3.FieldByName('produktivitas').AsFloat;

a1[i] := adoquerydetilbelajar.fieldbyname('premis_a1').asFloat;
b1[i] := adoquerydetilbelajar.fieldbyname('premis_b1').asFloat;
c1[i] := adoquerydetilbelajar.fieldbyname('premis_c1').asFloat;

p1 := adoquerydetilbelajar.fieldbyname('nilai_p1').AsFloat;
q1 := adoquerydetilbelajar.fieldbyname('nilai_q1').AsFloat;
r1 := adoquerydetilbelajar.fieldbyname('nilai_r1').AsFloat;
p2 := adoquerydetilbelajar.fieldbyname('nilai_p2').AsFloat;
q2 := adoquerydetilbelajar.fieldbyname('nilai_q2').AsFloat;
r2 := adoquerydetilbelajar.fieldbyname('nilai_r2').AsFloat;

f1[i] := FieldByName('Nilai_F1').AsFloat;
f2[i] := FieldByName('Nilai_F2').AsFloat;

n9b := round(n7b * f1);
n10b := round(n8b * f2);

```

Keterangan:

proses yang terjadi sama dengan proses belajar, perbedaannya hanya dilakukan pada proses tahap maju. Parameter premis diambil dari hasil proses belajar begitu juga pada lapisan 4 LSE diambilkan dari sistem berupa nilai  $f_1$  dan  $f_2$ .

#### 4.2.3 Proses Uji Coba

Setelah proses belajar sistem akan diuji coba untuk memprediksi pendapatan petani. Proses uji coba adalah menguji sistem berdasarkan parameter fungsi keanggotaan yang telah diperbaharui melalui proses pembelajaran dengan cara diberi inputan berupa data uji. Pada proses uji coba hanya digunakan tahap maju dari algoritma *Neuro-Fuzzy*.

Untuk proses uji coba, keluaran jaringan adalah hasil prediksi untuk tahun 2005. Selanjutnya untuk menguji kebenaran hasil prediksi dibandingkan dengan data tahun 2005 yang sudah disiapkan. Proses yang terjadi dalam form uji coba sama dengan proses validasi. Perbedaannya hanya terletak pada data yang digunakan dan dilakukannya proses LSE seperti pada saat awal belajar proses belajar.

#### 4.3 Pengujian Sistem

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



#### 4.3.1 Menu Utama

Form Menu Utama digunakan untuk mengakses form-form yang ada pada aplikasi Prediksi Tingkat Pendapatan Petani. Tampilan form Utama dapat dilihat pada Gambar 4.1.



Gambar 4.1. Form Menu Utama

#### 4.3.2 Form Data


Form Data digunakan untuk *maintenance* data produksi pertanian. Tampilan form Data dapat dilihat pada Gambar 4.2. Form *Maintenance* data produksi pertanian berfungsi untuk meng-*input*-kan data produksi sekaligus melakukan proses *update* data.


**FORM MAINTENANCE DATA**

Kategori: Data Produksi Pertanian ▼

Jenis: jagung ▼

Tanggal: 01/01/2004 ▼

 TAMPIL





	produktivitas	nilai produksi	pengeluaran	bibit	pestisida	pup
▶	20,9	445606	110662	12031	3576	
▶	20,9	445606	110662	12031	3576	
▶	20,9	445606	110662	12031	3576	
▶	20,9	445606	110662	12031	3576	
▶	21,75	451008	115486	12554	3480	
▶	21,75	451008	115486	12554	3480	
▶	21,75	451008	115486	12554	3480	
▶	21,75	451008	115486	12554	3480	
▶	20,73	448922	114588	12758	3593	


KODE:

NAMA:

NILAI:

 TAMBAH

 EDIT

 KELUAR

Gambar 4.2. Form Data

### 4.3.3 Proses Belajar

Form Proses Belajar digunakan untuk proses belajar Neuro-Fuzzy struktur Mod\_ANFIS dari sejumlah pasangan data latih. Output dari proses belajar akan menghasilkan parameter premis yang telah diperbarui. Dimana parameter-parameter premis ini memiliki *error* yang minimum dari sejumlah iterasi yang dilakukan. Form proses belajar seperti pada Gambar 4.3, menampilkan beberapa variabel-variabel penting yang ada dalam perhitungan Neuro-Fuzzy struktur Mod\_ANFIS, mulai dari parameter premis *default*, hasil dari *Least Square Estimator*, nilai output jaringan hingga kesalahan pada jaringan, atau yang disebut dengan *Error* jaringan.

Gambar 4.3. Form Belajar

Pasangan data latih yang digunakan untuk proses belajar adalah tahun 1994 hingga tahun 2001 sebagai *input*-an sedangkan tahun berikutnya hingga tahun 2005 digunakan sebagai pengontrol jaringan atau sebagai *output*. Hasil dari proses belajar dapat dilihat pada Tabel 4.1.

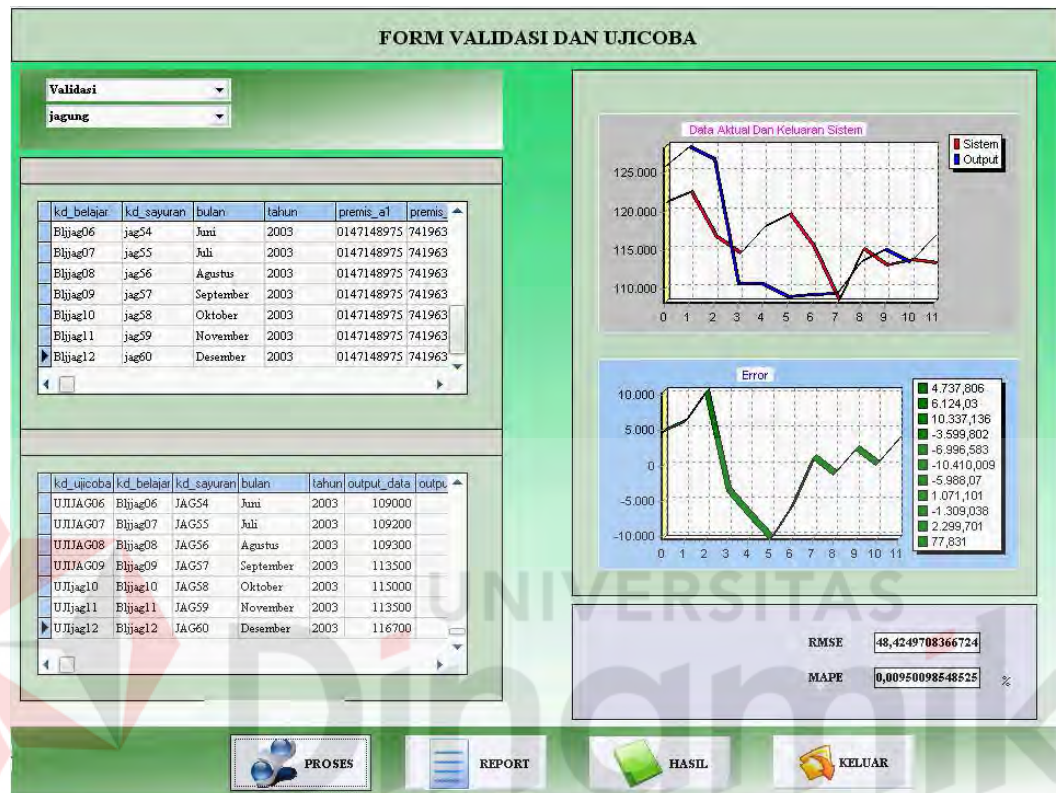
Tabel 4.1. Hasil Proses Belajar

TANAMAN	Tahun	RMSE	MAPE (%)
PADI	2005	141,392	0,0237
JAGUNG	2005	17,5053	0,0034
KEDELA	2005	96,9795	0,0079
KETELA POHON	2005	5,0111	0,0043
KETELA RAMBAT	2005	9,0169	8,0505
KACANG TANAH	2005	116,991	0,0045



#### 4.3.4 Form Validasi

Form Validasi digunakan untuk mengevaluasi hasil proses belajar yang telah dilakukan. Gambar 4.4 menunjukkan proses validasi.



Gambar 4.4. Form Validasi

Hasil dari proses validasi dapat dilihat pada Tabel 4.2.

Tabel 4.2. Hasil proses validasi

TANAMAN	Tahun	RMSE	MAPE (%)
PADI	2005	141,392	0,0237
JAGUNG	2005	17,5053	0,0034
KEDELA	2005	96,9795	0,0079
KETELA POHON	2005	5,0111	0,0043
KETELA RAMBAT	2005	9,0169	8,0505
KACANG TANAH	2005	116,991	0,0045

Hasil proses validasi pada Tabel 4.2 sama dengan hasil proses belajar pada Tabel 4.1. Hal ini menunjukkan proses yang dilakukan sistem *valid*.

#### 4.3.5 Form Report

User dapat membuat *report* hasil prediksi dengan cara menekan tombol 'Report' pada Form Uji Coba dan Validasi kemudian akan ditampilkan Form Report seperti pada Gambar 4.5.



Gambar 4.5. Pemilihan Report

User dapat menampilkan report sesuai dengan tanggal yang diinginkan, yaitu dengan cara memilih tanggal mulai sampai dengan tanggal akhir. Tombol 'Preview' digunakan untuk menampilkan hasil report di layar seperti pada Gambar 4.6. User juga bisa langsung mencetak dengan cara menekan tombol 'Print'.



HASIL PREDIKSI					
JENIS TANAMAN :		JAGUNG			
BULAN	TAHUN	DATA AKTUAL	PREDIKSI	SELISIH	
Januari	2003	125700	121019	4681	
Februari	2003	128300	122246	6054	
Maret	2003	126700	116472	10228	
April	2003	110700	113878	-3178	
Mei	2003	110700	117804	-7104	
Juni	2003	109000	119486	-10486	

Gambar 4.6. Preview Report

#### 4.3.6 Form Detil

Form Detil digunakan untuk melakukan perhitungan nilai produksi, serta perhitungan nilai pengeluaran dan keuntungan bersih. Gambar 4.7 menunjukkan Form Detil.

MUSIM PANEN KE-1	
NILAI PRODUKSI	3111536.48
JUMLAH PENGELUARAN	2620800
BIBIT	490736.48
PESTISIDA	490736.48
PUPUK	490736.48
KEUNTUNGAN	490736.48
SEWA ALAT	490736.48
BIAYA PENGAIRAN	490736.48
UPAH BURUH	490736.48
PENGELUARAN LAIN	490736.48

MUSIM PANEN KE-2	
NILAI PRODUKSI	3312936.9
JUMLAH PENGELUARAN	2620800
BIBIT	692136.9
PESTISIDA	692136.9
PUPUK	692136.9
KEUNTUNGAN	692136.9
SEWA ALAT	692136.9
BIAYA PENGAIRAN	692136.9
UPAH BURUH	692136.9
PENGELUARAN LAIN	692136.9

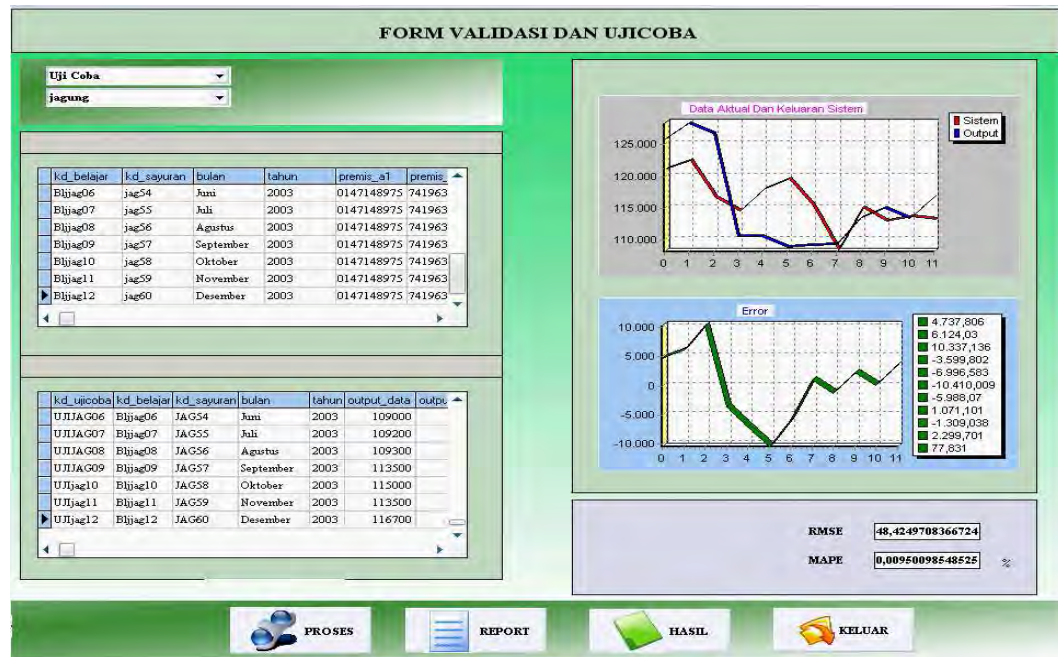
MUSIM PANEN KE-3	
NILAI PRODUKSI	3633345.72
JUMLAH PENGELUARAN	2620800
BIBIT	1012545.72
PESTISIDA	1012545.72
PUPUK	1012545.72
KEUNTUNGAN	1012545.72
SEWA ALAT	1012545.72
BIAYA PENGAIRAN	1012545.72
UPAH BURUH	1012545.72
PENGELUARAN LAIN	1012545.72

Buttons: PROSES, HAPUS, KELUAR

Gambar 4.7. Form Detil

#### 4.3.7 Form Uji Coba

Form Uji Coba digunakan untuk melakukan uji coba dengan menggunakan parameter premis yang telah diperbaharui. Gambar 4.8 menunjukkan proses uji coba. Form Uji Coba menggunakan variabel-variabel dan input hasil dari proses belajar, dan parameter premis yang juga didapat dari proses pembelajaran.



Gambar 4.8. Form uji coba

Pada proses uji coba yaitu, keluaran jaringan adalah hasil prediksi untuk tahun 2005. Selanjutnya untuk menguji kebenaran hasil prediksi dibandingkan dengan data tahun 2005 yang sudah disiapkan.

Tabel 4.3. Kesalahan prediksi uji coba sistem

TANAMAN	Tahun	RMSE	MAPE (%)
PADI	2005	180,937	4,1501
JAGUNG	2005	112,356	0,0015
KEDELAI	2005	237,994	0,0049
KETELA POHON	2005	11,3308	0,0012
KETELA RAMBAT	2005	50,246	0,0987
KACANG TANAH	2005	30,6249	0,0108

#### 4.4 Hasil Prediksi

Dari hasil prediksi yang dilakukan sistem dapat dilihat untuk Padi pada tahun 2005 dengan perhitungan RMSE sebesar 141,392 untuk MAPE 0,0237 %, Jagung pada tahun 2005 dengan perhitungan RMSE sebesar 17,5053 untuk

MAPE 0,034 %, Kedelai pada tahun 2005 dengan perhitungan RMSE sebesar 96,9795 untuk MAPE 0,0079 %, Ketela Pohon pada tahun 2005 dengan perhitungan RMSE sebesar 5,0111 untuk MAPE 0,0043 %, Ketela Rambat pada tahun 2005 dengan perhitungan RMSE sebesar 9,0169 untuk MAPE 8,0505 %, Kacang Tanah pada tahun 2005 dengan perhitungan RMSE sebesar 116,030 untuk MAPE 0,0045 %.

#### 4.5 Evaluasi Sistem

Setelah dilakukan pengujian sistem, kemudian dilakukan evaluasi sistem. Tahap ini mengevaluasi hasil pengujian sistem dengan cara membandingkannya dengan metode lain dan mengukur kesalahan prediksi dengan ukuran ketepatan metode peramalan.

##### 4.5.2 Membandingkan Hasil Prediksi

Untuk membuktikan hasil prediksi menggunakan Neuro-Fuzzy struktur Mod\_ANFIS mempunyai nilai *error* yang minimum, maka hasil prediksi dibandingkan dengan prediksi yang dihasilkan oleh Deptan. Hasil perbandingan dapat dilihat pada Tabel 4.4.

Tabel 4.4. Perbandingan hasil prediksi

Jenis Tanaman	Metode	2005
		MAPE
Padi	A	0,0237
	B	0,0094
Jagung	A	0,034
	B	8,846
Kedelai	A	0,0079
	B	0,916
Ketela Pohon	A	0,0043
	B	0,467

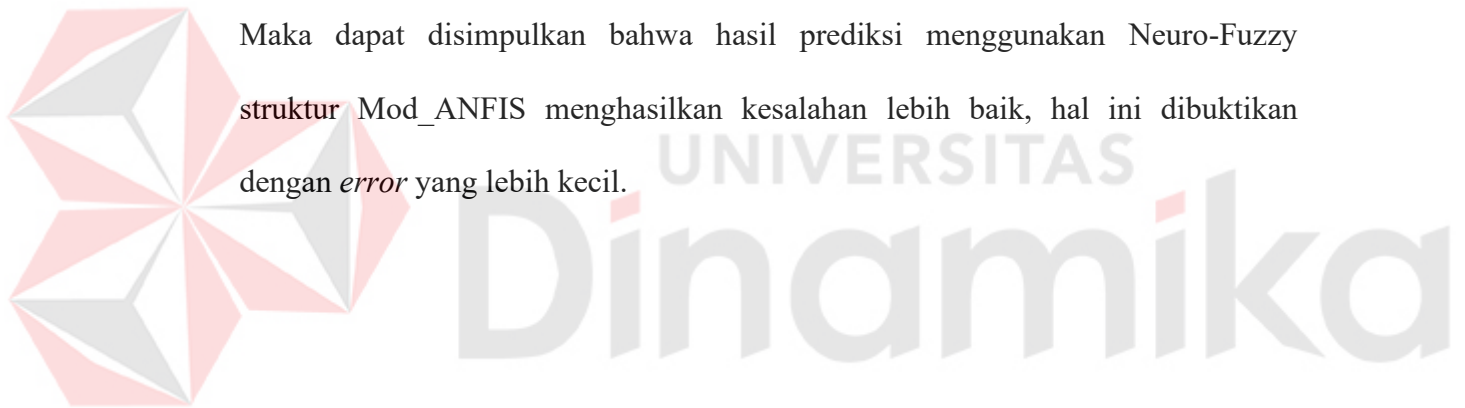


Ketela Rambut	A	8,0505
	B	0,925
Kacang Tanah	A	0,0045
	B	9,805

Keterangan:

1. Metode A = Neuro-Fuzzy struktur Mod\_ANFIS.
2. Metode B = Peramalan oleh Deptan.
3. MAPE dalam persen (%)

Berdasarkan Tabel 4.4 dapat dilihat bahwa hasil prediksi sistem memiliki jumlah kesalahan yang lebih kecil daripada hasil prediksi oleh Deptan. Maka dapat disimpulkan bahwa hasil prediksi menggunakan Neuro-Fuzzy struktur Mod\_ANFIS menghasilkan kesalahan lebih baik, hal ini dibuktikan dengan *error* yang lebih kecil.



## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Secara garis besar hasil perancangan dan pembuatan sistem Penerapan Neuro-Fuzzy struktur Mod\_ANFIS untuk Prediksi Tingkat Pendapatan Petani yang telah dilakukan dapat disimpulkan bahwa:

1. Sistem Neuro-Fuzzy struktur Mod\_ANFIS dapat dipergunakan untuk memprediksi tingkat pendapatan petani
2. Aplikasi dapat membantu menghasilkan keputusan yang lebih akurat untuk dijadikan landasan dalam proses penghitungan Prediksi Tingkat Pendapatan Petani.
3. Selisih rata-rata antara peramalan oleh Deftan dan Neuro-Fuzzy untuk MAPE Padi sebesar 0,0165 %, Jagung sebesar 4,445 %, Kedelai sebesar 0,461 %, Ketela Pohon sebesar 0,235 %, Ketela Rambat sebesar 4,487 %, Kacang Tanah sebesar 4,904 %.

#### 5.2 Saran

Sistem dapat dibuat *online* atau *mobile* sehingga dapat diakses oleh siapa saja dan tidak ada batasan waktu.

## DAFTAR PUSTAKA

Jang, J.-S.R, Sun, C.-T dan Mizutani, E, 1996, *Neuro-Fuzzy and Soft Computing*, Prentice Hall.

Kusumadewi, Sri., 2002, *Analisis & Desain Sistem Fuzzy Menggunakan Toolbox Matlab*, Graha Ilmu, Yogyakarta.

Makridakis, Spyros, Wheelwright, Steven C., McGee, Victor E., 1993, *Metode dan Aplikasi Peramalan edisi kedua jilid satu*, PT. Binarupa Aksara, Jakarta.

Assauri, Sofjan, 1984, *Teknik & Metode Peramalan*, Fakultas Ekonomi Universitas Indonesia, Jakarta.

Ayres, Frank, 1994, *Teori dan Soal-soal Matriks*, Erlangga, Jakarta.

Jogiyanto, 1999, *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*, Andi, Yogyakarta.

Pranata, Antony, 2000, *Pemrograman Borland Delphi*, Andi, Yogyakarta.

Setiawan, Kuswara, 2003, *Paradigma Sistem Cerdas*, Banyumedia Publising, Malang.

Rahmat, Basuki, 2004, *Penerapan Sistem Neuro-Fuzzy untuk Prediksi Curah Hujan Daerah Banyuwangi*, Jurnal Teknik Komputer:GEMATEK, 6(2), 63-72.

Martina, Inge, 2002, *Database Client/Server Menggunakan Delphi*, PT. Elex Media Komputindo, Jakarta

Romeo, 2003, *Testing dan Implementasi Sistem*, STIKOM, Surabaya.