

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sampai saat ini, pengangguran merupakan salah satu masalah yang belum terselesaikan meskipun semua upaya dilakukan untuk menurunkan jumlah tingkat pengangguran. Pengangguran adalah semua orang dalam referensi waktu tertentu, yaitu pada usia angkatan kerja yang tidak bekerja, baik dalam arti mendapatkan upah atau bekerja mandiri, kemudian mencari pekerjaan, dalam arti mempunyai kegiatan aktif dalam mencari kerja tersebut. (Murti,2003). Pada tahun 2000, tingkat pengangguran masih mengalami kenaikan, seiring pertumbuhan ekonomi yang mencapai 4,8%, namun jumlah penganggur tahun 2000 masih jauh lebih tinggi dibandingkan sebelum krisis ekonomi tahun 1997. (BPS,2001). Oleh karena tingginya jumlah pengangguran ini pemerintah berupaya untuk meramalkan kondisi keadaan jumlah tingkat pengangguran yang akan datang, agar pemerintah mampu menanggulangi jumlah pengangguran yang tinggi tersebut.

Prediksi atau *prediction* adalah proses perkiraan atau pengukuran berdasarkan perilaku data masa lampau untuk diproyeksikan ke masa depan dengan memanfaatkan persamaan matematika dan statistika. Perkiraan atau pengukuran dapat dilakukan secara kualitatif maupun kuantitatif. Perkiraan secara kualitatif biasanya menggunakan pendapat ahli, sedangkan perkiraan secara kuantitatif menggunakan metode statistik dan matematik.(Siana,2006). Dalam penentuan besarnya jumlah tingkat pengangguran ini pemerintah menggunakan metode *Ekonometrik*, yaitu salah satu tehnik prediksi menggunakan metode *eksplanatoris*, dimana metode ini belum diketahui keakuratannya. Dari masalah ini munculah metode-metode lain yang digunakan untuk prediksi salah satunya adalah metode *Adaptive Neuro Fuzzy Inference System (ANFIS)*.

*ANFIS* adalah suatu sistem yang menggabungkan kemampuan jaringan syaraf tiruan dan logika fuzzy. Jaringan syaraf tiruan adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Logika *Fuzzy* adalah suatu cara untuk memetakan suatu ruang input ke dalam suatu ruang output. Dimana didalamnya

terdapat teori himpunan *fuzzy* yang pada dasarnya merupakan perluasan dari teori himpunan klasik. (Kusumadewi,2006). Pada logika *fuzzy* memiliki kemampuan lebih dalam menangani data pengetahuan lingkungan luar serta kemampuan dalam persepsi dan penalaran seperti otak manusia namun tidak memiliki kemampuan untuk belajar dan beradaptasi. Pada sistem jaringan syaraf tiruan memiliki kemampuan untuk belajar dan beradaptasi namun tidak memiliki kemampuan penalaran seperti yang dimiliki pada sistem logika *fuzzy*. (Jantzen,1998).

Kelebihan dari sistem *ANFIS* jika dibandingkan dengan sistem tunggal dan sistem *Neuro-Fuzzy* yang lain adalah secara otomatis mencari nilai-nilai pada parameter baik premis maupun konsekuen dengan algoritma pembelajaran terhadap sekumpulan data, dimana kelebihan ini dipertegas lagi dengan adanya penelitian Teguh dan Dyan yang mengimplementasikan sistem *ANFIS* dalam prediksi produksi air minum di PDAM Surabaya dan dari penelitian menghasilkan kesalahan belajar sebesar 1,98% dengan kesalahan prediksi rata-rata sebesar 2,245%.

Berdasarkan latar belakang yang telah dipaparkan, maka judul yang diambil dalam skripsi ini adalah **“Prediksi Tingkat Pengangguran Kota Malang Menggunakan Adaptive Neuro Fuzzy Inference System”**.

## 1.2 Perumusan Masalah

Rumusan masalah dalam skripsi ini antara lain :

1. Bagaimana mengimplementasikan *ANFIS* sebagai sistem yang digunakan untuk meramalan tingkat pengangguran.
2. Berapa prosentase tingkat keakuratan atau akurasi dalam prediksi pengangguran menggunakan metode *ANFIS*.

## 1.3 Batasan Masalah

Batasan masalah dalam skripsi ini adalah sebagai berikut :

1. Masalah pengangguran dibatasi pada studi tentang faktor-faktor terjadinya pengangguran.
2. Parameter yang digunakan dalam sistem ini berdasarkan data-data teknis yang diambil dari BPS Malang. Data-data ini antara lain : jumlah penduduk usia produktif, jumlah angkatan kerja, jumlah pekerja,TPAK ,TKK dan Indeks SDM.

#### **1.4 Tujuan Penelitian**

Adapun tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah :

1. Mengimplementasikan sistem *ANFIS* kedalam sistem prediksi tingkat pengangguran.
2. Mengukur prosentase tingkat keakuratan atau akurasi prediksi pengangguran dengan menggunakan metode *ANFIS*.

#### **1.5 Manfaat Penelitian**

Dari penelitian ini, diharapkan sistem yang dibangun *ANFIS* dapat digunakan untuk meramalkan tingkat pengangguran di waktu yang akan datang, sehingga akan didapatkan berapa prosentase tingkat akurasinya. Dan untuk lebih lanjut, dapat dikembangkan untuk aplikasi-aplikasi yang berhubungan dengan sistem prediksi.

#### **1.6 Metodologi Pemecahan Masalah**

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan skripsi ini adalah :

1. Studi Literatur  
Studi dilakukan dengan cara mencari sekaligus mempelajari beberapa literatur dan artikel mengenai pengangguran. Disamping itu juga mempelajari metode yang digunakan untuk meramalkan tingkat pengangguran menggunakan *ANFIS*, sebagai acuan dalam perencanaan dan pembuatan skripsi.
2. Perancangan dan pembuatan program  
Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut dalam suatu program komputer.
3. Uji coba dan analisa hasil implementasi  
Menguji perangkat lunak, dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.
4. Penyusunan laporan  
Membuat laporan tertulis mengenai hasil pembuatan skripsi ini.

#### **1.7 Sistematika Penulisan**

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut :

## 1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.

## 2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan konsep *ANFIS* dan pengangguran.

## 3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam prediksi tingkat pengangguran.

## 4. BAB IV IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa sistem perangkat lunak yang dirancang.

## 5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dan seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.



## BAB II TINJAUAN PUSTAKA

### 2.1 PREDIKSI

Prediksi atau *prediction* adalah proses perkiraan atau pengukuran berdasarkan perilaku data masa lampau untuk diproyeksikan ke masa depan dengan memanfaatkan persamaan matematika dan statistika. Perkiraan atau pengukuran dapat dilakukan secara kualitatif maupun kuantitatif.

#### 2.1.1 Metode Kualitatif

Perkiraan secara kualitatif biasanya menggunakan pendapat ahli. Metode ini digunakan dimana tidak ada model matematik, biasanya dikarenakan data yang ada tidak cukup representatif untuk meramalkan masa yang akan datang.

#### 2.1.2 Metode Kuantitatif

Perkiraan kuantitatif menggunakan metode statistik dan matematik, metode ini biasa disebut *time series*. Data *time series* adalah data deret waktu yaitu sekumpulan data pada satu periode waktu tertentu. Dengan kata lain metode ini digunakan dimana tidak ada model matematik, biasanya dikarenakan data yang ada tidak cukup representatif untuk meramalkan masa yang akan datang. (Siana, 2006.)

### 2.2 PENGANGGURAN

#### 2.2.1 Definisi Pengangguran

Deinisi pengangguran secara teknis adalah orang yang tidak bekerja, sedang mencari pekerjaan, mempersiapkan suatu usaha baru, dan tidak mencari pekerjaan karena merasa tidak mungkin mendapat pekerjaan. Selain definisi di atas masih banyak jatilah arti definisi pengangguran diantaranya :

- **Definisi pengangguran menurut Sadono Sukimo**  
Pengangguran adalah suatu keadaan dimana seseorang yang tergolong dalam angkatan kerja ingin mendapatkan pekerjaan tetapi belum dapat memperolehnya.
- **Definisi pengangguran menurut Payman J. Simanjutak**  
Pengangguran adalah orang yang tidak bekerja sama sekali

atau bekerja kurang dari dua hari selama seminggu sebelum pemecahan dan berusaha memperoleh pekerjaan.

- **Definisi pengangguran berdasarkan istilah umum dan latihan tenaga kerja**

Pengangguran adalah orang yang tidak mampu mendapatkan pekerjaan yang menghasilkan uang.

### 2.2.2 Sebab-Sebab Terjadinya Pengangguran

Faktor-faktor yang menyebabkan terjadinya pengangguran adalah sebagai berikut:

- Besarnya Angkatan Kerja Tidak Seimbang dengan Kesempatan Kerja
- Struktur Lapangan Kerja Tidak Seimbang
- Kebutuhan jumlah dan jenis tenaga terdidik dan penyediaan tenaga terdidik tidak seimbang
- Meningkatnya peranan dan aspirasi Angkatan Kerja Wanita dalam seluruh struktur Angkatan Kerja Indonesia
- Penyediaan dan Pemanfaatan Tenaga Kerja antar daerah tidak seimbang

### 2.2.3 Parameter Pengangguran

Adapun Faktor-faktor pokok yang mempengaruhi tingkat pengangguran antara lain :

- **Tenaga Kerja**

Tenaga kerja adalah jumlah seluruh penduduk dalam suatu negara yang dapat memproduksi barang-barang dan jasa-jasa jika ada permintaan terhadap tenaga mereka dan jika mereka mau berpartisipasi dalam aktivitas tersebut. Berdasarkan Survey Sosial dan Ekonomi Nasional 2000 mengatakan bahwa tenaga kerja adalah seluruh yang berumur 15 tahun keatas penduduk usia kerja.

- **Bekerja**

Bekerja adalah mereka yang selama seminggu sebelum melakukan pekerjaan atau bekerja dengan maksud memperoleh atau membantu memperoleh penghasilan atau keuntungan selama paling sedikit satu jam dalam seminggu yang lalu dan tidak boleh terputus.

- **Angkatan Kerja**

Angkatan kerja adalah mereka yang selama seminggu yang lalu mempunyai pekerjaan, baik yang bekerja maupun yang sementara tidak bekerja karena suatu sebab seperti yang sedang menunggu panen, dan pegawai cuti. Di samping itu mereka tidak mempunyai pekerjaan tetapi sedang mencari pekerjaan juga termasuk dalam kelompok angkatan kerja.

- **Tingkat Partisipasi Angkatan Kerja (TPAK)**

Tingkat Partisipasi Angkatan Kerja adalah perbandingan antara jumlah angkatan kerja dengan penduduk usia kerja dalam kelompok umur yang sama.

$$TPAK = \frac{\text{jumlah angkatan kerja}}{\text{Jumlah penduduk usia kerja}} \cdot 100\%$$

- **Kesempatan Kerja**

Pengangguran dapat terjadi karena jumlah lapangan kerja dengan angkatan kerja atau jumlah permintaan dan penawaran tidak seimbang. Sehingga jumlah pertumbuhan tenaga kerja diatas mempengaruhi pertumbuhan angka kesempatan kerja. Pertumbuhan kesempatan kerja yang kecil dikarenakan faktor investasi dan faktor pertumbuhan ekonomi tidak bertambah.

- **Kesempatan Kerja (TKK)**

Tingkat Kesempatan Kerja adalah perbandingan antara jumlah angkatan kerja yang bekerja dengan angkatan kerja.

$$TKK = \frac{\text{jumlah angkatan kerja yang bekerja}}{\text{Jumlah angkatan kerja}} \cdot 100\%$$

- **Indeks Sumber Daya Manusia**

Indeks SDM merupakan pengukuran perbandingan dari harapan hidup, melek huruf, pendidikan dan standar hidup untuk semua negara seluruh dunia. Indeks ini digunakan untuk mengklasifikasikan apakah sebuah negara adalah negara maju, negara berkembang atau negara terbelakang dan juga untuk mengukur pengaruh dari kebijaksanaan ekonomi terhadap kualitas hidup.

## 2.3 LOGIKA FUZZY

### 2.3.1 Himpunan Klasik (Crisp)

Pada dasarnya, himpunan *fuzzy* merupakan perluasan dari teori himpunan klasik. Pada teori himpunan klasik, keberadaan suatu elemen pada suatu himpunan  $A$ , hanya akan memiliki 2 kemungkinan keanggotaan, yaitu menjadi anggota  $A$  atau tidak menjadi anggota  $A$  (Chak,1998). Suatu nilai yang menunjukkan seberapa besar tingkat keanggotaan suatu elemen ( $x$ ) dalam suatu himpunan ( $A$ ), sering dikenal dengan derajat keanggotaan, yang dinotasikan sebagai  $\mu_A(x)$

### 2.3.2 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item  $x$  dalam suatu himpunan  $A$ , yang sering ditulis dengan  $\mu_A(x)$ , memiliki dua kemungkinan, yaitu :

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Jika pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 1 atau 0, pada himpunan *fuzzy* nilai keanggotaannya terletak pada rentang 0 sampai 1. Apabila  $x$  memiliki nilai keanggotaan *fuzzy*  $\mu_A(x) = 1$  berarti  $x$  menjadi anggota penuh pada himpunan  $A$ . Himpunan *fuzzy* memiliki 2 atribut, yaitu :

- Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami.
- Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu :

- Variabel *fuzzy* merupakan variabel yang akan dibahas dalam sistem *fuzzy*, misalnya umur, temperature, permintaan, dsb.
- Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi tertentu dalam variabel *fuzzy*, misalnya variabel umur dibagi atas 3 himpunan *fuzzy*, yaitu muda, tua, parobaya.
- Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*, misalnya semesta pembicaraan variabel umur adalah 0 sampai 100.



- d. Domain adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam himpunan *fuzzy*, misalnya domain umur muda 20-45, domain parobaya 25-65, dan domain tua 45-70.(Kusumadewi, 2010)

## 2.4 FUNGSI KEANGGOTAAN

Fungsi keanggotaan (*Membership Function*) himpunan *fuzzy* satu dimensi dengan masukan tunggal yang digunakan dalam prediksi kali ini adalah **Generalized Bell**. Gambar 2.1 menunjukkan karakteristik fungsi kurva gauss. Bentuk dari fungsi keanggotaan *generalized bell* ditentukan oleh parameter  $a$ ,  $b$  dan  $c$ , yang didefinisikan pada persamaan 2.1

$$bell(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.1)$$

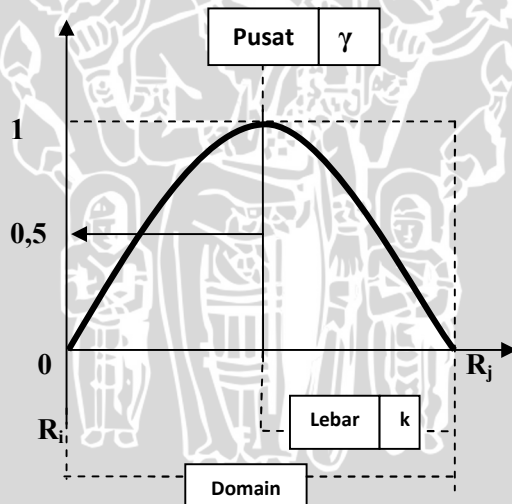
Dimana :

$x$  : data input-an

$b$  : 1

$a$  : nilai standar deviasi (stdev)

$c$  : rata-rata (mean)



**Gambar 2.1** Karakteristik Fungsional Kurva GAUSS

Dapat dilihat pada kurva *GAUSS* ini menjelaskan bahwa ( $\gamma$ ) menunjukkan nilai domain pada pusat kurva, dan ( $k$ ) menunjukkan lebar kurva.

## 2.5 K-MEANS CLUSTERING

K-Means Clustering pada prediksi kali ini digunakan untuk menentukan cluster suatu data, dimana data-data ini sebelumnya belum diketahui jumlah clusternya. Data clustering merupakan salah satu metode data mining yang bersifat tanpa arahan (*unsupervised*). Ada dua jenis data *clustering* yang sering digunakan dalam proses pengelompokan data yaitu :

- a. *Hierarchical (hiraki) data clustering*
- b. *Non-Hierarchical (non-hiraki) data clustering*

K-Means merupakan salah satu metode data *clustering non hiraki* yang mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*. Metode ini mempartisi data ke dalam *cluster* yang sama sehingga data yang memiliki karakteristik sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam cluster yang lain.

Langkah-langkah untuk melakukan pengelompokan dengan tujuan menghasilkan suatu data yang terkelompok adalah sebagai berikut:

1. Standarisasi data yang akan dikelompokkan (menentukan bobot dari data mentah yang telah didapatkan). Hal ini dilakukan agar data mempunyai skala yang sama, sehingga pengelompokan akan stabil seperti pada **persamaan 2.2** :

$$x_{ij} = \frac{x_{ij} - xMin_j}{xMax_j - xMin_j} \quad (2.2)$$

Dimana :

- $x_{ij}$  = data yang distandarisasi  
 $xMin_j$  = nilai min pada table ke-j  
 $xMax_j$  = nilai max pada table ke-j

2. Melakukan pengelompokan dengan metode *K-Means Clustering*, langkah-langkahnya sebagai berikut :
  - a. Pilih jumlah *cluster* k.
  - b. Inisialisasi k pusat *cluster* ini bisa dilakukan dengan berbagai cara. Yang paling sering dilakukan adalah dengan cara *random*. Pusat-pusat *cluster* diberi nilai awal dengan angka-angka random.
  - c. Tempatkan setiap data atau obyek ke *cluster* terdekat. Kedekatan dua obyek ditentukan jarak antara data dengan

pusat *cluster*. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat *cluster*. Jarak paling dekat antara satu data dengan data satu *cluster* tertentu akan menentukan suatu data masuk dalam *cluster* mana.

Untuk menentukan ukuran kemiripan atau ketidakmiripan antar data dengan metode jarak *Euclidean* seperti pada **persamaan 2.3**

$$d(x, y) = ||x - y|| = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (2.3)$$

Dimana :

$d(x, y)$  : ukuran ketidakmiripan

$X$  ( $x_1, x_2, \dots, x_j$ ) : variabel data

$Y$ : ( $y_1, y_2, \dots, y_j$ ) : variabel pada titik pusat *cluster*.

- d. Hitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data atau obyek dalam *cluster* tertentu. **Persamaan 2.4** untuk pencarian rata-rata.

$$w = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (2.4)$$

Dimana :

$w$  : bobot tiap inputan

$x_n$  : data inputan ke-n

$n$  : jumlah data

Jika dikehendaki bisa juga memakai median dari *cluster* tersebut. Jika rata-rata (*mean*) bukan satu-satunya ukuran yang bisa dipakai.

- e. Tugaskan lagi setiap obyek dengan memakai pusat *cluster* yang baru. Jika pusat *cluster* sudah tidak berubah lagi, maka proses pengklasteran selesai. Atau, kembali lagi ke langkah ketiga sampai pusat *cluster* tidak berubah.
- f. Setelah selesai, maka didapatkanlah data yang telah ter*cluster* yang akan dimasukkan untuk pencarian *mean* pada **persamaan 2.5** dan standar deviasi pada persamaan 2.6

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (2.5)$$

$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{(n-1)}} \quad (2.6)$$

Dimana :

$\bar{x}$  : rata-rata (mean)

$\sigma$  : standar deviasi

$x_i$  : data inputan ke-i

$n$  : banyaknya data

## 2.6 SISTEM INFERENSI FUZZY

Sistem *inferensi fuzzy* merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *IF-THEN* dan penalaran *fuzzy*. Secara garis besar, proses *inferensi fuzzy* dapat dijelaskan sebagai berikut :

- Sistem *inferensi fuzzy* menerima input *crisp*
- Kemudian dikirim ke basis pengetahuan yang berisi 'n' aturan *fuzzy* dalam bentuk *IF-THEN*
- *Fire Strength* akan dicari pada setiap aturan, apabila jumlah aturan lebih dari satu, maka akan dilakukan agregasi dari semua aturan
- Selanjutnya, pada hasil agregasi akan dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai *output* sistem.

Sistem *inferensi fuzzy* yang umum digunakan pada sistem seperti *ANFIS*, yaitu metode Takagi Sugeno Kang (TSK).

Penalaran dengan metode SUGENO hampir sama dengan penalaran MAMDANI, hanya saja *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

### a. Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model *fuzzy* SUGENO Orde-Nol adalah:

$$IF (x_1 \text{ is } A_1) \cdot (x_2 \text{ is } A_2) \cdot (x_3 \text{ is } A_3) \cdot \dots \cdot (x_N \text{ is } A_N) THEN z=k$$

Dimana :

$A_i$  : himpunan *fuzzy* ke-i sebagai anteseden

$k$  : suatu konstanta (tegas) sebagai konsekuen

### b. Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model *fuzzy* SUGENO Orde-Satu adalah:

$$IF (x_1 \text{ is } A_1) \cdot \dots \cdot (x_N \text{ is } A_N) THEN z = p_1 * x_1 + \dots + p_N * x_N + q$$

Dimana :

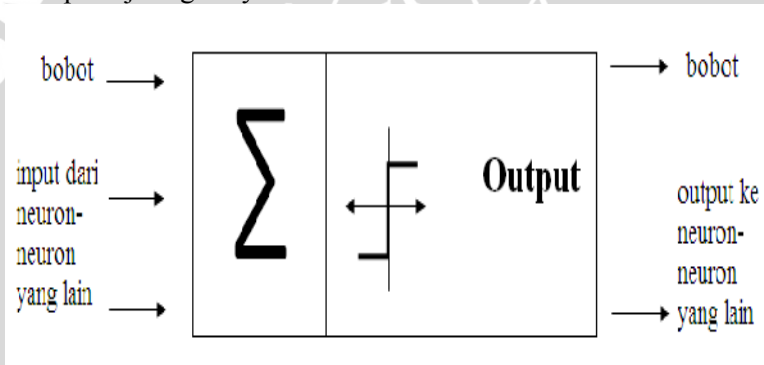
$A_i$  : himpunan *fuzzy* ke- $i$  sebagai anteseden

$p_i$  : suatu konstanta (tegas) sebagai konsekuen

$q$  : juga merupakan konstanta dalam konsekuen.

## 2.7 JARINGAN SYARAF TIRUAN

Seperti halnya otak manusia, jaringan syaraf tiruan juga terdiri dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarannya menuju ke neuron-neuron yang lain. Pada jaringan syaraf tiruan, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. **Gambar 2.2** menunjukkan struktur neuron pada jaringan syaraf.



**Gambar 2.2** Struktur *Neuron* Jaringan Syaraf

Jika kita lihat, neuron buatan ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron-neuron biologis. Informasi (*input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan keluaran (*output*) melalui bobot-

bobot keluarannya ke semua neuron yang berhubungan dengannya, demikian seterusnya.

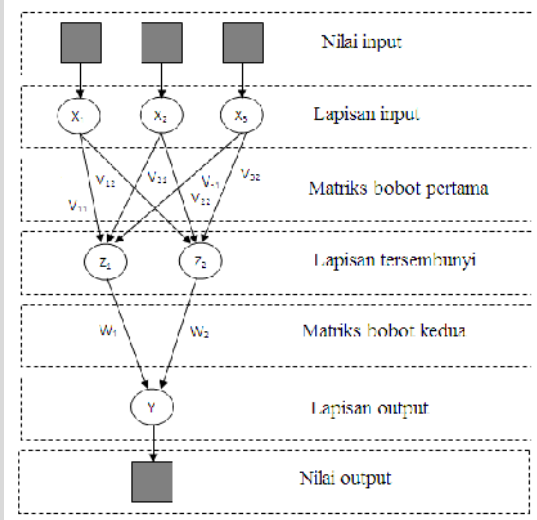
### 2.7.1 Arsitektur Jaringan Syaraf

Hubungan antara neuron dalam jaringan syaraf mengikuti pola tertentu tergantung pada arsitektur jaringan syarafnya. Pada dasarnya ada 3 macam arsitektur jaringan syaraf, yaitu (Kusumadewi,2006) :

- a. Arsitektur Jaringan dengan Lapisan Tunggal (*Single Layer Net*)
- b. Arsitektur Jaringan dengan Banyak Lapisan (*Multilayer Net*)
- c. Arsitektur Jaringan dengan Lapisan Kompetitif (*Competitive Layer Net*)

Pada kasus prediksi ini arsitektur jaringan yang dipakai adalah *Multilayer Net*. Ciri khas *layer* ini adalah memiliki 3 jenis *layer* yaitu *layer input*, *layer output* dan *layer* tersembunyi. Kelebihan arsitektur ini adalah dapat menyelesaikan permasalahan yang lebih kompleks dibanding lapisan tunggal, Namun proses pelatihan sering membutuhkan waktu yang cenderung lama.

**Gambar 2.3** Menunjukkan arsitektur multy-layer net.



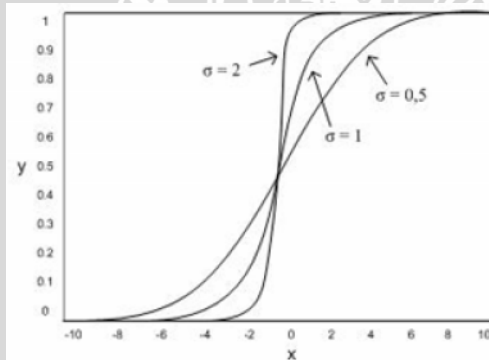
**Gambar 2.3** Arsitektur Multy-layer Net

### 2.7.2 Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering dipakai dalam jaringan syaraf tiruan, antara lain :

- Fungsi undak *biner*
- Fungsi *bipolar*
- Fungsi *linear*
- Fungsi satu rating *linear*
- Fungsi *symmetric satu rating linear*
- Fungsi *sigmoid biner*
- Fungsi *sigmoid bipolar*

Pada kasus prediksi kali ini fungsi aktivasi yang dipakai adalah fungsi aktivasi *sigmoid biner*. Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi *sigmoid biner* memiliki nilai pada *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada *interval* 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1. **Gambar 2.4** menunjukkan fungsi aktivasi *sigmoid biner*.



**Gambar 2.4** Fungsi Aktivasi *Sigmoid Biner*

Fungsi *sigmoid biner* dirumuskan sebagai berikut :

$$y = f(x) = \frac{1}{1+e^{\sigma x}} \quad (2.7)$$

dengan :

$$f'(x) = \sigma f(x)[1 - f(x)] \quad (2.8)$$

### 2.7.3 Proses Pembelajaran

Pada umumnya jika menggunakan jaringan syaraf tiruan, hubungan *input* dan *output* harus diketahui secara pasti dan jika hubungan tersebut telah diketahui maka dapat dibuat suatu model. Hal lain yang penting adalah proses belajar hubungan *input-output*

dilakukan dengan pembelajaran. Proses belajar dalam hal ini dapat dikategorikan dalam dua jenis yaitu :

- a. Pembelajaran Terawasi (*Supervised Learning*)  
Dikatan terawasi jika output yang diharapkan telah diketahui sebelumnya. Proses belajar dengan pengawasan memerlukan keluaran yang diinginkan sebagai dasar perubahan pembobotnya. Biasanya pembelajaran dilakukan dengan menggunakan data yang telah ada.
- b. Pembelajaran Tak Terawasi (*Unsupervised Learning*)  
Pada metode pembelajaran yang tak terawasi ini tidak memerlukan target output. Di dalam perubahan pembobotnya dilakukan dengan sendirinya sebagai tanggapan atas masukan tanpa memerlukan acuan keluaran.

## 2.8 ADAPTIVE NEURO FUZZY INFERENCE SYSTEM

*Adaptive Neuro Fuzzy Inference System* (ANFIS) atau biasa dikenal dengan *NEURO-FUZZY* adalah gabungan dari dua sistem yaitu sistem logika *fuzzy* dan jaringan syaraf tiruan. Dimana arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Biasa dikatan bahwa *ANFIS* adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. Hal ini membuat basis aturan-aturan untuk beradaptasi. (Jantzen,1998).

Agar jaringan dengan fungsi basis radial *ekuivalen* dengan *fuzzy* berbasis aturan model Sugeno orde 1 ini, diperlukan batasan, yaitu:

- a. Keduanya harus memiliki metode agregasi yang sama untuk menurunkan semua outputnya.
- b. Jumlah fungsi aktivasi harus sama dengan jumlah aturan *fuzzy*.
- c. Jika ada beberapa *input* pada basis aturannya, maka tiap-tiap fungsi aktivasi harus sama dengan fungsi keanggotaan setiap *inputnya*.
- d. Fungsi aktivasi dan aturan-aturan *fuzzy* harus memiliki fungsi yang sama untuk neuron-neuron dan aturan-aturan yang ada di sisi *outputnya*.(Kusumadewi,2006).



### 2.8.1 Arsitektur ANFIS

Jika terdapat dua input  $x_1$  dan  $x_2$  dan 1 output,  $y$ . Ada 2 aturan pada basis aturan model Sugeno orde 1 yaitu:

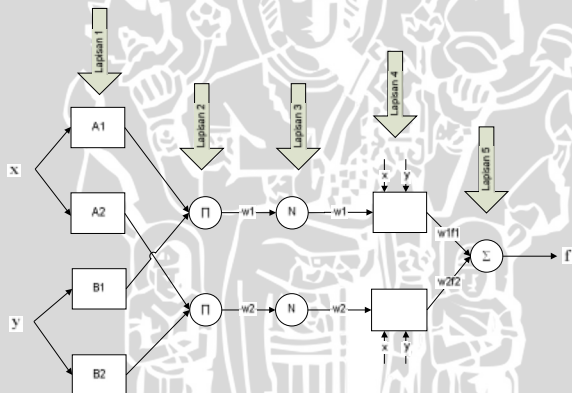
$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } B_1 \text{ THEN } y_1 = c_{11}x_1 + c_{22}x_2 + c_{10}$$

$$\text{IF } x_1 \text{ is } A_2 \text{ AND } x_2 \text{ is } B_2 \text{ THEN } y_2 = c_{21}x_1 + c_{22}x_2 + c_{20}$$

Jika predikat untuk kedua aturan  $w_1$  dan  $w_2$ , maka dapat dihitung rata-rata terbobot :

$$\bar{y} = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2} = w_1 y_1 + w_2 y_2 \quad (2.9)$$

Arsitektur jaringan ANFIS terdiri dari 5 lapisan, Lapisan yang disimbolkan dengan kotak adalah lapisan yang bersifat *adaptif*. Sedangkan yang disimbolkan dengan lingkaran adalah bersifat tetap. Setiap keluaran dari masing-masing lapisan disimbolkan dengan " $O_{1,i}$ ", dimana  $i$  adalah urutan simpul dan 1 adalah urutan lapisan. Seperti terlihat pada **Gambar 2.5**.



**Gambar 2.5** Arsitektur Jaringan ANFIS

Berikut ini adalah penjelasan untuk setiap lapisan, yaitu:

#### a. Lapisan 1 : Fuzzifikasi

Setiap simpul  $i$  pada lapisan ini adalah simpul *adaptif*, yang berfungsi untuk membangkitkan derajat keanggotaan :

$$o_{1,j} = \mu_{A1}(x) \quad i = 1, 2, \dots, n \quad (2.10)$$

atau

$$o_{1,j} = \mu_{B1}(y) \quad i = 1, 2, \dots, n$$

Dimana :

$x$  dan  $y$  : masukakan bagi simpul ke- $i$

$A_i$  : label bahasa (*linguistic label*)

$O_{i,j}$  : tingkat keanggotaan dari himpunan fuzzy A

Untuk menghitung derajat keanggotaan menggunakan **persamaan**

**2.1 :**

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

Dimana :

$a$  dan  $c$  : himpunan parameter/parameter premis

$\mu_A(x)$  : derajat keanggotaan

### **b. Lapisan 2 : Perkalian Fuzzy**

Setiap simpul pada lapisan ini diberi label  $\Pi$ , bersifat *non-adaptif*. Berfungsi untuk membangkitkan kuat penyulutan (*firing-strength*). Setiap *node* pada lapisan ini adalah *node* tetap dengan keluarannya adalah produk dari sinyal yang datang.

$$O_{2i} = w_i = \mu_A(x), \mu_B(y) \quad i=1,2,..n \quad (2.11)$$

Dimana :

$w_i$  : *firing-strength*/kuat penyulutan

$\mu_A(x)$  : derajat keanggotaan himpunan A

$\mu_B(y)$  : derajat keanggotaan himpunan B

### **c. Lapisan 3 : Pembobotan**

Setiap simpul pada lapisan ini diberi label N, yang juga bersifat *non-adaptif*. Berfungsi untuk menormalkan *firing strength*. Masing-masing simpul menampilkan derajat pengaktifan ternormalisasi dengan bentuk:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad (2.12)$$

Dimana :

$w_i$  : *firing-strength*

$w_1$  dan  $w_2$  : *output*

$\bar{w}_i$  : *normalized firing strength*

### **d. Lapisan 4 : Penjumlahan**

Berfungsi untuk menghitung keluaran kaidah berdasarkan parameter *consequent*. Tiap simpul pada lapisan ini berupa simpul *adaptif*, dengan fungsi simpul:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y) \quad (2.13)$$

atau

$$O_{4,i} = \overline{w}_i y_i = w_i (c_{i1}x_1 + c_{i2}x_2 + \dots + c_{in}x_n + c_{i0})$$

Dimana :

$\overline{w}_i$  : derajat pengaktifan/kuat penyulutan ternormalisasi

$p_i, q_i$  : himpunan parameter simpul

$y_i$  : output

$(c_{i1}, c_{i2}, \dots, c_{in}, c_{i0})$  : consequent parameter

### e. Lapisan 5 : Defuzzifikasi

Simpul pada lapisan ini diberi label  $\Sigma$ . Node tunggal pada lapisan ini adalah node tetap yang menghitung keluaran keseluruhan sebagai penjumlahan semua sinyal yang datang.

$$O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.14)$$

atau

$$O_{5,i} = \sum_i \overline{w}_i y_i = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

## 2.8.2 Algoritma Pembelajaran Hybrid

ANFIS dilatih dengan algoritma pembelajaran *hybrid*. Algoritma pembelajaran *Hybrid* terdiri dari dua langkah, yaitu [Jang, 1997], mengemukakan algoritma *hybrid* yang akan menggabungkan antara *steepest-descent* dan *least square estimation*. langkah maju (*feed forward*) dan langkah balik (*feed back*).

Pada langkah maju, input jaringan akan merambat maju sampai ke lapisan keempat di mana consequent parameter,  $(c_{i1}, c_{i2}, \dots, c_{in}, c_{i0})$ , akan diidentifikasi dengan menggunakan metode *least square estimator*. Sedangkan pada langkah mundur, error sinyal akan merambat mundur, dan parameter-parameter a dan c akan diperbaiki dengan menggunakan metode *gradient-descent* atau *backpropagation*.

Meskipun dapat digunakan algoritma *gradient-descent* atau *backpropagation* untuk mengidentifikasi parameter-parameter pada suatu jaringan *adaptif*, namun biasanya penggunaan algoritma ini membutuhkan waktu yang relatif lama untuk *konvergen*.

## 2.8.3 Metode Least Square Estimator (LSE)

Ketika nilai parameter-parameter bagian premise telah ditentukan, maka total keluaran dapat dinyatakan sebagai kombinasi

linier dari parameter-parameter konsekuen. Jika diketahui keluaran dari model linier  $y$  yang diekspresikan melalui **persamaan 2.15**:

$$y = \theta_1 f_1(u) + \theta_2 f_2(u) + \dots + \theta_n f_n(u) \quad (2.15)$$

Dimana :

$y$   $[y_1, y_2, \dots, y_m]^T$  model *vector output*

$u$   $[u_1, u_2, \dots, u_p]^T$  model *vector input*

$f_n$  fungsi  $u$  yang diketahui

$\theta_n$  parameter yang diestimasi

Dengan menggunakan notasi matrik didapatkan :

$$A\theta = y \quad (2.16)$$

Penyelesaian terbaik untuk  $\theta$ , yang meminimalkan

$\|A\theta - y\|^2$  adalah *least square estimator* (LSE)  $\theta$  :

$$\theta = (A^T A)^{-1} A^T y \quad (2.17)$$

Dimana:

$A^T$  : transpose dari  $A$ .

### 2.8.4 Error Back Propagation (EBP)

Setelah pembelajaran dengan LSE, selanjutnya jaringan *adaptif* dapat dilatih untuk mendapatkan parameter  $a$  (standar deviasi) dan  $c$  (*mean/rata-rata*) dengan konsep *propagasi error steepest descent* untuk mendapatkan parameter ( $a$  dan  $c$ ) baru.

Misalkan terdapat jaringan *adaptif* yang terdiri 5 lapisan, seperti **gambar 2.5** dan terdapat sebanyak  $N(L)$  neuron pada lapisan ke- $L$ , yang hanya memiliki 1 simpul pada lapisan output (simpul/neuron 5) maka :

#### a. Error pada lapisan ke-5

*Propogasi error* yang menuju lapisan ke-5 dapat dirumuskan :

$$\varepsilon_5 = \frac{\partial E_p}{\partial x_{13}} = -2 (y_p - y_p) \quad (2.18)$$

Dimana:

$y_p$  : target *output* data pelatihan ke- $p$

$y_{p^*}$  : *output* jaringan pada data pelatihan ke- $p$

**b. Error pada lapisan ke-4**

*Propogasi error* yang menuju lapisan ke-4 dapat dirumuskan :

$$\varepsilon_{11} = \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) = \varepsilon_{13} \left( \frac{\partial f_{13}}{\partial x_{11}} \right) = \varepsilon_{13}(1) = \varepsilon_{13} \quad (2.19)$$

Karena,  $f_{13} = \overline{w}_1 f_1 + \overline{w}_2 f_2$ , maka  $\left( \frac{\partial f_{13}}{\partial \overline{w}_2 f_2} \right) = 1$

**c. Error pada lapisan ke-3**

*Propogasi error* yang menuju lapisan ke-3 dapat dirumuskan :

$$\varepsilon_9 = \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) \left( \frac{\partial f_{11}}{\partial x_9} \right) = \varepsilon_{11} \left( \frac{\partial f_{11}}{\partial x_9} \right) = \varepsilon_{11} f_1 \quad (2.20)$$

$$\varepsilon_{10} = \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) \left( \frac{\partial f_{12}}{\partial x_{10}} \right) = \varepsilon_{11} \left( \frac{\partial f_{12}}{\partial x_{10}} \right) = \varepsilon_{12} f_2 \quad (2.21)$$

**d. Error pada lapisan ke-2**

*Propogasi error* yang menuju lapisan ke-2 dapat dirumuskan :

$$\begin{aligned} \varepsilon_7 &= \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) \left( \frac{\partial f_{11}}{\partial x_9} \right) \left( \frac{\partial f_9}{\partial x_7} \right) + \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) \left( \frac{\partial f_{12}}{\partial x_{10}} \right) \left( \frac{\partial f_{10}}{\partial x_7} \right) \\ &= \varepsilon_9 \left( \frac{\partial f_9}{\partial x_7} \right) + \varepsilon_{10} \left( \frac{\partial f_{10}}{\partial x_7} \right) = \varepsilon_9 \left( \frac{w_2}{(w_1 + w_2)^2} \right) + \varepsilon_{10} \left( -\frac{w_2}{(w_1 + w_2)^2} \right) \\ &= \varepsilon_9 \left( \frac{w_2}{(w_1 + w_2)^2} \right) + \varepsilon_{10} \left( -\frac{w_2}{(w_1 + w_2)^2} \right) \\ &= \left( \frac{w_2}{(w_1 + w_2)^2} \right) (\varepsilon_9 - \varepsilon_{10}) \end{aligned} \quad (2.22)$$

$$\begin{aligned} \varepsilon_8 &= \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) \left( \frac{\partial f_{11}}{\partial x_9} \right) \left( \frac{\partial f_9}{\partial x_8} \right) + \left( \frac{\partial E_p}{\partial x_{13}} \right) \left( \frac{\partial f_{13}}{\partial x_{11}} \right) \left( \frac{\partial f_{12}}{\partial x_{10}} \right) \left( \frac{\partial f_{10}}{\partial x_8} \right) \\ &= \varepsilon_{10} \left( \frac{\partial f_{10}}{\partial x_8} \right) + \varepsilon_9 \left( \frac{\partial f_9}{\partial x_8} \right) = \varepsilon_{10} \left( \frac{w_2}{(w_1 + w_2)^2} \right) + \varepsilon_9 \left( -\frac{w_2}{(w_1 + w_2)^2} \right) \\ &= \left( \frac{w_2}{(w_1 + w_2)^2} \right) (\varepsilon_{10} - \varepsilon_9) \end{aligned} \quad (2.23)$$

**e. Error pada lapisan ke-1**

*Propogasi error* yang menuju lapisan ke-1 dapat dirumuskan :

$$\varepsilon_3 = \varepsilon_7 \left( \frac{\partial f_7}{\partial x_3} \right) = \varepsilon_7 \mu_{B1}(x_2) \quad (2.24)$$

$$\varepsilon_4 = \varepsilon_8 \left( \frac{\partial f_8}{\partial x_4} \right) = \varepsilon_8 \mu_{B2}(x_2) \quad (2.25)$$

$$\varepsilon_5 = \varepsilon_7 \left( \frac{\partial f_7}{\partial x_5} \right) = \varepsilon_7 \mu_{A1}(x_1) \quad (2.26)$$

$$\varepsilon_6 = \varepsilon_8 \left( \frac{\partial f_8}{\partial x_6} \right) = \varepsilon_8 \mu_{A2}(x_1) \quad (2.27)$$

Selanjutnya *error* tersebut dapat digunakan untuk mencari informasi *error* terhadap parameter a dan c, diperoleh:

$$\varepsilon_{aik} = \frac{2(x_i - c_{ik})^2}{a_{ik}^3 \left( 1 + \left( \frac{x_i - c_{ik}}{a_{ik}} \right)^2 \right)^2} \quad (2.28)$$

$$\varepsilon_{cik} = \frac{2(x_i - c_{ik})}{a_{ik}^2 \left( 1 + \left( \frac{x_i - c_{ik}}{a_{ik}} \right)^2 \right)^2} \quad (2.29)$$

Dari persamaan 2. dan 2. di atas, perubahan nilai parameter  $a_{ij}$  dan  $c_{ij}$  ( $\Delta a_{ij}$  dan  $\Delta c_{ij}$ ), sebagai berikut :

$$\Delta a_{ik} = \eta \varepsilon_{aik} X_i, \quad \text{dan} \quad \Delta c_{ik} = \eta \varepsilon_{cik} X_i \quad (2.30)$$

$\eta$  : laju pembelajaran, terletak antara *interval* [0,1]. Sehingga nilai  $a_{ij}$  dan  $c_{ij}$  yang baru adalah :

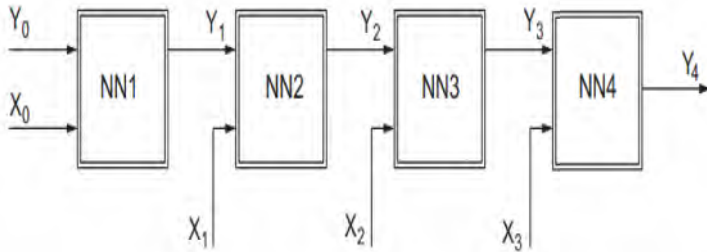
$$a_{ik} = a_{ik}(\text{lama}) + \Delta a_{ik} \quad (2.31)$$

$$c_{ik} = c_{ik}(\text{lama}) + \Delta c_{ik} \quad (2.32)$$

### 2.8.5 ANFIS unfolded in time

ANFIS unfolded in time merupakan sistem *neuro-fuzzy* yang digunakan untuk membangun *temporal multilayer* propagasi maju (*feed forward*) jaringan syaraf. Propagasi maju jaringan syaraf digandakan sebanyak T kali, dimana T menyatakan banyaknya *interval* yang diperlukan untuk menyelesaikan permasalahan.

Struktur dari ANFIS unfolded in time terdapat pada **gambar 2.6**. Setiap *box*  $NN_{(t-n)}, \dots, NN_{(t-1)}, NN_{(t)}$  merepresentasikan struktur ANFIS yang ada pada **gambar 2.5**. Pada **gambar 2.6** ANFIS digandakan sebanyak 4 kali, dengan *input* dari struktur *neuro-fuzzy* terdiri dari dua buah element, yaitu X dan Y.  $X_{(t-n)}, \dots, X_{(t-1)}, X_{(t)}$  merupakan data *time-series*, sementara  $Y_{(t-n)}$  merupakan target *output* pada waktu ke (t-n) dan untuk  $Y_{(t-(n+1))}, \dots, Y_{(t-1)}, Y_{(t)}$  merupakan *output* jaringan.

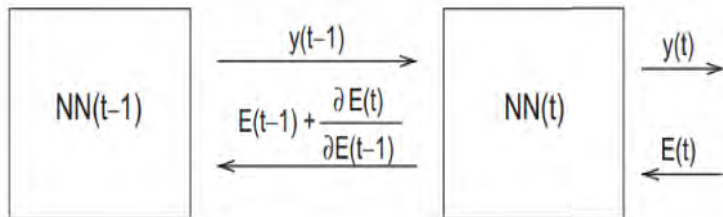


**Gambar 2.6** ANFIS\_unfolded\_in\_time

Berdasarkan **gambar 2.6**  $X_0$  dan  $Y_0$  merupakan *input* untuk  $NN_1$  yang menghasilkan *output*  $Y_1$ . Sementara *input* untuk  $NN_2$  adalah  $X_1$  dan  $Y_1$ , dimana  $X_1$  merupakan *input* dari luar sistem, sementara  $Y_1$  merupakan hasil dari *output* sistem sebelumnya. Hal yang sama dilakukan sampai  $NN_4$  yang menghasilkan *output*  $Y_4$  sebagai *output* dari sistem ANFIS\_unfolded\_in\_time. (Alpaslan, 2004).

### 2.8.6 Temporal Backpropagation

Parameter dalam jaringan akan diupdate berdasarkan *error output* dari sistem *neuro-fuzzy* dengan menggunakan *backpropagation*. Struktur metode ini terdapat pada **gambar 2.7**



**Gambar 2.7** Perhitungan *error interval* waktu

*Error* yang diperoleh dari jaringan ke- $t$  digunakan untuk mengupdate semua parameter yang ada pada jaringan  $NN_{(t)}$ . Tetapi untuk mengupdate parameter yang ada pada jaringan  $NN_{(t-1)}$ , dibutuhkan *error* yang ada pada jaringan ke- $t$  dengan melakukan *backpropagation* melalui **persamaan 2.33**:

$$\varepsilon_{(t-1)} + \frac{\partial \varepsilon_{(t)}}{\partial \varepsilon_{(t-1)}} \quad (2.33)$$

## 2.9 AKURASI HASIL PENGUJIAN

Sebagai hasil akhir sebuah penelitian, sering dicari akurasi hasil pengujian yang merupakan *prosentase* kebenaran dari hasil percobaan. Akurasi ini di dapat dihtung dengan :

$$\text{Akurasi} = \frac{\text{angka kebenaran}}{\text{angka pengujian}} \cdot 100\% \quad (2.34)$$

Selain menggunakan akurasi seperti ini metode yang dapat digunakan untuk mengukur tingkat akurasi adalah *Root Mean Square Error (RMSE)*. Semakin kecil *RMSE* maka semakin besar tingkat keberhasilan proses pelatihan.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (y_i - \bar{y}_1)^2}{N}} \quad (2.35)$$

Dimana :

N : banyak data

$y_i$  : *output* aktual/target jaringan

$\bar{y}_1$  : *output* nilai resiko/*output* jaringan



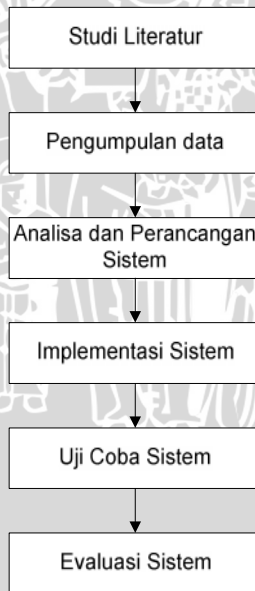
### BAB III

## METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas metode, rancangan yang digunakan dan langkah-langkah yang dilakukan dalam penelitian tentang prediksi tingkat pengangguran menggunakan *adaptive neuro fuzzy inference system*. Langkah-langkah yang akan dijalankan dalam penelitian ini, yaitu :

1. Mempelajari literatur yang berhubungan dengan unsur-unsur cuaca serta metode ANFIS.
2. Melakukan pengumpulan data tentang unsur-unsur cuaca.
3. Menganalisa dan melakukan perancangan system.
4. Mengimplementasikan dalam bentuk program berdasarkan analisis dan perancangan yang telah dilakukan.
5. Melakukan uji coba terhadap program yang telah dibuat.
6. Melakukan evaluasi output terhadap perangkat lunak.

Secara umum, tahapan dari penelitian ditunjukkan pada **gambar 3.1**



**Gambar 3.1** Tahapan penelitian

### 3.1 Studi Literatur

Dalam merealisasikan tujuan dan pemecahan masalah, penelitian ini dilakukan dengan studi literatur. Teori-teori mengenai konsep *ANFIS*, dan prediksi pengangguran yang digunakan sebagai dasar penelitian diperoleh dari buku dan internet. Kemudian data-data yang telah diperoleh diubah ke bentuk yang dapat diproses oleh program sehingga dapat digunakan untuk melakukan analisa.

### 3.2 Pengumpulan Data-data

Data yang digunakan untuk penelitian ini merupakan data-data dari : jumlah penduduk usia produktif, jumlah angkatan kerja, jumlah pekerja, TPAK, TKK dan Indeks SDM. Yang berisi data *time series* dari tahun 2000 s/d tahun 2011 yang berasal dari BPS dan Disnakaertrans Malang.

### 3.3 Analisa dan Perancangan Sistem

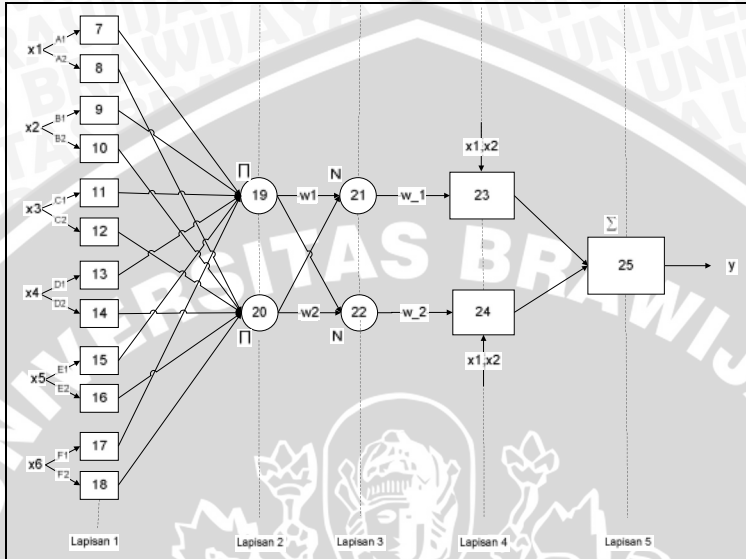
Pada tahap ini, dilakukan analisa dan perancangan mengenai prediksi tingkat pengangguran menggunakan metode *ANFIS* dengan aturan penalaran sistem inferensi *fuzzy* Takagi Sugeno orde 1.

#### 3.3.1 Deskripsi Sistem

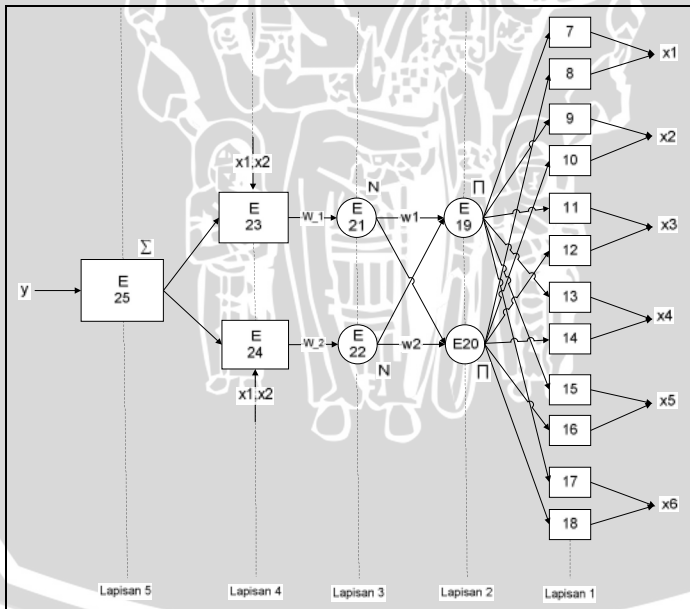
Sistem yang akan dibangun ini merupakan perangkat lunak yang digunakan untuk melakukan prediksi terhadap tingkat pengangguran. Hasil prediksi berupa nilai yang mewakili keadaan tingkat pengangguran, yaitu cukup rendah, rendah, sedang, cukup tinggi dan tinggi. Pada tahap awal, sistem harus memiliki data-data parameter pengangguran sebagai basis pengetahuan. Data parameter ini berupa data numerik, sehingga dapat digunakan untuk proses pembelajaran (*training*) dengan menggunakan metode *ANFIS\_unfolded\_in\_time*. Pembelajaran metode ini akan menggunakan algoritma *hybrid*, yaitu *Least Square Error (LSE)* sebagai algoritma *feed forward* dan *steepest-descent* sebagai algoritma *feed back*.

#### 3.3.2 Desain Arsitektur Sistem

Pada penelitian sekripsi ini, desain arsitektur sistem mengacu pada dasar arsitektur alur maju yang ditunjukkan pada **gambar 3.2** dan alur mundur pada **gambar 3.3**.



**Gambar 3.2** Arsitektur alur maju sistem ANFIS



**Gambar 3.3** Arsitektur alur mundur sistem ANFIS

### 3.3.3 Rancangan Penelitian

Berdasarkan desain arsitektur diatas terdapat 6 buah parameter tingkat pengangguran sebagai input awal, yaitu : jumlah penduduk usia produktif, jumlah angkatan kerja, jumlah pekerja,TPAK, TKK dan Indeks SDM. Dari parameter tersebut akan dibentuk 5 buah *cluster* berdasarkan output kondisi tingkat pengangguran, yaitu : sangat rendah, rendah, sedang, tinggi dan sangat tinggi.

#### 3.3.3.1 Pelatihan dan Pengujian Sistem ANFIS

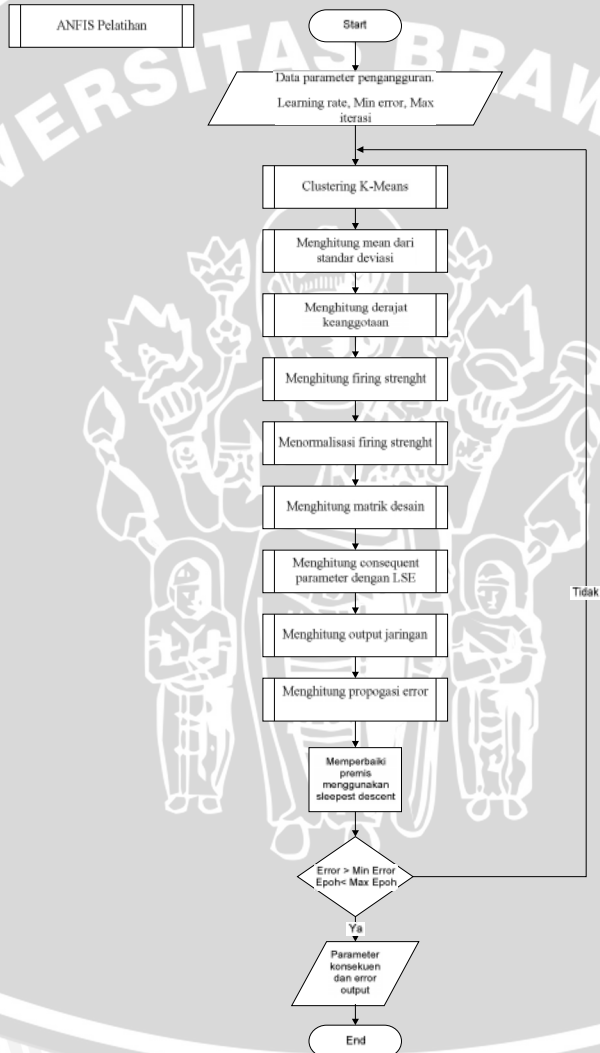
Proses untuk memprediksi tingkat pengangguran menggunakan *ANFIS* dilakukan dengan melakukan pelatihan terhadap data latih terlebih dahulu kemudian melakukan pengujian terhadap data. Proses pelatihan terhadap data latih akan menghasilkan nilai *error output* prediksi tingkat pengangguran, secara umum meliputi langkah-langkah:

- a. Lakukan pengelompokan data dengan algoritma *fuzzy K-Means*.
- b. Lakukan perhitungan standar deviasi (a) dan *mean* (c) berdasarkan *cluster* yang terbentuk.
- c. Lakukan perhitungan derajat keanggotaan dari tiap-tiap masukan dengan fungsi keanggotaan *Gaussian Bell*.
- d. Lakukan pengalihan dengan derajat keanggotaan dan parameter koefisien. Parameter ini digunakan untuk menghitung prediksi tingkat pengangguran pada pengujian data. Rancangan sistem pelatihan *ANFIS* keseluruhan untuk mendapatkan nilai kuat penyuluhan (*fire strength*),  $w_i$ .
- e. Lakukan *normalisasi fire strength* untuk mendapatkan nilai kuat penyuluhan ternormalisasi.
- f. Lakukan perhitungan *matriks* desain yang digunakan untuk *inputan LSE*.
- g. Lakukan penentuan parameter-parameter konsekuen yang adaptif, dengan metode *LSE*.
- h. Lakukan perhitungan keluaran keseluruhan dari semua sinyal yang dihasilkan sebagai penjumlahan semua sinyal yang datang dan dianggap sebagai *output* jaringan.
- i. Lakukan perbandingan *output* jaringan dengan target *output* untuk mencari *error* jaringan.
- j. Jika *error* yang didapatkan masih lebih besar dari *error* yang diharapkan, maka dilakukan perbaikan parameter premis dengan

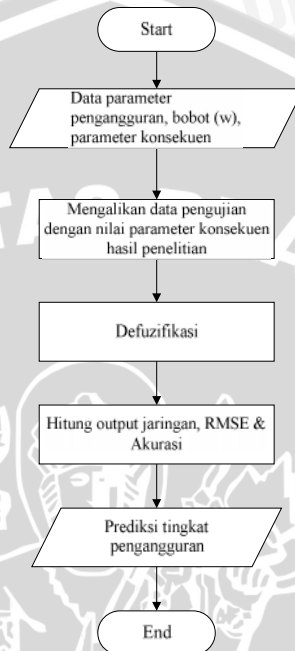
algoritma *steepest descend*.

- k. Lakukan perbandingan target *output* dengan *output* jaringan yang terbentuk yang kemudian dicocokkan dengan nilai *error* jaringan

Untuk lebih jelasnya, *flowchart ANFIS* pelatihan digambarkan seperti pada **gambar 3.4** dan untuk *flowchart* tahap pengujian digambarkan pada **gambar 3.5**.



**Gambar 3.4** Diagram Alir pelatihan sistem *ANFIS*



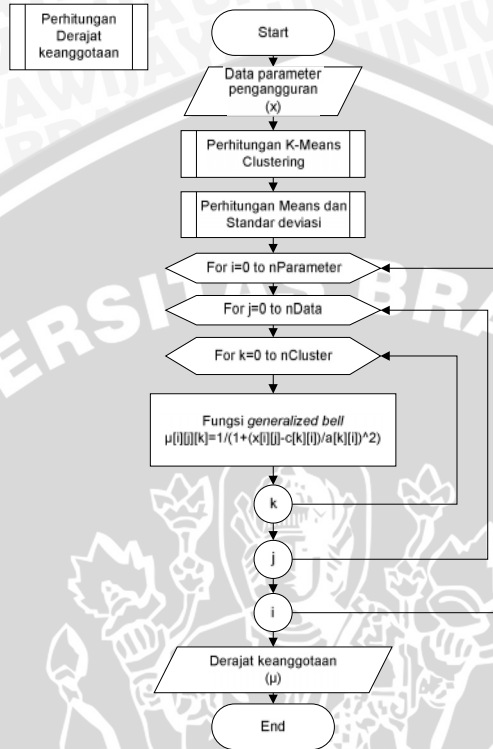
**Gambar 3.5** Diagram Alir pengujian sistem ANFIS

### 3.3.3.2 Perhitungan Derajat Keanggotaan

Proses perhitungan derajat keanggotaan ini memiliki beberapa langkah, yaitu :

1. Memasukkan data parameter pengangsuran.
2. Masuk pada proses perhitungan *k-means clustering* dan perhitungan *mean* dan standar deviasi.
3. Melakukan perulangan untuk parameter, *clustering* dan jumlah data yang digunakan.
4. Melakukan perhitungan derajat keanggotaan dengan fungsi *generalized bell*, seperti pada **persamaan 2.1**.
5. Didapatkan nilai *output* dari lapisan pertama pada sistem ANFIS.

Proses perhitungan derajat keanggotaan digambarkan pada diagram alir pada **gambar 3.6**:



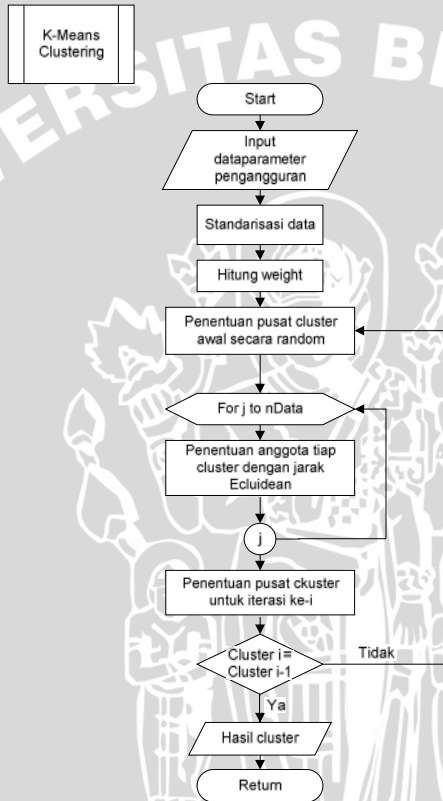
**Gambar 3.6** Diagram Alir Derajat Keanggotaan

### 3.3.3.3 Perhitungan *K-Means Clustering*

Pada proses *clustering*, menggunakan *K-Means Clustering* dengan mengelompokkannya menjadi 3 *cluster*. Langkah-langkah dalam melakukan *K-Means Clustering* adalah sebagai berikut :

1. Melakukan *load* data
2. Menstandarisasi data parameter pengangguaran seperti pada **persamaan 2.2**.
3. Melakukan perhitungan bobot dengan mencari rata-rata dari tiap *input* seperti pada **persamaan 2.4**.
4. Menentukan pusat *cluster* awal dengan nilai *random* dalam *interval* nilai *minimum* dan *maximum* pada nilai bobot (*weight*).
5. Menentukan anggota tiap *cluster* berdasarkan jarak *Euclidean* terkecil
6. Menentukan pusat *cluster* baru dengan cara mencari *mean* dari tiap-tiap anggota *cluster* yang terbentuk.

7. Iterasi terus dilakukan sampai anggota *cluster* baru yang terbentuk tidak mengalami perubahan dibandingkan dengan iterasi sebelumnya.
8. Telah didapatkan anggota hasil tiap *cluster*.  
Langkah-langkah melakukan *clustering* dengan *K-Means clustering*, seperti pada **gambar 3.7**



**Gambar 3.7** Diagram Alir *K-Means Clustering*

### 3.3.3.4 Perhitungan *Mean* dan *Standar Deviasi*

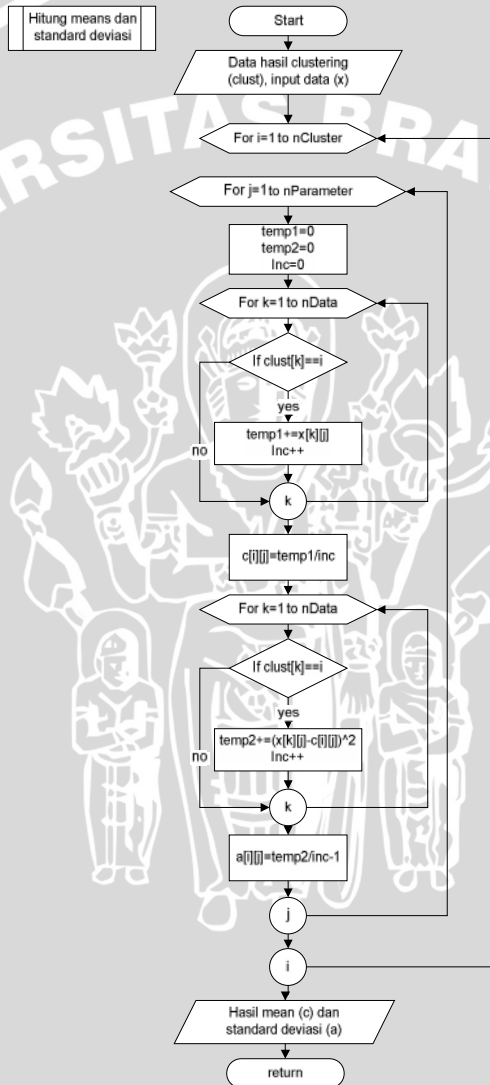
Perhitungan *mean* dan standar deviasi ini digunakan untuk nilai awal  $a$  dan  $c$  yang akan digunakan pada **persamaan 2.1**, langkah-langkahnya adalah sebagai berikut :

1. Menggunakan data hasil *clustering* dengan *K-Means*.
2. Kemudian dilakukan perhitungan untuk mencari nilai *mean* (**persamaan 2.5**) dan nilai standar deviasi (**persamaan 2.6**).



3. Melakukan perulangan hingga iterasi selesai dan didapatkan nilai *mean* dan standar deviasi.

**Gambar 3.8** merupakan diagram alir perhitungan *means* dan standar deviasi.



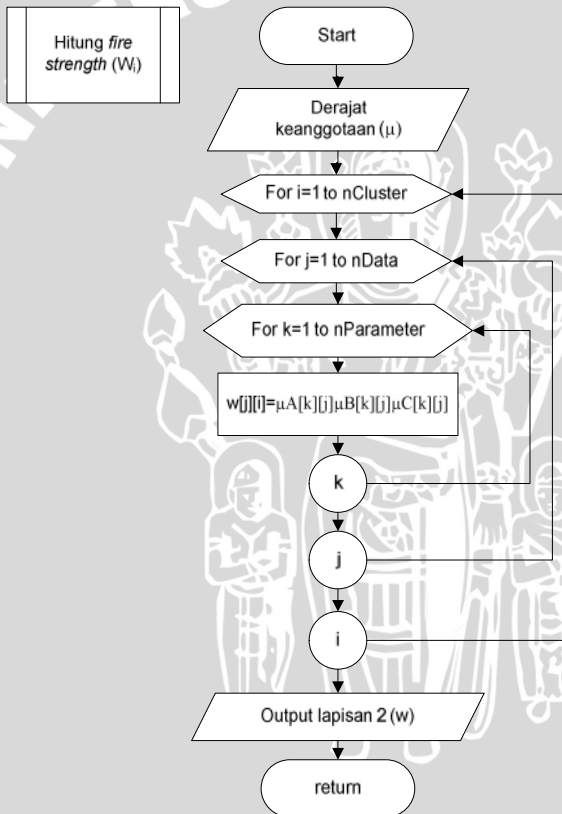
**Gambar 3.8** Diagram Alir *Means* dan Standar Deviasi

### 3.3.3.5 Perhitungan *Fire Strength*

Perhitungan *fire strength* merupakan perhitungan untuk lapisan kedua, langkah dari perhitungannya sebagai berikut :

1. Menggunakan *input* dari lapisan pertama.
2. Melakukan perulangan untuk parameter, *clustering* dan jumlah data yang digunakan.
3. Melakukan perhitungan seperti pada **persamaan 2.11**.
4. Dilakukan iterasi hingga selesai dan didapatkan nilai bobot ( $w_i$ ).

**Gambar 3.9** merupakan diagram alir perhitungan *fire strength*.



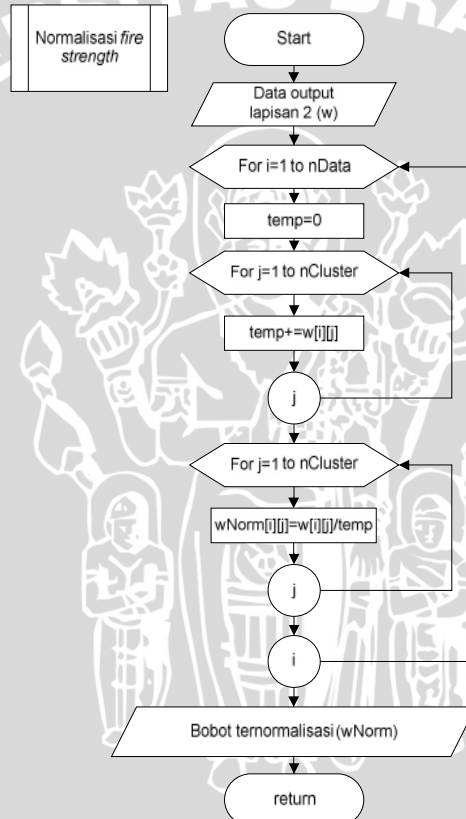
**Gambar 3.9** Diagram Alir *Fire Strength*

### 3.3.3.6 Penormalisasian *Fire Strength*

Penormalisasian *fire strength* merupakan perhitungan pada lapisan ketiga, langkah-langkahnya sebagai berikut :

1. Menggunakan nilai *fire strength* yang didapatkan pada lapisan ketiga.
2. Melakukan perulangan untuk parameter, *clustering*, dan jumlah data yang digunakan.
3. Menghitung normalisasi bobot seperti pada **persamaan 2.12**.
4. Melakukan iterasi hingga selesai, dan didapatkan nilai bobot ternormalisasi.

**Gambar 3.10** merupakan diagram alir penormalisasian *fire strength*.



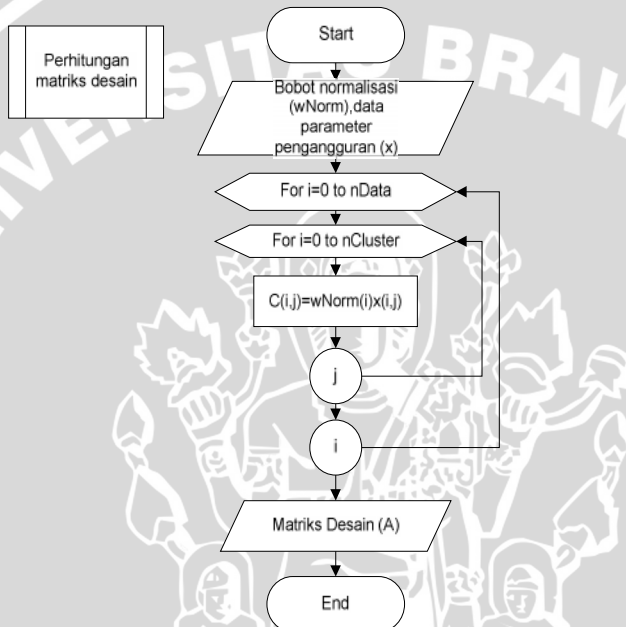
**Gambar 3.10** Diagram Alir Penormalisasian *Fire Strength*

### 3.3.3.7 Perhitungan Matriks Desain

Perhitungan *matriks* desain merupakan perhitungan pada lapisan keempat, langkah-langkahnya adalah sebagai berikut :

1. Memasukkan *input* bobot yang ternormalisasi dan data parameter pengangguran.
2. Melakukan iterasi.
3. Melakukan perhitungan pada **persamaan 2.13**, hingga didapatkan *matriks* desain.

**Gambar 3.11** merupakan diagram alir untuk perhitungan *matriks* desain.



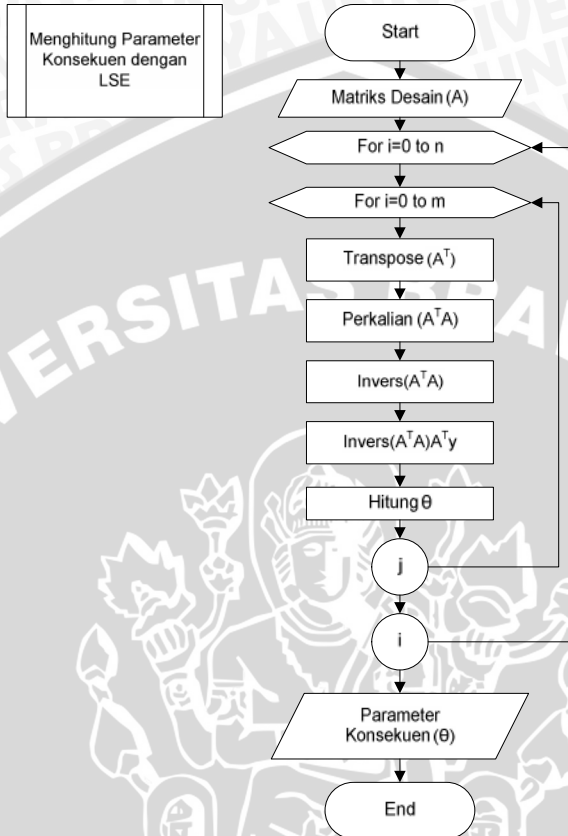
**Gambar 3.11** Diagram Alir Perhitungan *Matriks Desain*

### 3.3.3.8 Perhitungan Parameter Konsekuen

Langkah dari perhitungan parameter konsekuen dengan *LSE* adalah sebagai berikut :

1. Memasukkan nilai *matriks* desain (*output* lapisan 4).
2. Transpose *matriks* desain.
3. Perkalian *matriks* desain dengan transposenya ( $P_0$ )
4. Invers nilai  $P_0$
5. Perkalian nilai  $P_0$  dengan transpose *matriks* desain dan target *output*.
6. Parameter konsekuen telah didapatkan ( $\theta$ )

**Gambar 3.12** merupakan diagram alir untuk perhitungan parameter konsekuen dengan *LSE* :



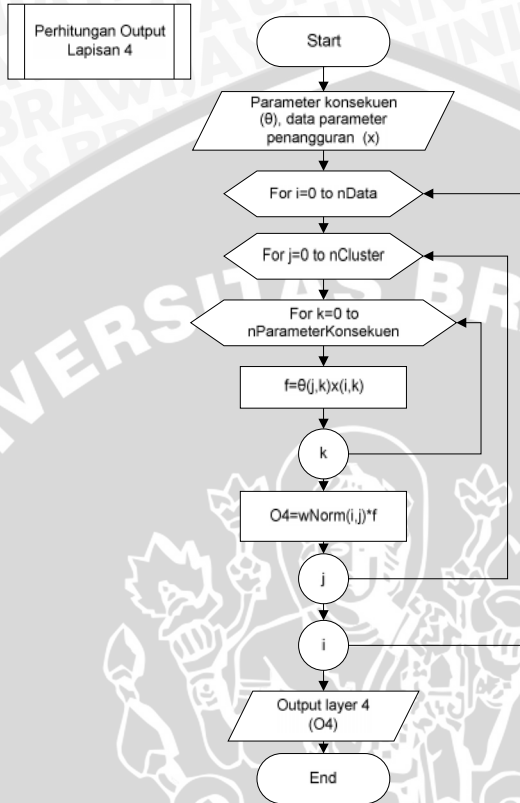
**Gambar 3.12** Diagram Alir Perhitungan Parameter Konsekuensi dengan *LSE*

### 3.3.3.9 Perhitungan Output Lapisan 4

Perhitungan *output* lapisan ke-4 ini memiliki beberapa langkah, yaitu :

1. Menggunakan Parameter konsekuensi dan data parameter pengangguran sebagai *input* data.
2. Melakukan perhitungan lapisan ke-4 seperti pada **persamaan 2.13**.
3. Didapatkan nilai *output* dari lapisan ke-4.

**Gambar 3.13** merupakan diagram alir untuk perhitungan *output* lapisan ke-4 :



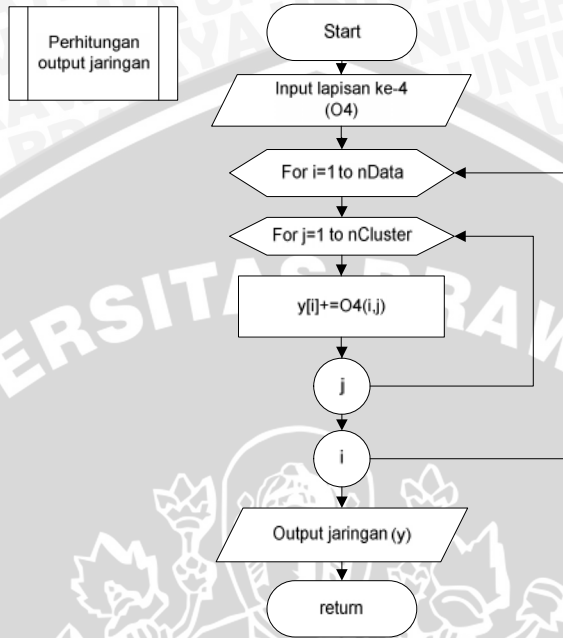
**Gambar 3.13** Diagram Alir Perhitungan *Output* Lapisan ke-4

### 3.3.3.10 Perhitungan Output Jaringan

Parameter *output* jaringan memiliki beberapa langkah yaitu:

1. Menggunakan hasil *output* lapisan ke-4.
2. Melakukan perhitungan seperti pada **persamaan 2.14**.
3. Didapatkan nilai *output* jaringan.

**Gambar 3.14** merupakan diagram alir untuk perhitungan *output* jaringan :



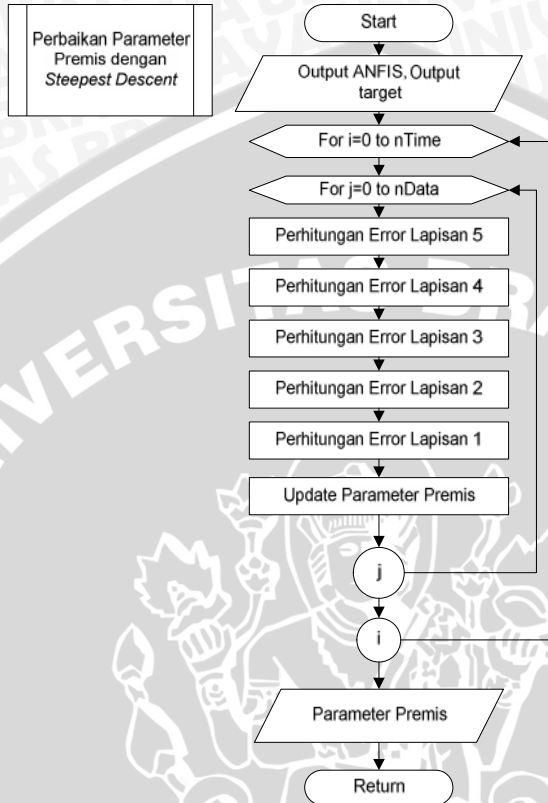
**Gambar 3.14** Diagram Alir Perhitungan *Output Jaringan*

### 3.3.3.11 Perbaikan Parameter Premis

Pebbaikan parameter premis ini memiliki beberapa langkah, yaitu :

1. Menggunakan *output* jaringan sebagai *input* data.
2. Dilakukan perhitungan *error* di lapisan kelima dengan **persamaan 2.18**.
3. Menghitung kembali *error* pada lapisan keempat dengan **persamaan 2.19 dan 2.20**.
4. Menggunakan **persamaan 2.21 dan 2.22** untuk menghitung *error* pada lapisan ketiga.
5. Menghitung kembali *error* lapisan kedua dengan **persamaan 2.23 dan 2.24**.
6. Untuk lapisan pertama perhitungan *error* menggunakan **persamaan 2.25, 2.26, 2.27, 2.28 dan 2.29**.
7. Dilakukan update parameter premis.

**Gambar 3.15** Merupakan diagram alir perbaikan parameter premis



**Gambar 3.15** Diagram Alir Perbaikan Parameter *Premis*

### 3.4 Perhitungan Manual

Pada sub-bab ini akan menampilkan contoh perhitungan manual untuk prediksi kondisi pengangguran dengan *ANFIS\_Unfolded\_In\_Time*. Perhitungan manual yang akan dilakukan ini menggunakan 2 waktu yang berbeda, misal : tahun 2000 dan 2001, dimana setiap waktu akan dilakukan proses *ANFIS*.

#### 3.4.1 Perhitungan ANFIS TAHUN 2001

Perhitungan *ANFIS* pada tahun 2001 ini akan menggunakan data dengan 5 buah variabel *input*, yaitu :  $x_1$ ,  $x_2$ ,  $x_3$ ,  $x_4$ ,  $x_5$  dan  $x_6$ , dan  $y$  sebagai target *output* pada tahun 2001 ini. Berikut **tabel 3.1** yang merupakan data yang akan digunakan.



**Tabel 3.1** Data tahun 2001

X1	X2	X3	X4	X5	X6	Y
4,9634	2,8336	2,5947	0,000476	0,000763	2,39E-05	7,67E-09
5,0035	2,8304	2,5431	0,000471	0,000749	2,87E-05	9,41E-09
4,8035	2,8094	2,3531	0,000487	0,000698	4,56E-05	1,62E-08
4,7623	2,9005	2,4352	0,000508	0,0007	4,65E-05	1,59E-08
4,6767	2,8954	2,5346	0,000516	0,000729	3,61E-05	1,19E-08
5,1346	2,836	2,5673	0,00046	0,000754	2,69E-05	8,72E-09
4,5606	2,7564	2,3436	0,000504	0,000709	4,13E-05	1,47E-08
4,6434	2,9731	2,4632	0,000534	0,00069	5,10E-05	1,73E-08
4,7423	2,6987	2,3451	0,000474	0,000724	3,54E-05	1,26E-08
4,8308	2,8534	2,4312	0,000492	0,00071	4,22E-05	1,45E-08
4,6634	2,8356	2,3451	0,000507	0,000689	4,91E-05	1,74E-08
4,8574	2,7696	2,3658	0,000475	0,000712	4,04E-05	1,42E-08

Dari 12 data yang ada akan dikelompokkan menjadi 5 buah *cluster* yang didasarkan pada kondisi pengangguran yaitu cukup rendah, rendah, sedang, cukup tinggi dan tinggi. Untuk meng*cluster* data akan digunakan algoritma *K-Means*, berikut langkah-langkah yang dilakukan:

1. Melakukan standarisasi data *input* menggunakan **persamaan 2.2**, dan dihitung bobot dari data yang telah distandarisasi dengan rata-rata selanjutnya duhitung bobotnya menggunakan **persamaan 2.4**, seperti pada **tabel 3.2**.

$$x_{ij} = \frac{x_{ij} - x_{Min_j}}{x_{Max_j} - x_{Min_j}} \quad \text{dan} \quad \text{bobot}_1 = w = \frac{x_{ij}}{n} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

**Tabel 3.2** Standarisasi data dan bobot

X1	X2	X3	X4	X5	X6	BOBOT
0,702	0,492	1	0,216	1	0	0,568
0,772	0,480	0,795	0,149	0,811	0,177	0,530
0,423	0,403	0,038	0,365	0,122	0,801	<b>0,359</b>

0,351	0,735	0,365	0,649	0,149	0,834	<b>0,514</b>
0,202	0,717	0,761	0,757	0,541	0,450	0,571
1	0,500	0,891	0	0,878	0,111	0,563
0	0,210	0	0,595	0,270	0,642	0,286
0,144	1	0,476	1	0,014	1	0,606
0,317	0	0,006	0,189	0,473	0,424	<b>0,235</b>
0,471	0,564	0,349	0,432	0,284	0,675	0,462
0,179	0,499	0,006	0,635	0	0,930	0,375
0,517	0,258	0,088	0,203	0,311	0,609	0,331

2. Melakukan iterasi pertama dengan menggunakan rumus *Euclidean* seperti pada **persamaan 2.3**.

$$d(x, y) = ||x - y|| = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2}$$

Dengan pusat *cluster* awal ditentukan secara acak antara rentang nilai *minimum* dan maksimum dari bobot yang diperoleh. Hasil dari iterasi pertama ditunjukkan pada **tabel 3.4**. Nilai pusat *cluster* awal yang diperoleh secara *random* ditunjukkan pada **table 3.3**.

**Tabel 3.3** Pusat *Cluster* Pertama

RANDOM				
C1	C2	C3	C4	C5
0,2348	0,3586	0,4625	0,5138	0,6057

**Tabel 3.4** Data Iterasi Pertama

BOBOT	ITERASI 1				
	C1	C2	C3	C4	C5
0,5683	0,3335	0,2097	0,1058	0,0545	<b>0,0374</b>
0,5304	0,2956	0,1718	0,0679	<b>0,0166</b>	0,0753
<b>0,3586</b>	0,1238	<b>0</b>	0,1039	0,1552	0,2471
<b>0,5138</b>	0,2790	0,1552	0,0513	<b>0</b>	0,0919
0,5712	0,3364	0,2126	0,1087	0,0574	<b>0,0345</b>

0,5634	0,3286	0,2048	0,1009	0,0496	<b>0,0423</b>
0,2862	<b>0,0514</b>	0,0724	0,1763	0,2276	0,3195
0,6057	0,3709	0,2471	0,1432	0,0919	<b>0</b>
<b>0,2348</b>	<b>0</b>	0,1238	0,2277	0,2790	0,3709
0,4625	0,2277	0,1039	<b>0</b>	0,0513	0,1432
0,3748	0,1400	<b>0,0162</b>	0,0877	0,1390	0,2309
0,3310	0,0962	<b>0,0276</b>	0,1315	0,1828	0,2747

3. Iterasi terus dijalankan untuk menguji kemiripan *cluster* dengan iterasi sebelumnya. Pusat *cluster* iterasi kedua diperoleh dengan menggunakan rata-rata. Hasil dari iterasi kedua ditunjukkan pada **tabel 3.6**. Nilai pusat *cluster* iterasi kedua ditunjukkan pada **table 3.5**

**Tabel 3.5** Pusat *Cluster* Kedua

C1	C2	C3	C4	C5
0,2605	0,3548	0,4625	0,5221	0,5771

**Tabel 3.6** Data Iterasi Kedua

BOBOT	ITERASI 2				
	C1	C2	C3	C4	C5
0,5683	0,3078	0,2135	0,1058	0,0462	<b>0,0088</b>
0,5304	0,2699	0,1756	0,0679	<b>0,0083</b>	0,0467
<b>0,3586</b>	0,0981	<b>0,0038</b>	0,1039	0,1635	0,2185
<b>0,5138</b>	0,2533	0,1590	0,0513	<b>0,0083</b>	0,0633
0,5712	0,3107	0,2164	0,1087	0,0491	<b>0,0059</b>
0,5634	0,3029	0,2086	0,1009	0,0413	<b>0,0137</b>
0,2862	<b>0,0257</b>	0,0686	0,1763	0,2359	0,2909
0,6057	0,3452	0,2509	0,1432	0,0836	<b>0,0286</b>
<b>0,2348</b>	<b>0,0257</b>	0,1200	0,2277	0,2873	0,3423
0,4625	0,2020	0,1077	<b>0,0000</b>	0,0596	0,1146
0,3748	0,1143	0,0877	<b>0,0200</b>	0,1473	0,2023
0,3310	0,0705	<b>0,0238</b>	0,1315	0,1911	0,2461

Perbandingan dari hasil iterasi pertama dan iterasi kedua menunjukkan tidak ada perubahan *cluster*, sehingga iterasi tidak perlu dilakukan lagi. Setelah iterasi tidak ada perubahan maka tahap ini di hentikan dan hasil dari *clustering* ditunjukkan pada **tabel 3.7**

**Tabel 3.7** Data Hasil *Clustering*

X1	X2	X3	X4	X5	X6	C
4,9634	2,8336	2,5947	0,000476	0,000763	2,39E-05	5
5,0035	2,8304	2,5431	0,000471	0,000749	2,87E-05	4
4,8035	2,8094	2,3531	0,000487	0,000698	4,56E-05	2
4,7623	2,9005	2,4352	0,000508	0,0007	4,65E-05	4
4,6767	2,8954	2,5346	0,000516	0,000729	3,61E-05	5
5,1346	2,836	2,5673	0,00046	0,000754	2,69E-05	5
4,5606	2,7564	2,3436	0,000504	0,000709	4,13E-05	1
4,6434	2,9731	2,4632	0,000534	0,00069	5,10E-05	5
4,7423	2,6987	2,3451	0,000474	0,000724	3,54E-05	1
4,8308	2,8534	2,4312	0,000492	0,00071	4,22E-05	3
4,6634	2,8356	2,3451	0,000507	0,000689	4,91E-05	3
4,8574	2,7696	2,3658	0,000475	0,000712	4,04E-05	2

### 3.4.1.1 Perhitungan Lapisan ke-1

Untuk menghitung derajat keanggotaan dari masing-masing *input* dengan fungsi *generalized bell*, akan dicari terlebih dahulu nilai *a*, yaitu standar deviasi dan *c*, yaitu nilai *mean* berdasarkan *cluster* yang terbentuk. Berikut **tabel 3.8** dan **tabel 3.9** hasil perhitungan *mean* dan standar deviasi. **Persamaan 2.5** untuk menghitung *mean*.

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

**Tabel 3.8** Nilai *c* (*mean*)

MEAN ( <u>x</u> )					
X1	X2	X3	X4	X5	X6
4,651E+00	2,728E+00	2,344E+00	4,890E-04	7,165E-04	3,835E-05

4,830E+00	2,790E+00	2,359E+00	4,810E-04	7,050E-04	4,300E-05
4,747E+00	2,845E+00	2,388E+00	4,995E-04	6,995E-04	4,565E-05
4,883E+00	2,865E+00	2,489E+00	4,895E-04	7,245E-04	3,760E-05
4,855E+00	2,834E+00	2,379E+00	5,050E-04	7,000E-04	4,545E-05

**Persamaan 2.6** untuk menghitung standar deviasi

$$\sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{(n - 1)}}$$

**Tabel 3.9** Nilai a (standar deviasi)

STANDAR DEVIASI ( $\sigma$ )					
X1	X2	X3	X4	X5	X6
1,285E-01	4,080E-02	1,061E-03	2,121E-05	1,061E-05	4,172E-06
3,811E-02	2,814E-02	8,980E-03	8,485E-06	9,899E-06	3,677E-06
1,671E-01	7,879E-02	7,317E-02	2,823E-05	1,676E-05	6,152E-06
1,706E-01	4,957E-02	7,630E-02	2,616E-05	3,465E-05	1,259E-05
1,360E-01	5,083E-02	1,119E-01	2,062E-05	2,949E-05	1,015E-05

Setelah ditemukan nilai dari a dan c dari masing-masing *cluster*, akan dicari derajat keanggotaan untuk tiap-tiap *cluster* dengan menggunakan **persamaan 2.1**.

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

Hasil perhitungan derajat keanggotaan atau output lapisan pertama terdapat pada **tabel 3.10**.

### 3.4.1.2 Perhitungan Lapisan ke-2

Setelah mendapatkan nilai derajat keanggotaan, kemudian dilanjutkan mencari nilai *fire strength* seperti **persamaan 2.11**. Dari perhitungan didapatkan *output* seperti pada **tabel 3.11**

$$O_{2i} = w_i = \mu_A(x), \mu_B(y) \quad i=1,2,..n$$

**Tabel 3.11** Output Lapisan ke-2

NILAI FIRE STRENGTH ( $w_i$ )					
O2,1	O2,2	O2,3	O2,4	O2,5	O2,6
1,482E-03	2,405E-05	7,259E-11	3,319E-12	7,399E-14	1,580E-14
6,024E-03	1,040E-04	1,451E-08	9,416E-13	6,763E-13	1,033E-13
4,677E-03	3,673E-06	2,280E-10	5,741E-13	5,030E-13	5,772E-14
3,337E-04	5,265E-07	9,965E-12	9,571E-14	4,660E-14	1,169E-14
1,326E-03	1,134E-06	2,303E-10	2,220E-13	4,613E-14	6,223E-15
5,387E-04	2,321E-06	3,088E-09	4,345E-13	2,490E-12	2,726E-13
8,165E-03	2,423E-06	7,334E-10	2,924E-13	3,910E-13	3,019E-14
4,719E-02	1,112E-05	9,605E-09	2,573E-12	3,350E-13	5,088E-14
4,569E-01	1,387E-05	4,672E-10	3,818E-12	1,795E-12	3,790E-13
7,533E-06	1,201E-08	7,614E-11	1,747E-15	1,481E-15	5,401E-15
5,671E-07	2,786E-11	1,838E-13	6,113E-17	5,955E-18	1,163E-15
2,207E-10	8,071E-15	3,907E-18	1,075E-19	3,718E-21	2,826E-17

### 3.4.1.3 Perhitungan Lapisan ke-3

Perhitungan pada lapisan ke-3 ini adalah mencari nilai *fire strength* ternormalisasi (*normalized fire strength*). Untuk mencari nilai *normalized fire strength* digunakan **persamaan 2.12**. Hasil perhitungan ini terdapat pada **tabel 3.12**

$$O_{3,i} = \frac{w_i}{w_1 + w_2}$$

**Tabel 3.12** Output Lapisan ke-3

NORMALISASI FIRE STRENGTH					
O3,1	O3,2	O3,3	O3,4	O3,5	O3,6
9,840E-01	1,597E-02	4,820E-08	2,204E-09	4,913E-11	1,049E-11
9,830E-01	1,697E-02	2,367E-06	1,536E-10	1,104E-10	1,685E-11
9,992E-01	7,847E-04	4,871E-08	1,226E-10	1,075E-10	1,233E-11
9,984E-01	1,575E-03	2,981E-08	2,863E-10	1,394E-10	3,497E-11
9,991E-01	8,540E-04	1,735E-07	1,672E-10	3,475E-11	4,688E-12
9,957E-01	4,290E-03	5,707E-06	8,030E-10	4,603E-09	5,038E-10
9,997E-01	2,966E-04	8,980E-08	3,580E-11	4,788E-11	3,697E-12

9,998E-01	2,356E-04	2,035E-07	5,451E-11	7,097E-12	1,078E-12
1,000E+00	3,036E-05	1,023E-09	8,357E-12	3,929E-12	8,295E-13
9,984E-01	1,592E-03	1,009E-05	2,316E-10	1,963E-10	7,159E-10
1,000E+00	4,912E-05	3,241E-07	1,078E-10	1,050E-11	2,051E-09
1,000E+00	3,656E-05	1,770E-08	4,869E-10	1,684E-11	1,280E-07

#### 3.4.1.4 Perhitungan Lapisan ke-4

Perhitungan lapisan ke-4 ini dilakukan dengan dua tahapan, yaitu mencari *matriks* desain sebagai *input* untuk perhitungan *LSE* dan pencarian parameter konsekuen dengan *LSE*. Perhitungan *LSE* dan hasil perhitungan *matriks* desain dan perhitungan parameter konsekuen terdapat terdapat pada **tabel 3.13** dan **3.14**.



UNIVERSITAS BRAWIJAYA





Setelah ditemukan parameter konsekuensi selanjutnya dicari *output* lapisan ke-4 melalui **persamaan 2.13**. Hasil dari perhitungan lapisan ke-4 terdapat pada **tabel 3.15**.

$$O_{4,i} = \bar{w}_i y_i = w_i (c_{i1}x_1 + c_{i2}x_2 + \dots + c_{in}x_n + c_{io})$$

**Tabel 3.15** *Output* Lapisan ke-4

OUTPUT LAPISAN 4 (y <sub>i</sub> )					
O4,1	O4,2	O4,3	O4,4	O4,5	O4,6
1,55E+01	2,52E-01	7,60E-07	3,47E-08	7,74E-10	1,65E-10
1,55E+01	2,49E-01	7,54E-07	3,44E-08	7,65E-10	1,64E-10
1,49E+01	2,41E-01	7,27E-07	3,32E-08	7,38E-10	1,58E-10
1,53E+01	2,48E-01	7,50E-07	3,43E-08	7,62E-10	1,63E-10
1,56E+01	2,52E-01	7,62E-07	3,48E-08	7,73E-10	1,65E-10
1,55E+01	2,51E-01	7,58E-07	3,46E-08	7,69E-10	1,65E-10
1,48E+01	2,38E-01	7,19E-07	3,28E-08	7,30E-10	1,56E-10
1,66E+01	2,53E-01	7,63E-07	3,49E-08	7,75E-10	1,66E-10
2,51E+01	2,35E-01	7,11E-07	3,25E-08	7,22E-10	1,54E-10
1,51E+01	2,46E-01	7,43E-07	3,39E-08	7,54E-10	1,61E-10
1,49E+01	2,41E-01	7,30E-07	3,33E-08	7,41E-10	1,58E-10
1,47E+01	2,39E-01	7,23E-07	3,30E-08	7,35E-10	1,57E-10

### 3.4.1.5 Perhitungan Lapisan ke-5

Perhitungan pada lapisan kelima ini adalah penjumlahan seluruh sinyal yang datang. Perhitungan ini menggunakan **persamaan 2.14**. Hasil dari perhitungan ini terdapat pada **tabel 3.16**.

$$O_{5,i} = \sum_i \bar{w}_i y_i = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

**Tabel 3.16** *Output* Lapisan ke-5

OUTPUT
O5
1,58E+01
1,58E+01
1,52E+01

1,55E+01
1,58E+01
1,57E+01
1,51E+01
1,69E+01
2,53E+01
1,54E+01
1,51E+01
1,50E+01

*Output* jaringan yang diperoleh dari *ANFIS* Tahun 2001 akan digunakan sebagai target *input ANFIS* Tahun 2002

### 3.4.2 Perhitungan ANFIS Tahun 2002

Langkah-langkah perhitungan *ANFIS* untuk Tahun 2001 sama dengan perhitungan *ANFIS* yang pertama, namun perhitungan untuk Tahun ke-(t+1) menggunakan *output* target dari hasil *ANFIS* Tahun ke-t. Berikut **Tabel 3.17** merupakan data untuk Tahun 2002 dengan target *output* dari hasil *ANFIS* Tahun 2001.

**Tabel 3.17** Data Tahun 2002

X1	X2	X3	X4	X5	OUTPUT
					O5
4,9037	2,7933	2,4934	0,000475	0,000744	1,58E+01
5,0437	2,7732	2,3456	0,000458	0,000705	1,58E+01
4,9865	2,8093	2,4563	0,000469	0,000729	1,52E+01
4,7658	2,7765	2,5432	0,000485	0,000763	1,55E+01
5,2098	2,6543	2,4561	0,000425	0,000771	1,58E+01
4,779	2,7609	2,3456	0,000481	0,000708	1,57E+01
4,9631	2,6543	2,2478	0,000446	0,000706	1,51E+01
4,9786	2,5836	2,3598	0,000432	0,000761	1,69E+01
5,1345	2,7689	2,4367	0,000449	0,000733	2,53E+01
4,8796	2,7653	2,5013	0,000472	0,000754	1,54E+01

4,6443	2,4536	2,3461	0,00044	0,000797	1,51E+01
4,4562	2,5932	2,4685	0,000485	0,000793	1,50E+01

Perhitungan standarisasi data dan clustering memiliki persamaan dengan perhitungan di tahun 2001.

### 3.4.2.1 Perhitungan Lapisan ke-1

**Tabel 3.18** menunjukkan hasil dari perhitungan c (mean) dan **tabel 3.19** hasil perhitungan a (standar deviasi)

**Tabel 3.18** Nilai c (mean)

MEAN ( $\bar{x}$ )					
X1	X2	X3	X4	X5	X6
4,762E+00	2,609E+00	2,424E+00	4,560E-04	7,755E-04	1,860E-05
4,971E+00	2,619E+00	2,304E+00	4,390E-04	7,335E-04	3,155E-05
4,994E+00	2,708E+00	2,401E+00	4,530E-04	7,395E-04	3,065E-05
4,878E+00	2,712E+00	2,417E+00	4,640E-04	7,437E-04	2,950E-05
4,885E+00	2,793E+00	2,498E+00	4,763E-04	7,453E-04	2,953E-05

**Tabel 3.19** Nilai a (standar deviasi)

STANDAR DEVIASI ( $\sigma$ )					
X1	X2	X3	X4	X5	X6
1,664E-01	2,204E-01	1,097E-01	2,263E-05	3,041E-05	1,103E-05
1,096E-02	4,999E-02	7,920E-02	9,899E-06	3,889E-05	1,294E-05
3,064E-01	1,463E-01	1,579E-01	4,427E-05	4,535E-05	1,540E-05
2,604E-01	7,262E-02	4,511E-02	1,325E-05	3,179E-05	1,095E-05
7,884E-02	1,160E-02	3,083E-02	5,715E-06	1,205E-05	4,252E-06

Hasil dari perhitungan derajat keanggotaan ditunjukkan pada **tabel 3.20**.

### 3.4.2.2 Perhitungan Lapisan ke-2

**Tabel 3.21** menunjukkan hasil perhitungan lapisan ke-2

**Tabel 3.21** Output lapisan ke-2

NILAI FIRE STRENGTH ( $w_i$ )					
O2,1	O2,2	O2,3	O2,4	O2,5	O2,6
6,023E-04	2,266E-06	5,472E-05	1,402E-14	3,130E-12	3,205E-13
7,393E-03	3,044E-05	4,623E-06	8,059E-14	1,534E-12	2,257E-13
2,992E-04	3,277E-06	6,893E-07	8,435E-15	6,048E-13	1,338E-13
5,289E-04	1,144E-05	1,315E-06	3,084E-14	2,200E-12	2,514E-13
3,093E-05	3,168E-05	4,891E-05	5,938E-15	5,803E-12	7,977E-13
3,323E-01	3,644E-04	1,119E-03	3,395E-12	1,306E-11	1,605E-12
2,165E-02	3,725E-06	1,044E-05	2,433E-13	1,144E-12	1,454E-13
2,625E-05	1,828E-07	5,237E-05	6,112E-13	5,328E-13	6,723E-14
2,784E-04	8,022E-07	1,112E-05	4,462E-14	3,031E-12	4,009E-13
4,760E-08	8,112E-09	1,996E-07	1,825E-15	1,145E-15	1,200E-14
2,024E-10	1,128E-10	2,403E-11	1,204E-19	5,953E-18	4,274E-15
1,310E-11	1,421E-13	2,191E-13	5,304E-21	7,421E-20	2,848E-15

**3.4.2.3 Perhitungan Lapisan ke-3**

Tabel 3.22 menunjukkan hasil perhitungan lapisan ke-3

**Tabel 3.22** Output Lapisan ke-3

NORMALISASI FIRE STRENGTH					
O3,1	O3,2	O3,3	O3,4	O3,5	O3,6
9,136E-01	3,437E-03	8,299E-02	2,126E-11	4,748E-09	4,862E-10
9,953E-01	4,097E-03	6,224E-04	1,085E-11	2,065E-10	3,039E-11
9,869E-01	1,081E-02	2,274E-03	2,783E-11	1,995E-09	4,413E-10
9,765E-01	2,112E-02	2,427E-03	5,693E-11	4,062E-09	4,641E-10
2,773E-01	2,841E-01	4,386E-01	5,325E-11	5,203E-08	7,153E-09
9,956E-01	1,091E-03	3,352E-03	1,017E-11	3,911E-11	4,809E-12
9,993E-01	1,719E-04	4,822E-04	1,123E-11	5,282E-11	6,713E-12
3,331E-01	2,320E-03	6,646E-01	7,756E-09	6,761E-09	8,531E-10
9,589E-01	2,764E-03	3,829E-02	1,537E-10	1,044E-08	1,381E-09
1,864E-01	3,177E-02	7,818E-01	7,148E-09	4,483E-09	4,699E-08
5,965E-01	3,326E-01	7,083E-02	3,549E-10	1,755E-08	1,260E-05
9,730E-01	1,055E-02	1,628E-02	3,941E-10	5,513E-09	2,116E-04

#### 3.4.2.4 Perhitungan Lapisan ke-4

Hasil perhitungan matriks desain terdapat pada **tabel 3.23**, dan untuk perhitungan parameter konsekuen terdapat pada **tabel 3.24**. Sementara hasil perhitungan lapisan ke-4 terdapat pada **tabel 3.25**.

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



**Tabel 3.25** Output Lapisan ke-4

OUTPUT LAPISAN 4 (vi)					
O4,1	O4,2	O4,3	O4,4	O4,5	O4,6
1,37E+01	5,16E-02	1,25E+00	3,19E-10	7,13E-08	7,30E-09
1,35E+01	5,02E-02	1,21E+00	3,10E-10	6,93E-08	7,10E-09
1,37E+01	5,15E-02	1,24E+00	3,18E-10	7,11E-08	7,28E-09
1,38E+01	5,19E-02	1,25E+00	3,21E-10	7,17E-08	7,34E-09
1,33E+01	5,00E-02	1,21E+00	3,09E-10	6,90E-08	7,07E-09
2,04E+01	5,01E-02	1,21E+00	3,10E-10	6,91E-08	7,08E-09
1,33E+01	4,82E-02	1,16E+00	2,98E-10	6,66E-08	6,82E-09
1,29E+01	4,85E-02	1,17E+00	3,00E-10	6,70E-08	6,86E-09
1,35E+01	5,09E-02	1,23E+00	3,15E-10	7,03E-08	7,20E-09
1,37E+01	5,14E-02	1,24E+00	3,18E-10	7,10E-08	7,27E-09
1,25E+01	4,71E-02	1,14E+00	2,91E-10	6,51E-08	6,66E-09
1,32E+01	4,95E-02	1,19E+00	3,06E-10	6,84E-08	7,00E-09

### 3.4.2.5 Perhitungan Lapisan ke-5

Tabel 3.26 menunjukkan hasil perhitungan pada lapisan ke-

5

**Tabel 3.26** Output Lapisan ke-5

OUTPUT O5
1,50E+01
1,48E+01
1,50E+01
1,51E+01
1,45E+01
2,17E+01
1,45E+01
1,41E+01
1,48E+01
1,50E+01
1,37E+01
1,44E+01

Setelah dilakukan perhitungan sampai lapisan ke-5, selanjutnya akan update parameter c dan a dengan menggunakan *steepest descent*.

### 3.4.3 Propagasi Error – Steepest Descent

Jaringan adaptif yang didapatkan kemudian akan dilakukan pembelajaran untuk mendapatkan nilai c dan a yang baru dengan menggunakan metode *steepest descent*.

#### 3.4.3.1 Propagasi Error Tahun 2002

Untuk memperoleh nilai c dan a yang baru, terlebih dahulu dilakukan perhitungan *error* lapisan ke-5 sampai lapisan ke-1 dengan metode *steepest descent*. Berikut ini contoh perhitungan *error* tiap lapisan untuk data pertama :

- *Error* lapisan ke-5

$$\epsilon_5 = \frac{\partial E_p}{\partial x_{13}} = -2 (y_p - y'_p)$$

Hasil perhitungan *error* lapisan ke-5 dapat dilihat pada **tabel 3.27**.

**Tabel 3.27 Error Lapisan ke-5**

$\epsilon$
<b>LAP 5</b>
-1,60E+00
-2,00E+00
-4,00E-01
-8,00E-01
-2,60E+00
1,20E+01
-1,20E+00
-5,60E+00
-2,10E+01
-8,00E-01
-2,80E+00
-1,20E+00



- *Error* lapisan ke-4

Untuk Error pada lapisan 4 tidak berubah.

$$\begin{array}{l} \epsilon_{4A} = \epsilon_5 \\ \epsilon_{4B} = \epsilon_5 \\ \epsilon_{4C} = \epsilon_5 \end{array} \quad \begin{array}{l} \epsilon_{4D} = \epsilon_5 \\ \epsilon_{4E} = \epsilon_5 \\ \epsilon_{4F} = \epsilon_5 \end{array}$$

- *Error* lapisan ke-3

Hasil perhitungan error lapisan ke-3 dapat dilihat pada **tabel 3.28**.

$$\begin{array}{l} \epsilon_{3A} = \epsilon_{4A}f_1 \\ \epsilon_{3B} = \epsilon_{4B}f_2 \\ \epsilon_{3C} = \epsilon_{4C}f_3 \end{array} \quad \begin{array}{l} \epsilon_{3D} = \epsilon_{4D}f_1 \\ \epsilon_{3E} = \epsilon_{4E}f_2 \\ \epsilon_{3F} = \epsilon_{4F}f_3 \end{array}$$

**Tabel 3.28** *Error* Lapisan ke-3

ERROR LAP 3					
$\epsilon_A$	$\epsilon_B$	$\epsilon_C$	$\epsilon_D$	$\epsilon_E$	$\epsilon_F$
-5,71E+01	-2,15E-01	-5,19E+00	-1,33E-09	-2,97E-07	-3,04E-08
-7,19E+01	-2,70E-01	-6,53E+00	-1,67E-09	-3,73E-07	-3,82E-08
-1,44E+01	-5,42E-02	-1,31E+00	-3,35E-10	-7,49E-08	-7,67E-09
-2,81E+01	-1,06E-01	-2,55E+00	-6,54E-10	-1,46E-07	-1,50E-08
-9,52E+01	-3,58E-01	-8,65E+00	-2,22E-09	-4,95E-07	-5,07E-08
4,17E+02	1,57E+00	3,78E+01	9,69E-09	2,16E-06	2,22E-07
-4,21E+01	-1,58E-01	-3,82E+00	-9,79E-10	-2,19E-07	-2,24E-08
-1,97E+02	-7,41E-01	-1,79E+01	-4,59E-09	-1,02E-06	-1,05E-07
-7,67E+02	-2,89E+00	-6,97E+01	-1,79E-08	-3,99E-06	-4,08E-07
-2,84E+01	-1,07E-01	-2,58E+00	-6,61E-10	-1,48E-07	-1,51E-08
-9,33E+01	-3,51E-01	-8,48E+00	-2,17E-09	-4,85E-07	-4,97E-08
-3,97E+01	-1,50E-01	-3,61E+00	-9,25E-10	-2,07E-07	-2,12E-08

- *Error* lapisan ke-2

$$\epsilon_{2A} = \frac{w_1}{(w_1 + w_2 + w_3 + w_4 + w_5 + w_6)^2} (\epsilon_{3A} - \epsilon_{3B} - \epsilon_{3C} - \epsilon_{3D} - \epsilon_{3E} - \epsilon_{3F})$$

$$\epsilon_{2B} = \frac{w_2}{(w_1 + w_2 + w_3 + w_4 + w_5 + w_6)^2} (\epsilon_{3A} - \epsilon_{3B} - \epsilon_{3C} - \epsilon_{3D} - \epsilon_{3E} - \epsilon_{3F})$$

$$\varepsilon_{2C} = \frac{w_3}{(w_1 + w_2 + w_3 + w_4 + w_5 + w_6)^2} (\varepsilon_{3A} - \varepsilon_{3B} - \varepsilon_{3C} - \varepsilon_{3D} - \varepsilon_{3E} - \varepsilon_{3F})$$

$$\varepsilon_{2D} = \frac{w_4}{(w_1 + w_2 + w_3 + w_4 + w_5 + w_6)^2} (\varepsilon_{3A} - \varepsilon_{3B} - \varepsilon_{3C} - \varepsilon_{3D} - \varepsilon_{3E} - \varepsilon_{3F})$$

$$\varepsilon_{2E} = \frac{w_5}{(w_1 + w_2 + w_3 + w_4 + w_5 + w_6)^2} (\varepsilon_{3A} - \varepsilon_{3B} - \varepsilon_{3C} - \varepsilon_{3D} - \varepsilon_{3E} - \varepsilon_{3F})$$

$$\varepsilon_{2F} = \frac{w_6}{(w_1 + w_2 + w_3 + w_4 + w_5 + w_6)^2} (\varepsilon_{3A} - \varepsilon_{3B} - \varepsilon_{3C} - \varepsilon_{3D} - \varepsilon_{3E} - \varepsilon_{3F})$$

Hasil perhitungan error lapisan ke-2 dapat dilihat pada **tabel 3.29**.

**Tabel 3.29** Error Lapisan ke-2

ERROR LAP 2					
$\varepsilon A$	$\varepsilon B$	$\varepsilon C$	$\varepsilon D$	$\varepsilon E$	$\varepsilon F$
-4,32E+01	-6,11E-04	-3,56E-01	-2,34E-20	-1,17E-15	-1,22E-17
-6,44E+01	-1,09E-03	-2,52E-05	-7,66E-21	-2,77E-18	-6,01E-20
-1,27E+01	-1,52E-03	-6,75E-05	-1,01E-20	-5,19E-17	-2,54E-18
-2,43E+01	-1,14E-02	-1,50E-04	-8,25E-20	-4,20E-16	-5,48E-18
-6,63E+00	-6,96E+00	-1,66E+01	-2,44E-19	-2,33E-13	-4,41E-15
3,74E+02	4,49E-04	4,24E-03	3,90E-20	5,77E-19	8,72E-21
-3,80E+01	-1,13E-06	-8,85E-06	-4,80E-21	-1,06E-19	-1,72E-21
-1,98E+01	-9,60E-04	-7,88E+01	-1,07E-14	-8,16E-15	-1,30E-16
-6,39E+02	-5,31E-03	-1,02E+00	-1,64E-17	-7,57E-14	-1,32E-15
-8,94E-01	-2,60E-02	-1,57E+01	-1,31E-15	-5,17E-16	-5,68E-14
-3,01E+01	-9,35E+00	-4,24E-01	-1,06E-17	-2,60E-14	-1,34E-08
-3,41E+01	-4,01E-03	-9,54E-03	-5,59E-18	-1,09E-15	-1,61E-06

- Error lapisan ke-1

$$\varepsilon_{1A} = \varepsilon_{2A}\mu_{A1}$$

$$\varepsilon_{1D} = \varepsilon_{2D}\mu_{A1}$$

$$\varepsilon_{1B} = \varepsilon_{2B}\mu_{A2}$$

$$\varepsilon_{1E} = \varepsilon_{2E}\mu_{A2}$$

$$\varepsilon_{1C} = \varepsilon_{2C}\mu_{A3}$$

$$\varepsilon_{1F} = \varepsilon_{2F}\mu_{A3}$$

Hasil perhitungan error lapisan ke-1 dapat dilihat pada **tabel 3.30**.

**Tabel 3.30** Error Lapisan ke-1

RATA-RATA ERROR LAPISAN 1 ( $\epsilon_1$ )					
X1	X2	X3	X4	X5	X6
1,18E+01	-1,32E-02	-1,71E-02	-2,09E-26	-1,01E-24	-8,29E-19
1,60E+01	-6,81E-01	-3,86E+00	-5,51E-16	-2,00E-14	-1,26E-07
-8,04E+00	-1,29E+00	-6,63E+00	-8,29E-16	-1,79E-14	-2,82E-08
-1,13E+01	-1,44E-01	-3,78E+00	-3,09E-16	-1,72E-14	-1,64E-08
-2,53E+01	-1,09E-01	-3,77E+00	-2,71E-16	-1,58E-14	-2,75E-09

Selanjutnya dilakukan perubahan parameter  $c$  dan  $a$  dilakukan dengan menggunakan **persamaan 2.30** dengan menggunakan *learning rate* 0.0001. Berikut **tabel 3.31 dan 3.32** hasil perubahan parameter  $c$  dan  $a$ .

$$\Delta a_{ik} = \eta \epsilon_{aik} X_i, \text{ dan } \Delta c_{ik} = \eta \epsilon_{cik} X_i$$

**Tabel 3.31** Perubahan nilai parameter  $c$

ΔMEAN					
X1	X2	X3	X4	X5	X6
2,87E-08	-2,79E-10	-9,16E-13	-3,48E-32	-1,38E-29	-7,71E-19
3,43E-10	-1,14E-09	-3,89E-08	-1,29E-20	-9,74E-20	-2,71E-09
-3,16E-08	-1,09E-10	-3,07E-09	-8,38E-21	-3,03E-20	-3,77E-10
-1,05E-09	-8,99E-12	-1,08E-13	-2,82E-22	-3,88E-21	-6,19E-11
-1,05E-10	-3,85E-11	-4,45E-09	-1,60E-21	-3,56E-22	-5,22E-12

**Tabel 3.32** Perubahan nilai parameter  $a$

ΔSTDEV					
X1	X2	X3	X4	x5	X6
3,58E-08	-6,84E-10	1,38E-11	-1,97E-31	1,47E-29	-8,86E-19
-4,97E-11	-7,13E-10	-2,73E-08	-6,10E-21	-2,81E-19	1,05E-08
9,83E-08	1,75E-09	-3,08E-08	-5,30E-20	-1,83E-19	2,37E-09
-1,57E-08	4,97E-11	1,63E-11	9,35E-22	-3,74E-20	5,22E-10

-7,98E-10	4,75E-12	1,69E-09	5,61E-22	-2,53E-21	1,67E-11
-----------	----------	----------	----------	-----------	----------

Sehingga didapatkan nilai c dan a yang baru berdasarkan persamaan 2.31 dan 2.32. Berikut tabel 3.33 dan 3.34 hasil nilai c dan a yang baru.

$$a_{ik} = a_{ik}(\text{lama}) + \Delta a_{ik}$$

$$c_{ik} = c_{ik}(\text{lama}) + \Delta c_{ik}$$

**Tabel 3.33** Nilai baru parameter c

MEAN baru					
X1	X2	X3	X4	X5	X6
4,762	2,609	2,424	4,56E-04	7,76E-04	1,86E-05
4,971	2,619	2,304	4,39E-04	7,34E-04	3,15E-05
4,993	2,708	2,401	4,53E-04	7,40E-04	3,07E-05
4,878	2,712	2,417	4,64E-04	7,44E-04	2,95E-05
4,885	2,793	2,498	4,76E-04	7,45E-04	2,95E-05

**Tabel 3.34** Nilai baru parameter a

STANDAR DEVIASI baru					
X1	X2	X3	X4	X5	X6
1,66E-01	2,20E-01	1,10E-01	2,26E-05	3,04E-05	1,10E-05
1,10E-02	5,00E-02	7,92E-02	9,90E-06	3,89E-05	1,30E-05
3,06E-01	1,46E-01	1,58E-01	4,43E-05	4,53E-05	1,54E-05
2,60E-01	7,26E-02	4,51E-02	1,33E-05	3,18E-05	1,10E-05
7,88E-02	1,16E-02	3,08E-02	5,72E-06	1,21E-05	4,25E-06

### 3.4.3.2 Propagasi Error tahun 2001

Perubahan parameter c dan a dilakukan dengan menggunakan persamaan 2.30. Berikut tabel 3.35 dan 3.36 hasil perubahan parameter c dan a.

**Tabel 3.35** Perubahan nilai parameter c

$\Delta$ MEAN					
X1	X2	X3	X4	X5	X6
1,50E-07	2,65E-12	9,22E-25	1,50E-30	5,07E-31	4,74E-24
4,30E-08	3,54E-10	6,23E-17	3,44E-20	1,07E-20	7,20E-14
1,22E-05	6,94E-10	3,83E-17	1,44E-19	2,13E-20	4,23E-15
1,01E-06	2,12E-10	1,90E-18	1,78E-20	2,81E-20	3,08E-14
1,36E-07	2,64E-12	1,58E-18	2,42E-21	1,07E-20	6,94E-15

**Tabel 3.36** Perubahan nilai parameter a

$\Delta$ STDEV					
X1	X2	X3	X4	x5	X6
6,01E-07	2,61E-12	5,19E-24	9,42E-31	-2,73E-31	2,58E-24
-2,81E-09	8,27E-11	3,54E-17	6,43E-21	-2,86E-20	1,27E-13
-1,96E-05	8,15E-10	1,42E-16	1,63E-19	-4,70E-20	7,89E-15
-3,52E-06	1,28E-10	3,24E-18	8,41E-21	-3,61E-20	3,59E-14
-1,31E-07	7,73E-13	-8,96E-19	8,83E-22	-4,92E-21	3,15E-15

Sehingga didapatkan nilai c dan a yang baru berdasarkan persamaan 2.31 dan 2.32. Berikut tabel 3.37 dan 3.38 hasil nilai c dan a yang baru.

**Tabel 3.37** Nilai baru parameter c

MEAN baru					
X1	X2	X3	X4	X5	X6
4,762	2,609	2,424	4,56E-04	7,76E-04	1,86E-05
4,971	2,619	2,304	4,39E-04	7,34E-04	3,16E-05
4,994	2,708	2,401	4,53E-04	7,40E-04	3,07E-05
4,878	2,712	2,417	4,64E-04	7,44E-04	2,95E-05
4,885	2,793	2,498	4,76E-04	7,45E-04	2,95E-05

**Tabel 3.38** Nilai baru parameter a

STANDAR DEVIASI baru					
X1	X2	X3	X4	X5	X6
1,66E-01	2,20E-01	1,10E-01	2,26E-05	3,04E-05	1,10E-05
1,10E-02	5,00E-02	7,92E-02	9,90E-06	3,89E-05	1,29E-05
3,06E-01	1,46E-01	1,58E-01	4,43E-05	4,54E-05	1,54E-05
2,60E-01	7,26E-02	4,51E-02	1,33E-05	3,18E-05	1,10E-05
7,88E-02	1,16E-02	3,08E-02	5,72E-06	1,21E-05	4,25E-06

### 3.4.4 Perhitungan Error

#### 3.4.4.1 Error tahun 2002

Perhitungan nilai *Error* tahun 2002 menggunakan persamaan 2.35. Berikut tabel 3.39 perbandingan nilai *output* tahun 2002 dengan tahun 2001.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \bar{y}_1)^2}{N}}$$

**Tabel 3.9** Perbandingan *Output ANFIS* tahun 2001 dan 2002

Output	
2001	2002
1,58E+01	1,50E+01
1,58E+01	1,48E+01
1,52E+01	1,50E+01
1,55E+01	1,51E+01
1,58E+01	1,45E+01
1,57E+01	2,17E+01
1,51E+01	1,45E+01
1,69E+01	1,41E+01
2,53E+01	1,48E+01
1,54E+01	1,50E+01
1,51E+01	1,37E+01
1,50E+01	1,44E+01

Perhitungan *error* tahun 2002 menghasilkan nilai 4,04

### 3.4.4.2 Error tahun 2001

Tabel 3.40 merupakan perbandingan *output* tahun 2001 dengan *output aktual* tahun 2001.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \bar{y}_1)^2}{N}}$$

**Tabel 3.40** Perbandingan *Output aktual* dengan *Output ANFIS* tahun 2001

Output 2001	
Aktual	ANFIS
7,67E-09	1,58E+01
9,41E-09	1,58E+01
1,62E-08	1,52E+01
1,59E-08	1,55E+01
1,19E-08	1,58E+01
8,72E-09	1,57E+01
1,47E-08	1,51E+01
1,73E-08	1,69E+01
1,26E-08	2,53E+01
1,45E-08	1,54E+01
1,74E-08	1,51E+01
1,42E-08	1,50E+01

Perhitungan *error* tahun 2001 menghasilkan nilai 5,68, namun nilai *error* akhir menggunakan nilai *error* tahun 2002 dan 2001 dengan **persamaan 2.33**.

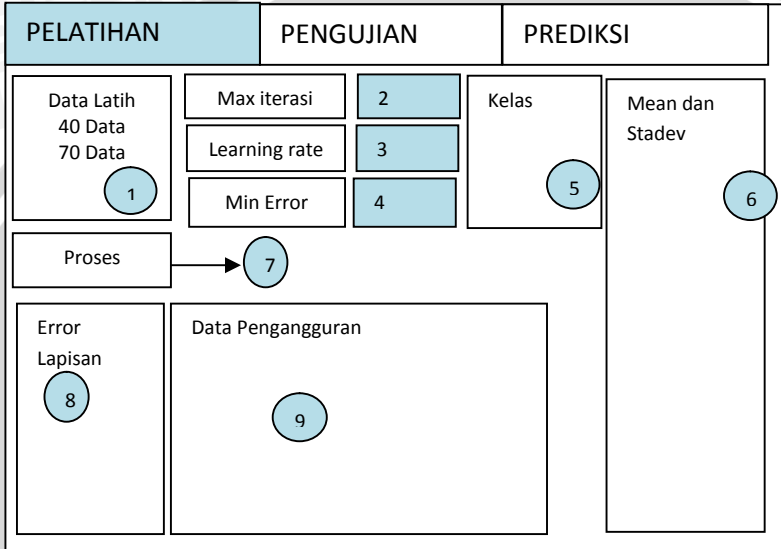
$$\begin{aligned} \text{Error} &= 5,68 + \frac{4,04}{5,68} \\ &= 6,39 \end{aligned}$$

Jika *error* jaringan masih melebihi *error* yang diinginkan, maka akan dilakukan iterasi lagi.

### 3.5 Perancangan Antarmuka

Berikut penjelasan rancangan antarmuka pada bagian pelatihan.

#### 3.5.1 Rancangan Antarmuka Pelatihan



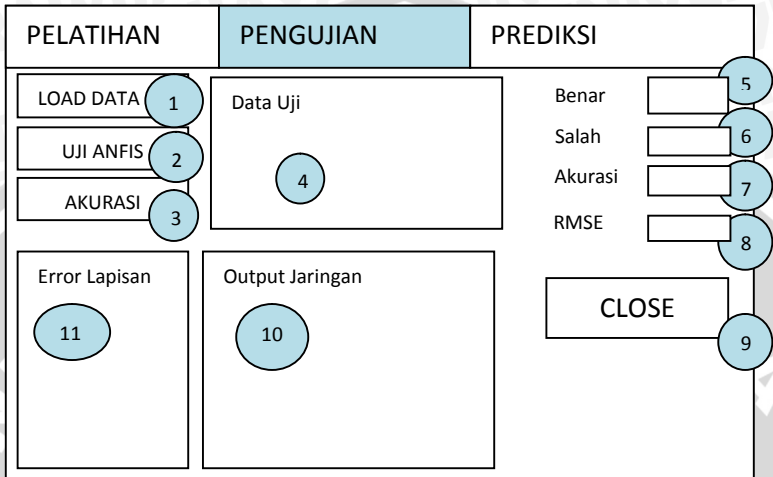
**Gambar 3.15** Rancangan antarmuka proses pelatihan *ANFIS*

Penjelasan gambar 3.15 :

1. Memilih jumlah data yang akan digunakan untuk pelatihan.
2. Mengisi maximal error untuk pelatihan.
3. Mengisi maximal iterasi yang diperlukan.
4. Mengisi *learning rate* untuk pelatihan.
5. Menampilkan *Centroid cluster* yang terbentuk.
6. Menampilkan perhitungan mean dan standar deviasi setelah diproses.
7. Memproses data yang akan digunakan dalam pelatihan sesuai jumlah data yang dilatih.
8. Menampilkan *Error Lapisan* pada parameter konsekuen yang akan dipakai untuk pengujian
9. Menampilkan data latih yang digunakan setelah dilakukan *load* data.



### 3.5.2 Rancangan Antarmuka Pengujian



**Gambar 3.16** Rancangan antarmuka pengujian *ANFIS*

Penjelasan gambar 3.16 :

1. Melakukan *load* data yang akan digunakan dalam pengujian.
2. Melihat hasil uji *ANFIS* dari data yang diujikan.
3. Melihat akurasi dari sistem *ANFIS*.
4. Menampilkan data pengujian.
5. Melihat jumlah data yang benar setelah diujikan.
6. Melihat jumlah data yang salah setelah diujikan.
7. Menampilkan hasil akurasi dari sistem.
8. Menampilkan nilai RMSE.
9. Menutup program.
10. Menampilkan *output* jaringan.
11. Menampilkan *error* yang telah dilatihkan.

### 3.5.3 Rancangan Antarmuka Prediksi

PELATIHAN	PENGUJIAN	PREDIKSI
Bulan + n <input type="text"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">1</span>		
<input type="text" value="PROSES"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">2</span>		
Hasil Prediksi		
<div style="border: 1px solid black; padding: 10px; min-height: 150px;"><span style="border: 1px solid black; border-radius: 50%; padding: 5px 10px; display: inline-block;">3</span></div>		

**Gambar 3.17** Rancangan antarmuka pengujian *ANFIS*

Rancangan antarmuka sistem ini berguna untuk melakukan prediksi tingkat pengangguran. Hasil prediksi didapatkan dari pelatihan dan pengujian sistem *ANFIS* ditahap sebelumnya, dan dari hasil tersebut akan digolongkan ke dalam golongan tingkat pengangguran yang menyatakan apakah sangat rendah, rendah, sedang, tinggi atau sangat tinggi. Penjelasan gambar 3.17:

1. Memproses data yang akan diramalkan sesuai jumlah waktu yang diinginkan.
2. Menampilkan interval waktu prediksi.
3. Menampilkan hasil prediksi.

### 3.6 Perancangan Uji Coba dan Analisis

Untuk melakukan tingkat keakurasian dari sistem *ANFIS* dalam prediksi pengangguran, maka akan dibuat beberapa rancangan untuk hasil dari uji coba yang menggunakan data pelatihan dengan jumlah data yang berbeda.

**Tabel 3.41** Contoh tabel nilai laju pembelajaran

Laju Pembelajaran	Jumlah Data	RMSE
0.00001		
0.00002		
...		
...		
...		
0.00009		

Penjelasan dari tabel 3.41 :

- Kolom laju *error* menunjukkan nilai *minimum error* yang digunakan ketika proses pelatihan.
- Kolom jumlah data menunjukkan jumlah data yang digunakan dalam proses pelatihan.
- Kolom RMSE digunakan untuk mencatat nilai RMSE yang dihasilkan pada proses pengujian jaringan *ANFIS*.

**Tabel 3.42** Contoh perbandingan nilai *output* jaringan dan *output aktual*

Prediksi Jaringan	Aktual

Penjelasan dari **tabel 3.42** :

- Kolom *output* jaringan menunjukkan nilai yang dihasilkan dari sistem *ANFIS*.
- Kolom *output* aktual menunjukkan nilai kondisi pengangguran dari data sebenarnya.

UNIVERSITAS BRAWIJAYA



## BAB IV IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras serta lingkungan perangkat lunak.

#### 4.1.1 Lingkungan perangkat keras

Perangkat keras yang digunakan dalam pembangunan sistem prediksi tingkat pengangguran ini adalah:

1. Prosesor IntelCore™2 Duo 2.20 GHz.
2. Memori 1 GB
3. *Harddisk* dengan kapasitas 160 GB
4. Monitor 14.0"
5. *Keyboard*
6. *Mouse*

#### 4.1.2 Lingkungan perangkat lunak

Perangkat lunak yang digunakan dalam pembangunan sistem prediksi tingkat pengangguran ini adalah :

1. Sistem Operasi *Microsoft Windows XP*
2. *Borland Delphi 7*
3. *Microsoft Access 2007*

### 4.2 Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi dari sistem prediksi tingkat pengangguran.

#### 4.2.1. Struktur Data

Struktur data yang digunakan untuk menyimpan data ketenagakerjaan, bobot dan output sistem adalah *array*. Struktur data pada *sourcecode 4.1* sebagai berikut:

```
1 Inv:array [1..200,1..200] of real;
2 C,D,E,F,O,P,Q:array of array of Real;
3 t,ck,n_Clu,Tw,Tf,bgf,Tproduktif1,Tangkatanr,
4 Tbekerjal,Ttpakl,Ttkks,Tsdmi:array [1..5] of Real;
5 jml1,nn:array [1..6,1..5] of real;
6 pstd,std,xx,xx1:array [1..5,1..6] of real;
7 cc1,cc2,cc3,cc4,cc5:array [1..100,1..6] of real;
8 AA,BB,CC,DD,EE,GG,wf,we,vv,ff,ea1,ea2,ea3,ea4,ea5,ea6,
9 ec1,ec2,ec3,ec4,ec5,ec6,da1,da2,da3,da4,da5,da6,
```

10	
11	dc1,dc2,dc3,dc4,dc5,dc6,a1,a2,a3,a4,a5,a6,bc1,bc2,bc3,
12	bc4,bc5,bc6,ff1,ff2,ff3,ff4,ff5,w1f1,w2f2,w3f3,
13	w4f4,w5f5:array [1..100,1..5] of real;
14	Sqlcmd,Tahun,O_Jar1: string;
15	O Jaringan:real;

**Sourcecode 4.1**Struktur data

Dari struktur data tersebut dapat dijelaskan bahwa penyimpanan setiap data merupakan *array* dari tipe data *real*. untuk menyimpan data parameter, *cluster* dan bobot digunakan data *array* 1 dimensi dan nilai output digunakan data *array* 2 dimensi.

**4.2.2 Proses Mencari Nilai Maksimum**

Untuk mencari nilai minimum dan maksimum dapat dilihat pada **Sourcecode 4.2** sebagai berikut:

```

1  begin
2      Q1.SQL.Text:='select * from 40'; //Latih40
3      Q1.Open;
4      Q2.Close;
5      Q2.SQL.Text:='SELECT MAX(produktif) as ld,
6  MIN(produktif) as ddl,MAX(angkatan) as um, MIN(angkatan)
7  as ru, '+
8          'MAX(bekerja) as hd, MIN(bekerja) as
9  dl, MAX(tpak) as ko, MIN(tpak) as ok, '+
10         'MAX(tkk) as si, MIN(tkk) as tkk,
11  MAX(sdm) as tr, '+
12         'MIN(sdm) as rt from 40';
13  end;
14  if RB70.Checked=true then
15  begin
16      Q1.SQL.Text:='select * from 70'; //Latih70
17      Q1.Open;
18      Q2.Close;
19      Q2.SQL.Text:='SELECT MAX(produktif) as ld,
20  MIN(produktif) as ddl,MAX(angkatan) as um, MIN(angkatan)
21  as ru, '+
22         'MAX(bekerja) as hd, MIN(bekerja) as
23  dl, MAX(tpak) as ko, MIN(tpak) as ok, '+
24         'MAX(tkk) as si, MIN(tkk) as tkk,
25  MAX(sdm) as tr, '+
26         'MIN(sdm) as rt from 70';
27  end;
28  Q2.Open;
29  Q2.First;
30  i:=1; n:=0;
31  max_weight:=0; min_weight:=10;
32  axReal:=0.0000000000000000000000000000000000000000000000000000001;
33  minReal:=-

```

```

34  0.00000000000000000000000000000000000000000000000001;
35  minB:=100; maxB:=0;
36  minErr:=100; maxErr:=0;
37  jumlah:=Q1.RecordCount;
38  SetLength(a,Q1.RecordCount);
39  SetLength(b,Q1.RecordCount);
40  SGData.RowCount:=Q1.RecordCount+1;
41  StringGrid4.RowCount:=Q1.RecordCount+1;
42  SGDataUji.RowCount:=Q1.RecordCount+1;

```

### Sourcecode 4.2 Mencari nilai minimum dan maksimum

Fungsi tersebut digunakan untuk mencari nilai minimum dan maksimum dari data parameter yang nantinya akan digunakan untuk proses standarisasi.

### 4.2.3 Proses Standarisasi

Untuk melakukan proses standarisasi dapat dilihat pada Sourcecode 4.3 sebagai berikut:

```

1  dt_produkatif:=Q1['produkatif'];
2  if (data2-data1) <> 0 then
3      std_produkatif:=(dt_produkatif-datal)/(data2-data1)
4  else
5      std_produkatif:=0;
6  dt_angkatan:=Q1['angkatan'];
7  if (data4-data3) <> 0 then
8      std_angkatan:=(dt_angkatan-data3)/(data4-data3)
9  else
10     std_angkatan:=0;
11  dt_bekerja:=Q1['bekerja'];
12  if (data6-data5) <> 0 then
13     std_bekerja:=(dt_bekerja-data5)/(data6-data5)
14  else
15     std_bekerja:=0;
16  dt_tpak:=Q1['tpak'];
17  if (data8-data7) <> 0 then
18     std_tpak:=(dt_tpak-data7)/(data8-data7)
19  else
20     std_tpak:=0;
21  dt_tkk:=Q1['tkk'];
22  if (data10-data9) <> 0 then
23     std_tkk:=(dt_tkk-data9)/(data10-data9)
24  else
25     std_tkk:=0;
26  dt_sdm:=Q1['sdm'];
27  if (data12-data11) <> 0 then
28     std_sdm:=(dt_sdm-data11)/(data12-data11)
29  else
30     std_sdm:=0;

```

### Sourcecode 4.3 Standarisasi

Fungsi tersebut digunakan untuk melakukan standarisasi terhadap data parameter tingkat pengangguran agar *range* data yang didapatkan sebagai hasil berkisar dari 0-1 .

#### 4.2.4 Proses Menghitung Bobot (w)

Fungsi untuk menghitung nilai bobot (w) dapat dilihat pada *Sourcecode 4.4* sebagai berikut:

```
1 weight:=(std_produkatif + std_angkatan + std_bekerja
2 +std_tpak + std_tkk + std_sdm)/6;
3 SGData.Cells[8,i]:=FloatToStr(weight);
4 if weight < min_weight then
5     min_weight:=weight;
6     if weight > max_weight then
7         max_weight:=weight;
8
9     Q1.Next;
10    Inc(i);
11    end;
```

*Sourcecode 4.4* Mengitung bobot (w)

Fungsi tersebut pada tahap selanjutnya digunakan untuk menghitung jarak Euclidean .

#### 4.2.5 Proses Pencarian Minimum dan Maksimum Bobot

Fungsi untuk mencari nilai minimum dan maksimum bobot dapat dilihat pada *Sourcecode 4.5* sebagai berikut:

```
1 if weight < min_weight then
2     min_weight:=weight;
3     if weight > max_weight then
4         max_weight:=weight;
5
6     Q1.Next;
7     Inc(i);
8     end;
```

*Sourcecode 4.5* Proses Inisialisasi Bobot

Fungsi diatas digunakan untuk menentukan bobot mana yang dipakai untuk pusat cluster awal.

#### 4.2.6 Proses Mencari Nilai Random Bobot

Fungsi untuk mencari nilai minimum dan maksimum bobot dapat dilihat pada *Sourcecode 4.6* sebagai berikut:



```

1 Randomize();
2 x:=Random; y:=Random; v:=Random; w:=Random; l:=Random;
3 random1:=(max_weight - min_weight)*x+min_weight;
4 random2:=(max_weight - min_weight)*y+min_weight;
5 random3:=(max_weight - min_weight)*v+min_weight;
6 random4:=(max_weight - min_weight)*w+min_weight;
7 random5:=(max_weight - min_weight)*l+min_weight;
8
9 EdCluster1.Text:=FloatToStr(random1);
10 EdCluAwal1.Text:=FloatToStr(random1);
11 EdCluster2.Text:=FloatToStr(random2);
12 EdCluAwal2.Text:=FloatToStr(random2);
13 EdCluster3.Text:=FloatToStr(random3);
14 EdCluAwal3.Text:=FloatToStr(random3);
15 EdCluster4.Text:=FloatToStr(random4);
16 EdCluAwal4.Text:=FloatToStr(random4);
17 EdCluster5.Text:=FloatToStr(random5);
18 EdCluAwal5.Text:=FloatToStr(random5);
19
20 sama:=false;
21 Q1.Open;
22 Q1.First;
23 while not sama do
24 begin
25 inc(n); c1:=0; c2:=0; c3:=0; c4:=0; c5:=0;
26 n_Clu1:=0; n_Clu2:=0; n_Clu3:=0; n_Clu4:=0;
27 n_Clu5:=0;
28 for i:=1 to SGData.RowCount-1 do
29 begin
30 weight:=StrToFloat(SGData.Cells[8,i]);
31
32 ca1:=StrToFloat(EdCluster1.Text);ca2:=StrToFloat(EdCluster
33 2.Text);
34
35 ca3:=StrToFloat(EdCluster3.Text);ca4:=StrToFloat(EdCluster
36 4.Text);
37 ca5:=StrToFloat(EdCluster5.Text);
38 jarak1:=abs(StrToFloat(SGData.Cells[8,i])-ca1);
39 jarak2:=abs(StrToFloat(SGData.Cells[8,i])-ca2);
40 jarak3:=abs(StrToFloat(SGData.Cells[8,i])-ca3);
41 jarak4:=abs(StrToFloat(SGData.Cells[8,i])-ca4);
42 jarak5:=abs(StrToFloat(SGData.Cells[8,i])-ca5);
43 jarak8:=abs(StrToFloat(SGData.Cells[8,i])-
44 StrToFloat(EdCluAwal1.Text));
45 jarak9:=abs(StrToFloat(SGData.Cells[8,i])-
46 StrToFloat(EdCluAwal2.Text));
47
48 SGData.Cells[10,i]:=FloatToStr(jarak1);SGData.Cells[11,i]:
49 =FloatToStr(jarak2);
50
51 SGData.Cells[12,i]:=FloatToStr(jarak3);SGData.Cells[13,i]:
52 =FloatToStr(jarak4);
53 SGData.Cells[14,i]:=FloatToStr(jarak5);
54 end;
55 sama:=true;

```

```

56
57     for i:=1 to SGData.RowCount-1 do
58     begin
59         t1:=StrToFloat(SGData.Cells[10,i]);
60         t2:=StrToFloat(SGData.Cells[11,i]);
61         t3:=StrToFloat(SGData.Cells[12,i]);
62         t4:=StrToFloat(SGData.Cells[13,i]);
63         t5:=StrToFloat(SGData.Cells[14,i]);
64
65         if(t1 < t2) and (t1 < t3) and (t1 < t4) and (t1 <
66 t5) then
67         begin
68             cls:='1';
69             SGData.Cells[15,i]:=cls;
70         end
71         else if(t2 < t1) and (t2 < t3) and (t2 < t4) and (t2
72 < t5) then
73         begin
74             cls:='2';
75             SGData.Cells[15,i]:=cls;
76         end
77         else if(t3 < t1) and (t3 < t2) and (t3 < t4) and (t3
78 < t5) then
79         begin
80             cls:='3';
81             SGData.Cells[15,i]:=cls;
82         end
83         else if (t4 < t1) and (t4 < t2) and (t4 < t3) and
84 (t4 < t5) then
85         begin
86             cls:='4';
87             SGData.Cells[15,i]:=cls;
88         end
89         else if (t5 < t1) and (t5 < t2) and (t5 < t3) and
90 (t5 < t4) then
91         begin
92             cls:='5';
93             SGData.Cells[15,i]:=cls;
94         end;
95     end;

```

**Sourcecode 4.6** Fungsi mencari nilai random bobot

Tahap pertama yang dilakukan adalah mengambil nilai bobot kemudian melakukan inisialisasi bobot secara acak. Setelah itu nilai bobot random ini di pakai sebagai pusat cluster awal. Pencarian nilai ini dilakukan untuk menentukan ukuran kemiripan atau tidak kemiripan antar data.

#### 4.2.7 Proses Penentuan Cluster

Fungsi untuk menghitung nilai aktivasi dapat dilihat pada **Sourcecode 4.7** sebagai berikut:

```

1   for i:=1 to 6 do
2     begin
3       for j:=1 to 5 do
4         begin
5           jml1[i,j]:=0;
6           nn[i,j]:=0;
7           std[i,j]:=0;
8           pstd[i,j]:=0;
9           xx[i,j]:=0;
10        end;
11      end;
12
13      for i:=0 to Q1.RecordCount-1 do
14        begin
15          if (SGData.Cells[15,i+1]='1') then
16            begin
17              Inc(c1);
18              n_Clu1:=n_Clu1+StrToFloat (SGData.Cells[8,i+1]);
19              EdJmlc1.Text:=IntToStr(c1);
20              for j:=1 to 6 do
21                begin
22
23                  jml1[1,j]:=jml1[1,j]+StrToFloat (SGData.Cells[j,i+1]);
24                  nn[1,j]:=nn[1,j]+1;
25                end;
26                if n=1 then
27                  begin
28                    a[i]:='0'; b[i]:='1';
29                  end
30                else
31                  begin
32                    a[i]:=b[i]; b[i]:='1';
33                  end
34                end
35                else if (SGData.Cells[15,i+1]='2') then
36                  begin
37                    Inc(c2);
38                    n_Clu2:=n_Clu2+StrToFloat (SGData.Cells[8,i+1]);
39                    EdJmlc2.Text:=IntToStr(c2);
40                    for j:=1 to 6 do
41                      begin
42
43                        jml1[2,j]:=jml1[2,j]+StrToFloat (SGData.Cells[j,i+1]);
44                        nn[2,j]:=nn[2,j]+1;
45                      end;
46                      if n=1 then
47                        begin
48                          a[i]:='0'; b[i]:='2';
49                        end
50                      else
51                        begin
52                          a[i]:=b[i]; b[i]:='2';
53                        end;
54                      end
55                    else if (SGData.Cells[15,i+1]='3') then

```

```

56     begin
57         Inc(c3);
58         n_Clu3:=n_Clu3+StrToFloat(SGData.Cells[8,i+1]);
59         EdJmlc3.Text:=IntToStr(c3);
60         for j:=1 to 6 do
61             begin
62
63                 jml1[3,j]:=jml1[3,j]+StrToFloat(SGData.Cells[j,i+1]);
64                 nn[3,j]:=nn[3,j]+1;
65             end;
66         if n=1 then
67             begin
68                 a[i]:='0'; b[i]:='3';
69             end
70         else
71             begin
72                 a[i]:=b[i]; b[i]:='3';
73             end;
74         end
75     else if (SGData.Cells[15,i+1]='4') then
76     begin
77         Inc(c4);
78         n_Clu4:=n_Clu4+StrToFloat(SGData.Cells[8,i+1]);
79         EdJmlc4.Text:=IntToStr(c4);
80         for j:=1 to 6 do
81             begin
82
83                 jml1[4,j]:=jml1[4,j]+StrToFloat(SGData.Cells[j,i+1]);
84                 nn[4,j]:=nn[4,j]+1;
85             end;
86         if n=1 then
87             begin
88                 a[i]:='0'; b[i]:='4';
89             end
90         else
91             begin
92                 a[i]:=b[i]; b[i]:='4';
93             end;
94         end
95     else if (SGData.Cells[15,i+1]='5') then
96     begin
97         Inc(c5);
98         n_Clu5:=n_Clu5+StrToFloat(SGData.Cells[8,i+1]);
99         EdJmlc5.Text:=IntToStr(c5);
100        for j:=1 to 6 do
101            begin
102
103                jml1[5,j]:=jml1[5,j]+StrToFloat(SGData.Cells[j,i+1]);
104                nn[5,j]:=nn[5,j]+1;
105            end;
106        if n=1 then begin
107            a[i]:='0'; b[i]:='5';
108        end
109        else
110        begin

```

```

111     a[i]:=b[i];  b[i]:='5';
112     end;
113     end;
114     end;

```

**Sourcecode 4.7** Fungsi penentuan *cluster*

Fungsi tersebut digunakan untuk menentukan data tersebut masuk ke cluster yang mana.

#### 4.2.8 Proses Menghitung Rata-rata (*Mean*)

Fungsi untuk menghitung rata-rata (*mean*) dapat dilihat pada **Sourcecode 4.8** sebagai berikut:

```

1  for i:=1 to 6 do
2  begin
3      for j:=1 to 5 do
4          begin
5              if nn[j,i]=0 then xx[j,i]:=0 else
6                  begin
7                      xx[j,i]:=jml1[j,i]/nn[j,i];
8                      SGRata.Cells[i,j]:=FloatToStr(xx[j,i]);
9                  end;
10             end;
11         end;

```

**Sourcecode 4.8** Rata-rata (*Mean*)

#### 4.2.9 Proses Menghitung *Standar Deviasi*

Fungsi untuk menghitung standar deviasi dapat dilihat pada **Sourcecode 4.9** sebagai berikut:

```

for i:=0 to Q1.RecordCount-1 do
1  begin
2      if (SGData.Cells[15,i+1]='1') then
3          begin
4              for j:=1 to 6 do
5                  begin
6                      if nn[1,j] < 2 then
7                          std[1,j]:=0
8                      else
9                          begin
10                             pstd[1,j]:=pstd[1,j]+SQR((StrToFloat(SGData.Cells[j,i+1])-
11                                 xx[1,j]));
12                             if (nn[1,j]-1) <> 0 then
13                                 std[1,j]:=(sqrt(pstd[1,j]/(nn[1,j]-1)))
14                             else
15                                 std[1,j]:=0;
16                             SGRata.Cells[j,7]:=FloatToStr(std[1,j]);
17                         end;

```

```

18     end;
19     end
20     else if (SGData.Cells[15,i+1]='2') then
21     begin
22     for j:=1 to 6 do
23     begin
24     if nn[2,j] < 2 then
25     std[2,j]:=0
26     else
27     begin
28
29     pstd[2,j]:=pstd[2,j]+SQR((StrToFloat(SGData.Cells[j,i+1])-
30     xx[2,j]));
31     if(nn[2,j]-1) <> 0 then
32     std[2,j]:= (sqrt (pstd[2,j]/(nn[2,j]-1)))
33     else
34     std[2,j]:=0;
35     SGRata.Cells[j,8]:=FloatToStr(std[2,j]);
36     end;
37     end;
38     end
39     else if (SGData.Cells[15,i+1]='3') then
40     begin
41     for j:=1 to 6 do
42     begin
43     if nn[3,j] < 2 then
44     std[3,j]:=0
45     else
46     begin
47
48     pstd[3,j]:=pstd[3,j]+SQR((StrToFloat(SGData.Cells[j,i+1])-
49     xx[3,j]));
50     if(nn[3,j]-1)<> 0 then
51     std[3,j]:= (sqrt (pstd[3,j]/(nn[3,j]-1)))
52     else
53     std[3,j]:=0;
54     SGRata.Cells[j,9]:=FloatToStr(std[3,j]);
55     end;
56     end;
57     end
58     else if (SGData.Cells[15,i+1]='4') then
59     begin
60     for j:=1 to 6 do
61     begin
62     if nn[4,j] < 2 then
63     std[4,j]:=0
64     else
65     begin
66
67     pstd[4,j]:=pstd[4,j]+SQR((StrToFloat(SGData.Cells[j,i+1])-
68     xx[4,j]));
69     if(nn[4,j]-1) <>0 then
70     std[4,j]:= (sqrt (pstd[4,j]/(nn[4,j]-1)))
71     else
72     std[4,j]:=0;

```

```

73     SGRata.Cells[j,10]:=FloatToStr(std[4,j]);
74     end;
75     end;
76     end
77     else if (SGData.Cells[15,i+1]='5') then
78     begin
79         for j:=1 to 6 do
80         begin
81             if nn[5,j] < 2 then
82                 std[5,j]:=0
83             else
84                 begin
85
86 pstd[5,j]:=pstd[5,j]+SQR((StrToFloat(SGData.Cells[j,i+1])-
87 xx[5,j]));
88                 if(xx[5,j]-1) <>0 then
89                     std[5,j]:=(sqrt(pstd[5,j]/(abs(xx[5,j]-1))))
90                 else
91                     std[5,j]:=0;
92                 SGRata.Cells[j,11]:=FloatToStr(std[5,j]);
93                 end;
94             end;
95         end;
96     end;
97
98

```

**Sourcecode 4.9** Proses menghitung standar deviasi

### 4.3 Perhitungan Lapisan ANFIS

#### 4.3.1 Nilai Derajat Keanggotaan

Fungsi untuk menghitung derajat keanggotaan dapat dilihat pada **Sourcecode 4.10** sebagai berikut:

```

.   if std[j,1]=0 then AA[j,i]:=1 else
1   AA[i,j]:=abs(1/(1+SQR((((StrToFloat(SGData.Cells[1,i])-
2   xx[j,1])/std[j,1])))));
3   if std[j,2]=0 then BB[j,i]:=1 else
4   BB[i,j]:=abs(1/(1+SQR((((StrToFloat(SGData.Cells[2,i])-
5   xx[j,2])/std[j,2])))));
6   if std[j,3]=0 then CC[j,i]:=1 else
7   CC[i,j]:=abs(1/(1+SQR((((StrToFloat(SGData.Cells[3,i])-
8   xx[j,3])/std[j,3])))));
9   if std[j,4]=0 then DD[j,i]:=1 else
10  DD[i,j]:=abs(1/(1+SQR((((StrToFloat(SGData.Cells[4,i])-
11  xx[j,4])/std[j,4])))));
12  if std[j,5]=0 then EE[j,i]:=1 else
13  EE[i,j]:=abs(1/(1+SQR((((StrToFloat(SGData.Cells[5,i])-
14  xx[j,5])/std[j,5])))));
15
16
17
18

```

```

19 xx[j,5])/std[j,5]))));
20     if std[j,6]=0 then GG[j,i]:=1 else
21
22 GG[i,j]:=abs(1/(1+SQR(((StrToFloat(SGData.Cells[6,i])-
23 xx[j,6])/std[j,6]))));
24

```

**Sourcecode 4.10** Nilai Derajat Keanggotaan

### 4.3.2 Nilai Firing Strength

Fungsi untuk menghitung firing strength dapat dilihat pada **Sourcecode 4.11** sebagai berikut:

```

1 we[i,j]:=AA[i,j]*BB[i,j]*CC[i,j]*DD[i,j]*EE[i,j]*GG[i,j];

```

**Sourcecode 4.11** Nilai Firing Strength

### 4.3.3 Normalisasi Firing Strength

Fungsi untuk menghitung normalisasi firing strength dapat dilihat pada **Sourcecode 4.12** sebagai berikut:

```

1 vv[i,j]:=we[i,j]/(we[i,1]+we[i,2]+we[i,3]+we[i,4]+we[i,5]);

```

**Sourcecode 4.12** Normalisasi Firing Strength

### 4.3.4 Matriks Desain

Fungsi untuk menghitung *matriks desain* dapat dilihat pada **Sourcecode 4.13** sebagai berikut:

```

. c10[i]:=vv[i,1];
1
2 cc1[i,k]:=(StrToFloat(SGData.Cells[k,i]))*vv[i,1];
3   c20[i]:=vv[i,2];
4
5 cc2[i,k]:=(StrToFloat(SGData.Cells[k,i]))*vv[i,2];
6   c30[i]:=vv[i,3];
7
8 cc3[i,k]:=(StrToFloat(SGData.Cells[k,i]))*vv[i,3];
9   c40[i]:=vv[i,4];
10
11 cc4[i,k]:=(StrToFloat(SGData.Cells[k,i]))*vv[i,4];
12   c50[i]:=vv[i,5];
13
14 cc5[i,k]:=(StrToFloat(SGData.Cells[k,i]))*vv[i,5];
15 for i:=1 to StringGrid4.RowCount-1 do
16   for j:=1 to StringGrid4.ColCount-1 do
17     begin
18       C[i-1,j-1]:=StrToFloat(StringGrid4.Cells[j,i]);
19     end;
20   for i:=1 to StringGrid4.ColCount-1 do

```



```

21
22     for j:=1 to StringGrid4.RowCount-1 do
23     begin
24         D[i-1,j-1]:=C[j-1,i-1];
25     end;
26
27     for i:=1 to StringGrid4.ColCount-1 do
28     for j:=1 to StringGrid4.ColCount-1 do
29     begin
30         E[i-1,j-1]:=0;
31         for k:=1 to StringGrid4.RowCount-1 do
32             E[i-1,j-1]:=E[i-1,j-1]+D[j-1,k-1]*C[k-1,i-1];
33         end;
34
35     for i:=1 to StringGrid4.ColCount-1 do
36     for j:=1 to StringGrid4.ColCount-1 do
37     begin
38         F[i-1,j-1]:=E[j-1,i-1];
39     end;

```

**Sourcecode 4.13** Nilai Matriks Desain

### 4.3.5 Invers Matriks

Fungsi untuk menghitung *invers matriks desain* dapat dilihat pada **Sourcecode 4.14** sebagai berikut:

```

.   for i:=1 to StringGrid4.ColCount-1 do
1   for j:=1 to StringGrid4.ColCount-1 do
2   begin
3   if i=j then
4   Inv[i,j]:=1
5   else
6   Inv[i,j]:=0;
7   end;
8
9   for i:=1 to StringGrid4.ColCount-2 do
10  for k:=i+1 to StringGrid4.ColCount-1 do
11  begin
12  z:=F[k-1,i-1]/F[i-1,i-1];
13  for j:=1 to StringGrid4.ColCount-1 do
14  begin
15  F[k-1,j-1]:=F[k-1,j-1]-z*F[i-1,j-1];
16  Inv[k,j]:=Inv[k,j]-z*Inv[i,j];
17  end;
18  F[k-1,i-1]:=0;
19  end;
20
21  for i:=StringGrid4.ColCount-1 downto 2 do
22  for k:=i-1 downto 1 do
23  begin
24  z:=F[k-1,i-1]/F[i-1,i-1];
25  for j:=1 to StringGrid4.ColCount-1 do
26  begin

```

```

27     F[k-1,j-1]:=F[k-1,j-1]-z*F[i-1,j-1];
28     Inv[k,j]:=Inv[k,j]-z*Inv[i,j];
29     end;
30     F[k-1,i-1]:=0;
31     end;
32
33     for i:=1 to StringGrid4.ColCount-1 do
34         for j:=1 to StringGrid4.ColCount-1 do
35             Inv[i,j]:=Inv[i,j]/F[i-1,i-1];
36

```

**Sourcecode 4.14 Invers Matriks Desain**

#### 4.4 Pencarian Error

##### 4.4.1 Error Lapisan 5

Fungsi untuk menghitung *error* lapisan 5 dapat dilihat pada **Sourcecode 4.15** sebagai berikut:

```

1 Er50[i]:=-2*(output_jar[i]-StrToFloat(SGData.Cells[7,i]));
2 TEr50:=TEr50+Er50[i]; bgEr50:=TEr50/jumlah;

```

**Sourcecode 4.15 Error Lapisan 5**

##### 4.4.2 Error Lapisan 4

Fungsi untuk menghitung *error* lapisan 4 dapat dilihat pada **Sourcecode 4.16** sebagai berikut:

```

1 Er49[i]:=bgEr50; Er48[i]:=bgEr50; Er47[i]:=bgEr50;
2 Er46[i]:=bgEr50; Er45[i]:=bgEr50;

```

**Sourcecode 4.16 Error Lapisan 4**

##### 4.4.3 Error Lapisan 3

Fungsi untuk menghitung *error* lapisan 3 dapat dilihat pada **Sourcecode 4.17** sebagai berikut:

```

1 Er44[i]:=bgEr50*Tf5; Er43[i]:=bgEr50*Tf4;
2 Er42[i]:=bgEr50*Tf3;
3     Er41[i]:=bgEr50*Tf2; Er40[i]:=bgEr50*Tf1;
4
5     TEr44:=TEr44+Er44[i]; bgEr44:=TEr44/jumlah;
6     TEr43:=TEr43+Er43[i]; bgEr43:=TEr43/jumlah;
7     TEr42:=TEr42+Er42[i]; bgEr42:=TEr42/jumlah;
8     TEr41:=TEr41+Er41[i]; bgEr41:=TEr41/jumlah;
9     TEr40:=TEr40+Er40[i]; bgEr40:=TEr40/jumlah;

```

**Sourcecode 4.17 Error Lapisan 3**

#### 4.4.4 Error Lapisan 2

Fungsi untuk menghitung *error* lapisan 2 dapat dilihat pada *Sourcecode 4.18* sebagai berikut:

```
1 Er39[i]:=(Tw[1]/SQR(totwifi[i]))*(Er44[i]-
2 (Er40[i]+Er41[i]+Er42[i]+Er43[i]));
3 Er38[i]:=(Tw[2]/SQR(totwifi[i]))*(Er43[i]-
4 (Er40[i]+Er41[i]+Er42[i]+Er44[i]));
5 Er37[i]:=(Tw[3]/SQR(totwifi[i]))*(Er42[i]-
6 (Er40[i]+Er41[i]+Er43[i]+Er44[i]));
7 Er36[i]:=(Tw[4]/SQR(totwifi[i]))*(Er41[i]-
8 (Er40[i]+Er42[i]+Er43[i]+Er44[i]));
9 Er35[i]:=(Tw[5]/SQR(totwifi[i]))*(Er40[i]-
10 (Er41[i]+Er42[i]+Er43[i]+Er44[i]));
11 TEr39:=TEr39+Er39[i]; bgEr39:=TEr39/jumlah;
12 TEr38:=TEr38+Er38[i]; bgEr38:=TEr38/jumlah;
13 TEr37:=TEr37+Er37[i]; bgEr37:=TEr37/jumlah;
14 TEr36:=TEr36+Er36[i]; bgEr36:=TEr36/jumlah;
15 TEr35:=TEr35+Er35[i]; bgEr35:=TEr35/jumlah;
```

*Sourcecode 4.18 Error Lapisan 2*

#### 4.4.5 Error Lapisan 1

Fungsi untuk menghitung *error* lapisan 1 dapat dilihat pada *Sourcecode 4.19* sebagai berikut:

```
1 Er34[i]:=Er39[i]*Tproduktifl[1];
2 Er22[i]:=Er37[i]*Tbekerjal[3];
3 Er33[i]:=Er38[i]*Tproduktifl[2];
4 Er21[i]:=Er36[i]*Tbekerjal[4];
5 Er32[i]:=Er37[i]*Tproduktifl[3];
6 Er20[i]:=Er35[i]*Tbekerjal[5];
7 Er31[i]:=Er36[i]*Tproduktifl[4];
8 Er19[i]:=Er39[i]*Ttpakl[1];
9 Er30[i]:=Er35[i]*Tproduktifl[5];
10 Er18[i]:=Er38[i]*Ttpakl[2];
11 Er29[i]:=Er39[i]*Tangkatanr[1];
12 Er17[i]:=Er37[i]*Ttpakl[3];
13 Er28[i]:=Er38[i]*Tangkatanr[2];
14 Er16[i]:=Er36[i]*Ttpakl[4];
15 Er27[i]:=Er37[i]*Tangkatanr[3];
16 Er15[i]:=Er35[i]*Ttpakl[5];
17 Er26[i]:=Er36[i]*Tangkatanr[4];
18 Er14[i]:=Er39[i]*Ttkks[1];
19 Er25[i]:=Er35[i]*Tangkatanr[5];
20 Er13[i]:=Er38[i]*Ttkks[2];
21 Er24[i]:=Er39[i]*Tbekerjal[1];
22 Er12[i]:=Er37[i]*Ttkks[3];
23 Er23[i]:=Er38[i]*Tbekerjal[2];
24 Er11[i]:=Er36[i]*Ttkks[4];
25
26 if (Er35[i]*Ttkks[5])> maxReal then
```

```

27 Er10[i]:=maxReal
28     else if (Er35[i]*Ttkks[5]) < minReal then
29 Er10[i]:=minReal
30     else Er10[i]:=Er35[i]*Ttkks[5];
31
32     if (Er39[i]*Tsdmi[1])> maxReal then
33 Er9[i]:=maxReal
34     else if (Er39[i]*Tsdmi[1]) < minReal then
35 Er9[i]:=minReal
36     else Er9[i]:=Er39[i]*Tsdmi[1];
37
38     Er8[i]:=Er38[i]*Tsdmi[2];
39
40     if (Er37[i]*Tsdmi[3])> maxReal then
41 Er7[i]:=maxReal
42     else if (Er37[i]*Tsdmi[3]) < minReal then
43 Er7[i]:=minReal
44     else Er7[i]:=Er37[i]*Tsdmi[3];
45
46     if (Er36[i]*Tsdmi[4])> maxReal then
47 Er6[i]:=maxReal
48     else if (Er36[i]*Tsdmi[4]) < minReal then
49 Er6[i]:=minReal
50     else Er6[i]:=Er36[i]*Tsdmi[4];
51
52     if (Er35[i]*Tsdmi[5])> maxReal then
53 Er5[i]:=maxReal
54     else if (Er35[i]*Tsdmi[5]) < minReal then
55 Er5[i]:=minReal
56     else Er5[i]:=Er35[i]*Tsdmi[5];
57
58 TEr34:=TEr34+Er34[i];   TEr19:=TEr19+Er19[i];
59 TEr33:=TEr33+Er33[i];   TEr18:=TEr18+Er18[i];
60 TEr32:=TEr32+Er32[i];   TEr17:=TEr17+Er17[i];
61 TEr31:=TEr31+Er31[i];   TEr16:=TEr16+Er16[i];
62 TEr30:=TEr30+Er30[i];   TEr15:=TEr15+Er15[i];
63 TEr29:=TEr29+Er29[i];   TEr14:=TEr14+Er14[i];
64 TEr28:=TEr28+Er28[i];   TEr13:=TEr13+Er13[i];
65 TEr27:=TEr27+Er27[i];   TEr12:=TEr12+Er12[i];
66 TEr26:=TEr26+Er26[i];   TEr11:=TEr11+Er11[i];
67 TEr25:=TEr25+Er25[i];   TEr10:=TEr10+Er10[i];
68 TEr24:=TEr24+Er24[i];   TEr9:=TEr9+Er9[i];
69 TEr23:=TEr23+Er23[i];   TEr8:=TEr8+Er8[i];
70 TEr22:=TEr22+Er22[i];   TEr7:=TEr7+Er7[i];
71 TEr21:=TEr21+Er21[i];   TEr6:=TEr6+Er6[i];
72 TEr20:=TEr20+Er20[i];   TEr5:=TEr5+Er5[i];
73
74     bgEr34:=TEr34/jumlah;   bgEr33:=TEr33/jumlah;
75 bgEr32:=TEr32/jumlah;
76     bgEr31:=TEr31/jumlah;   bgEr30:=TEr30/jumlah;
77 bgEr29:=TEr29/jumlah;
78     bgEr28:=TEr28/jumlah;   bgEr27:=TEr27/jumlah;
79 bgEr26:=TEr26/jumlah;
80     bgEr25:=TEr25/jumlah;   bgEr24:=TEr24/jumlah;
81 bgEr23:=TEr23/jumlah;

```

82	bgEr22:=TER22/jumlah;	bgEr21:=TER21/jumlah;
83	bgEr20:=TER20/jumlah;	
84	bgEr19:=TER19/jumlah;	bgEr18:=TER18/jumlah;
85	bgEr17:=TER17/jumlah;	
86	bgEr16:=TER16/jumlah;	bgEr15:=TER15/jumlah;
87	bgEr14:=TER14/jumlah;	
88	bgEr13:=TER13/jumlah;	bgEr12:=TER12/jumlah;
89	bgEr11:=TER11/jumlah;	
90	bgEr10:=TER10/jumlah;	bgEr9:=TER9/jumlah;
91	bgEr8:=TER8/jumlah;	
92	bgEr7:=TER7/jumlah;	bgEr6:=TER6/jumlah;
93	bgEr5:=TER5/jumlah;	

### Sourcecode 4.19 Error Lapisan 1

#### 4.5 Error Parameter $a_{ij}$ dan $c_{ij}$

Fungsi untuk menghitung *error* parameter dapat dilihat pada **Sourcecode 4.20** sebagai berikut:

1	if (std[j,1])=0 then
2	begin
3	ea1[i,j]:=0; ec1[i,j]:=0;
4	end
5	else
6	begin
7	ea1[i,j]:=((2*(SQR(Tproduktif-
8	xx[j,1])))/((Power(std[j,1],3))*((SQR((1+(SQR((Tproduktif-
9	xx[j,1])/std[j,1]))))));
10	ec1[i,j]:=((2*(Tproduktif-
11	xx[j,1]))/(SQR(std[j,1))*((SQR((1+(SQR((Tproduktif-
12	xx[j,1])/std[j,1]))))));
13	end;
14	
15	if (std[j,2])=0 then
16	begin
17	ea2[i,j]:=0; ec2[i,j]:=0;
18	end
19	else
20	begin
21	ea2[i,j]:=((2*(SQR(Tangkatan-
22	xx[j,2])))/((Power(std[j,2],3))*((SQR((1+(SQR((Tangkatan-
23	xx[j,2])/std[j,2]))))));
24	ec2[i,j]:=((2*(Tangkatan-
25	xx[j,2]))/(SQR(std[j,2))*((SQR((1+(SQR((Tangkatan-
26	xx[j,2])/std[j,2]))))));
27	end;
28	
29	if (std[j,3])=0 then
30	begin
31	ea3[i,j]:=0; ec3[i,j]:=0;
32	end
33	else
34	begin

```

35         ea3[i,j]:=((2*(SQR(Tbekerja-
36 xx[j,3]))/(Power(std[j,3],3))*((SQR((1+(SQR((Tbekerja-
37 xx[j,3])/std[j,3]))))))));
38         ec3[i,j]:=((2*(Tbekerja-
39 xx[j,3]))/(SQR(std[j,3))*((SQR((1+(SQR((Tbekerja-
40 xx[j,3])/std[j,3]))))))));
41         end;
42
43         if (std[j,4])=0 then
44         begin
45             ea4[i,j]:=0; ec4[i,j]:=0;
46             end
47
48         else
49         begin
50             ea4[i,j]:=((2*(SQR(Ttpak-
51 xx[j,4]))/(Power(std[j,4],3))*((SQR((1+(SQR((Ttpak-
52 xx[j,4])/std[j,4]))))))));
53             ec4[i,j]:=((2*(Ttpak-
54 xx[j,4]))/(SQR(std[j,4))*((SQR((1+(SQR((Ttpak-
55 xx[j,4])/std[j,4]))))))));
56             end;
57
58         if (std[j,5])=0 then
59         begin
60             ea5[i,j]:=0; ec5[i,j]:=0;
61             end
62         else
63         begin
64             ea5[i,j]:=((2*(SQR(Ttkk-
65 xx[j,5]))/(Power(std[j,5],3))*((SQR((1+(SQR((Ttkk-
66 xx[j,5])/std[j,5]))))))));
67             ec5[i,j]:=((2*(Ttkk-
68 xx[j,5]))/(SQR(std[j,5))*((SQR((1+(SQR((Ttkk-
69 xx[j,5])/std[j,5]))))))));
70             end;
71
72         if (std[j,6])=0 then
73         begin
74             ea6[i,j]:=0; ec6[i,j]:=0;
75             end
76
77         else
78         begin
79             ea6[i,j]:=((2*(SQR(Tsdm-
80 xx[j,6]))/(Power(std[j,6],3))*((SQR((1+(SQR((Tsdm-
81 xx[j,6])/std[j,6]))))))));
82             ec6[i,j]:=((2*(Tsdm-
83 xx[j,6]))/(SQR(std[j,6))*((SQR((1+(SQR((Tsdm-
84 xx[j,6])/std[j,6]))))))));
85             end;

```

**Sourcecode 4.20** Error parameter  $a_{ij}$  dan  $c_{ij}$

#### 4.5.1 Perhitungan $\Delta a_{ij}$

Fungsi untuk menghitung *error* parameter dapat dilihat pada *Sourcecode 4.21* sebagai berikut:

```
1 da1[i,j]:=LeaRat*ea1[i,j]*Tproduktif;
2 da2[i,j]:=LeaRat*ea2[i,j]*Tangkatan;
3 da3[i,j]:=LeaRat*ea3[i,j]*Tbekerja;
4 da4[i,j]:=LeaRat*ea4[i,j]*Ttpak;
5 da5[i,j]:=LeaRat*ea5[i,j]*Ttkk;
6 da6[i,j]:=LeaRat*ea6[i,j]*Tsdm;
```

*Sourcecode 4.21* Perhitungan  $\Delta a_{ij}$

#### 4.5.2 Perhitungan $\Delta c_{ij}$

Fungsi untuk menghitung  $\Delta c_{ij}$  dapat dilihat pada *Sourcecode 4.22* sebagai berikut:

```
1 dc1[i,j]:=LeaRat*ec1[i,j]*Tproduktif;
2 dc2[i,j]:=LeaRat*ec2[i,j]*Tangkatan;
3 dc3[i,j]:=LeaRat*ec3[i,j]*Tbekerja;
4 dc4[i,j]:=LeaRat*ec4[i,j]*Ttpak;
5 dc5[i,j]:=LeaRat*ec5[i,j]*Ttkk;
6 dc6[i,j]:=LeaRat*ec6[i,j]*Tsdm;
```

*Sourcecode 4.22* Perhitungan  $\Delta c_{ij}$

#### 4.5.3 Perhitungan nilai a dan c baru

Fungsi untuk menghitung nilai a dan c yang baru dapat dilihat pada *Sourcecode 4.23* sebagai berikut:

```
1 a1[i,j]:=xx1[j,1]+da1[i,j];
2 a2[i,j]:=xx1[j,2]+da2[i,j];
3 a3[i,j]:=xx1[j,3]+da3[i,j];
4 a4[i,j]:=xx1[j,4]+da4[i,j];
5 a5[i,j]:=xx1[j,5]+da5[i,j];
6 a6[i,j]:=xx1[j,6]+da6[i,j];
7 bc1[i,j]:=std[j,1]+dc1[i,j];
8 bc2[i,j]:=std[j,2]+dc2[i,j];
9 bc3[i,j]:=std[j,3]+dc3[i,j];
10 bc4[i,j]:=std[j,4]+dc4[i,j];
11 bc5[i,j]:=std[j,5]+dc5[i,j];
12 bc6[i,j]:=std[j,6]+dc6[i,j];
```

*Sourcecode 4.23* Perhitungan nilai a dan c baru

#### 4.6 Menghitung RMSE

Fungsi untuk menghitung nilai RMSE dapat dilihat pada *Sourcecode 4.24* sebagai berikut:

```

1 OJ[i]:=StrToFloat(SGDataUji.Cells[7,i]);
2   totrmse:=totrmse+SQR(OJ[i]-output_jar2[i]);
3   rmse:=sqrt(totrmse/jumlah2);
4   EdMAPE.Text:=FloatToStr(RMSE);
5   end;

```

### Sourcecode 4.24 Nilai RMSE

#### 4.6.1 Menentukan Jarak *Euclidean* dan Kelas

Fungsi untuk menghitung nilai jarak *euclidean* dapat dilihat pada **Sourcecode 4.25** sebagai berikut:

```

1  minout:=0;
2  for j:=1 to SGDataUji.RowCount-1 do
3    begin
4      for i:=1 to 5 do
5        begin
6          Jarak:= abs(StrToFloat(SGDataUji.cells[7,j])-
7  Kelas_out[i]);
8          if i=1 then
9            begin
10             minout:=jarak;
11             Kelas:=i;
12           end
13          else if jarak < minout then
14            begin
15             minout:=jarak;
16             Kelas:=i;
17           end;
18          end;
19          case kelas of
20            1: SGPengujian.cells[4,j]:='Sangat Rendah';
21            2: SGPengujian.cells[4,j]:='Rendah';
22            3: SGPengujian.cells[4,j]:='Sedang';
23            4: SGPengujian.cells[4,j]:='Tinggi';
24            5: SGPengujian.cells[4,j]:='Sangat Tinggi';
25          end;
26        end;
27        minout:=0;
28        for j:=1 to SGData.RowCount-1 do
29          begin
30            for i:=1 to 5 do
31              begin
32                Jarak:= abs(StrToFloat(SGData.cells[1,j])-
33  Kelas_out[i]);
34                if i=1 then
35                  begin
36                    minout:=jarak;
37                    Kelas:=i;
38                  end
39                else if jarak < minout then
40                  begin
41                    minout:=jarak;

```



```

42         Kelas:=i;
43     end;
44 end;
45 case kelas of
46 1: SGPengujian.cells[2,j]:='Sangat Rendah';
47 2: SGPengujian.cells[2,j]:='Rendah';
48 3: SGPengujian.cells[2,j]:='Sedang';
49 4: SGPengujian.cells[2,j]:='Tinggi';
50 5: SGPengujian.cells[2,j]:='Sangat Tinggi';
51 end;
52 end;
53 end;

```

**Sourcecode 4.25** Menentukan Jarak *Euclidean* dan Kelas

#### 4.6.2 Menghitung Akurasi

Fungsi untuk menghitung nilai akurasi dapat dilihat pada **Sourcecode 4.26** sebagai berikut:

```

1 begin
2   ujj:=0;
3   ujj2:=0;
4   for i:=1 to Q3.RecordCount do
5     begin
6       if (SGPengujian.Cells[2,i] = SGPengujian.Cells[4,i])
7       then
8         begin
9           inc(ujj);
10          akurasi:=(ujj/30)*100;
11          Edit2.Text:=FloatToStr(ujj);
12          EdAkurasi.Text:=FloatToStr(akurasi);
13        end
14        else if (SGPengujian.Cells[2,i] <>
15 SGPengujian.Cells[4,i]) then
16          begin
17            inc(ujj2);
18            Edit3.Text:=IntToStr(ujj2);
19          end;
20        end;
21      end;

```

**Sourcecode 4.26** Nilai Akurasi

#### 4.6.3 Menghitung Nilai Output Prediksi

Fungsi untuk menghitung nilai output prediksi dapat dilihat pada **Sourcecode 4.27** sebagai berikut:

```

1 Fillchar(data,sizeof(data),#0);
2   n_data:=SGPengujian.RowCount-1;
3   setlength(data,n_data);
4   for j:= 1 to SGRamal.RowCount-1 do
5     For i:= 0 to SGRamal.ColCount-1 do

```

```

6         SGRamal.Cells[i,j]:='';
7     SGRamal.RowCount:=2;
8     for i:= 1 to SGPengujian.RowCount-1 do
9         data[i-1]:= strtofloat(SGPengujian.Cells[1,i]);
10    for i:= 1 to strtoint(ComboBox1.Text) do
11        begin
12            if i=1 then
13                begin
14                    tahun:=
15                    strtoint(copy(SGPengujian.Cells[0,n_data],1,4));
16                    bulan:=
17                    strtoint(copy(SGPengujian.Cells[0,n_data],5,2));
18                    end
19                else
20                begin
21                    tahun:= strtoint(copy(SGRamal.Cells[0,i-
22                    1],1,4));
23                    bulan:= strtoint(copy(SGRamal.Cells[0,i-
24                    1],5,2));
25                    end;
26                if bulan < 12 then inc(Bulan)
27                else
28                begin
29                    bulan:=1;
30                    inc(tahun);
31                end;
32                SGRamal.Cells[0,i]:=copy('0000',1,4-
33                length(inttostr(tahun)) + inttostr(tahun);
34                SGRamal.Cells[0,i]:= SGRamal.Cells[0,i] +
35                copy('00',1,2-length(inttostr(bulan)) + inttostr(bulan);
36                if i > 1 then SGRamal.RowCount:=SGRamal.RowCount
37                +1;
38                sum_data:=0;
39                for j:= low(data) to high(data) do
40                    Sum_Data:= Sum_Data + data[j];
41                Hasil:= Sum_Data/(j);
42                if i=1 then
43                Lastdata:=strtofloat(SGPengujian.Cells[1,n_data])
44                else Lastdata:=strtofloat(SGRamal.Cells[1,i-1]);
45                SGRamal.Cells[1,i]:=floattostr(Lastdata + hasil);
46                minout:=0;
47                for j:=1 to 5 do
48                begin
49                    Jarak:= abs(StrToFloat(SGRamal.cells[1,i])-
50                    Kelas_out[j]);
51                    if j=1 then
52                    begin
53                        minout:=jarak;
54                        Kelas:=j;
55                    end
56                    else if jarak < minout then
57                    begin
58                        minout:=jarak;
59                        Kelas:=j;
60                    end;

```

```

61     end;
62     case kelas of
63     1: SGRamal.cells[2,i]:='Sangat Rendah';
64     2: SGRamal.cells[2,i]:='Rendah';
65     3: SGRamal.cells[2,i]:='Sedang';
66     4: SGRamal.cells[2,i]:='Tinggi';
67     5: SGRamal.cells[2,i]:='Sangat Tinggi';
68     end;
69
70     inc(n_data);
71     setlength(data,n_data);
72     data[n_data-1]:=strtofloat(SGRamal.Cells[1,i]);
73     end;
74 end;

```

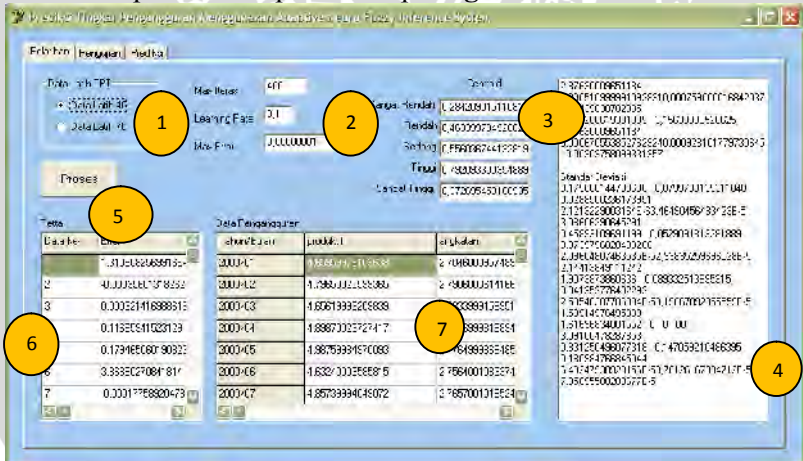
**Sourcecode 4.27** Nilai Output Prediksi

### 4.7 Implementasi Antarmuka

Berdasarkan rancangan antarmuka yang telah dijelaskan pada BAB III dihasilkan antarmuka yaitu:

#### 4.7.1 Form Pelatihan

Form Pelatihan yaitu form yang digunakan untuk memasukkan data parameter kedalam sistem yang akan dilatih. Proses data pelatihan dapat dilihat pada **gambar 4.1**.



**Gambar 4.1** Form Pelatihan

Keterangan gambar 4.1:

10. Memilih jumlah data yang akan digunakan untuk pelatihan.

11. Terdiri dari :

- maximal iterasi digunakan untuk menentukan banyaknya iterasi.
- *learning rate* digunakan untuk menentukan besarnya laju pembelajaran.
- manimum error digunakan untuk menentukan minimal error yang dicapai pada proses pelatihan.

12. Menampilkan *Centroid cluster* yang terbentuk.

13. Memo untuk menampilkan perhitungan *mean* dan standar deviasi setelah diproses.

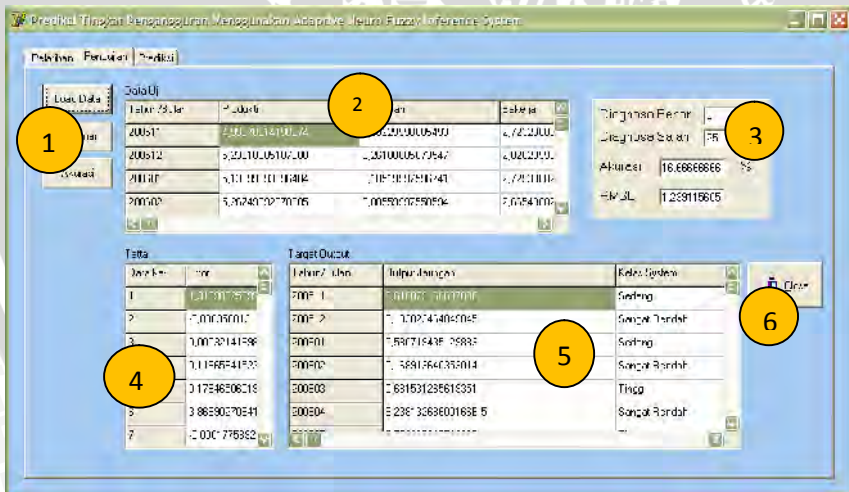
14. Tombol untuk proses data pelatihan sesuai jumlah data yang dilatih.

15. Menampilkan Error Lapisan pada parameter konsekuensi yang akan dipakai untuk pengujian.

16. Menampilkan data latih yang digunakan setelah dilakukan pelatihan data.

#### 4.7.2 Form Pengujian

Form pengujian pada sistem ini berguna untuk melakukan pengujian prediksi tingkat pengangguran. Proses pengujian dapat dilihat pada **gambar 4.2**.



**Gambar 4.2** Form Pengujian

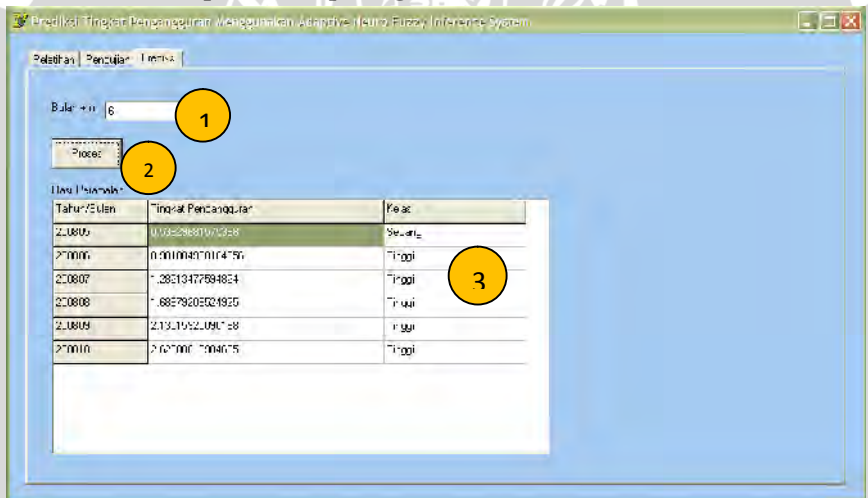
Keterangan gambar 4.2 :

12. Terdiri dari

- Tombol *load* data digunakan untuk melakukan *load* data yang akan digunakan dalam pengujian.
  - Tombol pengujian untuk uji *ANFIS* dari data yang diujikan.
  - Tombol akurasi untuk melihat besar akurasi dari sistem *ANFIS*.
13. Menampilkan data pengujian aktual.
  14. Menampilkan jumlah data yang benar, yang salah setelah diujikan dan menampilkan akurasi dan nilai RMSE.
  15. Menampilkan error yang telah dilatihkan pada proses pelatihan
  16. Menampilkan *output* jaringan.
  17. Menutup program.

### 4.7.3 Form Prediksi

Form Prediksi ini digunakan untuk meramalkan kondisi atau keadaan tingkat pengangguran sesuai periode waktu yang diinginkan. Form Prediksi dapat dilihat pada **gambar 4.3**



**Gambar 4.3** Form Prediksi

Keterangan gambar 4.3:

4. Menampilkan periode waktu prediksi yang diinginkan.
5. Tombol untuk memproses prediksi.
6. Menampilkan hasil prediksi berupa besaran tingkat pengangguran dan kategori kelas.

## 4.8 Hasil Pengujian

Untuk memperoleh struktur jaringan *ANFIS* yang terbaik yang digunakan untuk prediksi tingkat pengangguran maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan melatih jaringan *ANFIS* dengan parameter-parameter yang berbeda, yang nantinya akan diambil satu struktur jaringan yang terbaik yang digunakan untuk melakukan prediksi tingkat pengangguran.

### 4.8.1 Hasil Percobaan

Pada percobaan kali ini dilakukan sebanyak 5 kali pada 2 jumlah data latih yang berbeda yaitu 40 data dan 70 data.

#### 4.8.1.1 Pengaruh Nilai *Learning Rate*

Pada percobaan ini akan dilakukan perbandingan nilai RMSE yang diperoleh dari uji coba beberapa nilai *learning rate* yaitu antara 0,00001 sampai 0,00009. Percobaan dilakukan sebanyak 5 kali dimulai dengan *learning rate* terkecil sampai *learning rate* terbesar. Percobaan dilakukan dengan maksimum *epoch* atau iterasi sebanyak 400 iterasi hasil percobaan ditunjukkan pada **Tabel 4.1.**

**Tabel 4.1** Nilai *Learning Rate* terhadap RMSE

LR	RMSE
	40 data
0,1	1,341
0,2	1,367
0,3	1,316
0,4	1,3
0,5	1,298
0,6	1,274
0,7	1,284
0,8	1,281
0,9	1,303

Pada tabel 4.1 ditunjukkan bahwa nilai RMSE paling rendah terdapat pada *learning rate* 0,6 yaitu sebesar 1,274. Sedangkan nilai RMSE paling tinggi adalah 1,367 yaitu diperoleh pada *learning rate*

0,2 Sehingga *learning rate* optimal yang digunakan untuk pelatihan ANFIS adalah 0,6.

#### 4.8.1.2 Pengaruh Jumlah Data Latih

Dari hasil pelatihan didapatkan bobot terbaik yang akan digunakan untuk menguji kemampuan sistem dalam memberikan prediksi tingkat pengangguran. Berikut ini dilakukan pengujian kebenaran prediksi tingkat pengangguran terhadap 30 data uji menggunakan arsitektur ANFIS yang didapatkan dari pelatihan yang sudah dilakukan yaitu dengan menggunakan nilai *learning rate* sebesar 0.6.

Pengujian dilakukan dengan data latih yang berbeda. Data latih yang digunakan adalah data 40 dan data 70. Satu kali percobaan yaitu dengan menguji data yang sama dengan data uji sebanyak 30. Data uji yang digunakan adalah 30 data yang belum pernah dilatih sebelumnya. Hasil pengujian pengaruh jumlah data latih terhadap akurasi sistem terlihat pada **tabel 4.2**.

**Tabel 4.2** Banyak data latih terhadap akurasi

Data Latih	Data Uji	Benar	Salah	Akurasi
40	30	7	23	23,33%
70	30	28	2	93,33%

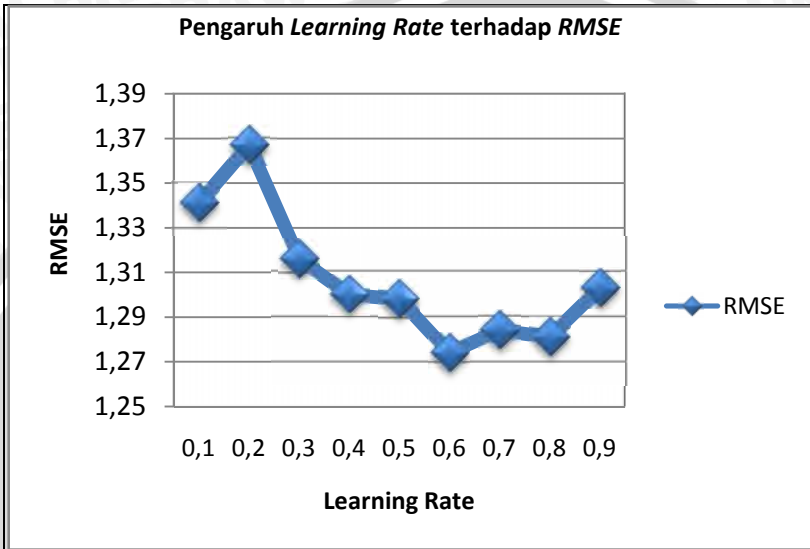
Hal ini menunjukkan bahwa makin banyak data yang dilatihkan dengan data uji yang sama maka tingkat akurasinya lebih besar. Sehingga metode ANFIS dalam prediksi tingkat pengangguran cukup baik digunakan untuk memprediksi data yang telah dilatih sebelumnya.

#### 4.8.2 Analisa Hasil

##### 4.8.2.1 Analisa Hasil *Learning Rate*

Pada hasil uji coba pengaruh *learning rate* terhadap RMSE pada **tabel 4.1** terlihat bahwa penambahan nilai *learning rate*  $\leq 0.6$  mengalami penurunan nilai RMSE, sedangkan penambahan nilai *learning rate* diatas 0.6 mengalami kenaikan MSE . Hal ini terjadi karena dengan penambahan nilai *learning rate* diatas 0.6 proses pelatihan telah melampaui keadaan optimal. Pada Gambar 4.5

disajikan grafik hubungan antara *learning rate* terhadap perubahan MSE.

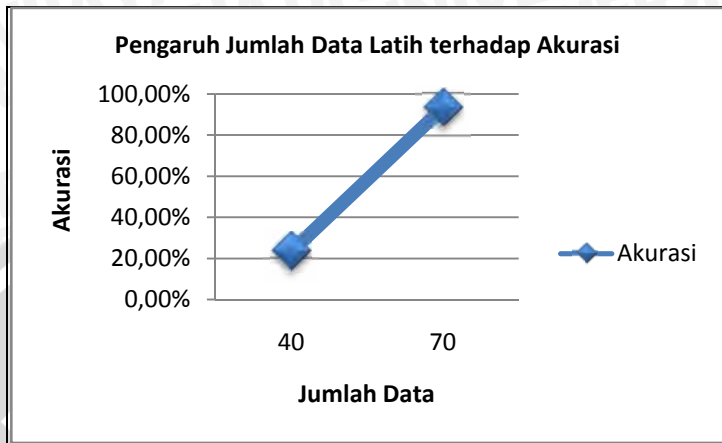


**Gambar 4.4** Grafik pengaruh penambahan *learning rate* terhadap perubahan MSE.

#### 4.8.2.2 Analisa Hasil Jumlah Data Latih

Pada hasil uji coba pengaruh jumlah data latih terhadap akurasi sistem pada **tabel 4.2** dapat dilihat bahwa semakin banyak jumlah data latih semakin baik tingkat akurasinya karena dengan melakukan proses pembelajaran menggunakan banyak data latih akan dapat dicapai nilai *error* yang minimum. Hasil pengenalan sistem tertinggi dalam memberikan prediksi tingkat pengangguran adalah sebesar 93.33%. Hal ini menunjukkan bahwa metode ANFIS dapat memprediksi tingkat pengangguran dengan baik karena menghasilkan tingkat akurasi yang cukup tinggi. Berdasarkan **tabel 4.2** dapat dibuat grafik hubungan antara jumlah data latih dengan akurasi sistem pada **gambar 4.5**.





**Gambar 4.5** Grafik pengaruh jumlah data latih terhadap akurasi

Berikut ini ditunjukkan hasil percobaan kebenaran prediksi tingkat pengangguran terhadap data uji. Hasil pengujian untuk 40 data latih dapat dilihat pada **tabel 4.3.** dan **tabel 4.4** untuk kebenaran prediksi tingkat pengangguran 70 data latih.

**Tabel 4.3** Perbandingan Output Jaringan dengan Aktual 40 Data

No	Output Jaringan	Aktual	No	Output Jaringan	Aktual
1	sangat tinggi	rendah	16	sedang	rendah
2	tinggi	rendah	17	rendah	<i>rendah</i>
3	sangat tinggi	rendah	18	sangat tinggi	rendah
4	sangat tinggi	rendah	19	sangat tinggi	rendah
5	sangat tinggi	rendah	20	sangat tinggi	rendah
6	sangat tinggi	rendah	21	rendah	<i>rendah</i>
7	rendah	<i>rendah</i>	22	sangat tinggi	rendah
8	sangat tinggi	rendah	23	sangat tinggi	rendah
9	rendah	<i>rendah</i>	24	sangat tinggi	rendah
10	sangat tinggi	rendah	25	sangat tinggi	rendah
11	rendah	<i>rendah</i>	26	rendah	<i>rendah</i>
12	sangat tinggi	rendah	27	sangat tinggi	rendah
13	sangat tinggi	rendah	28	sangat tinggi	tinggi

14	sangat endah	rendah	29	sangat tinggi	tinggi
15	sangat tinggi	rendah	30	rendah	rendah

**Tabel 4.3** Perbandingan Output Jaringan dengan Aktual 70 Data

No	Output Jaringan	Aktual	No	Output Jaringan	Aktual
1	sedang	sedang	16	sedang	sedang
2	sedang	sedang	17	sedang	sedang
3	sedang	sedang	18	sedang	sedang
4	sedang	sedang	19	sedang	sedang
5	sedang	sedang	20	sedang	sedang
6	sedang	sedang	21	sangat rendah	<i>sedang</i>
7	rendah	<i>sedang</i>	22	sedang	sedang
8	sedang	sedang	23	sedang	sedang
9	sedang	sedang	24	sedang	sedang
10	sedang	sedang	25	sedang	sedang
11	sedang	sedang	26	sedang	sedang
12	sedang	sedang	27	sedang	sedang
13	sedang	sedang	28	sedang	sedang
14	sedang	sedang	29	sedang	sedang
15	sedang	sedang	30	sedang	sedang

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Kesimpulan yang diperoleh dari tugas akhir ini:

1. Metode Adaptive Neuro Fuzzy Inference System (*ANFIS*) diimplementasikan untuk memprediksi tingkat pengangguran kota Malang dengan fungsi keanggotaan *generalized bell*, sistem inferensi *fuzzy* SUGENO Orde-Satu, fungsi aktivasi *sigmoid biner* dan 6 parameter yang berkaitan erat dengan pengangguran.
2. Hasil dari implementasi metode *ANFIS* ini diperoleh kombinasi learning rate sebesar 0,9 dengan jumlah data latih sebanyak 70 data. Dari hasil kombinasi menghasilkan RMSE pelatihan sebesar 1,274 dan mampu memprediksi dengan baik dengan tingkat akurasi sebesar 93,33% dari 30 data uji.

### 5.2 Saran

Saran yang diberikan pada pembuatan skripsi ini:

1. Dalam tugas akhir ini, sistem hanya menggunakan data-data time series untuk melakukan prediksi. Untuk percobaan berikutnya akan lebih baik jika parameter yang sangat erat kaitannya dengan pengangguran di tambahkan.
2. Akurasi yang didapat dengan menggunakan *Adaptive Neuro Fuzzy Inference System (ANFIS)* pada prediksi tingkat pengangguran cukup baik namun demikian perlu ada metode lain sebagai pembandingan untuk penelitian selanjutnya.

## DAFTAR PUSTAKA

- Ade, Murti. 2003. *Kemiskinan dan Pengangguran*. Gunadarma. Depok.
- Bambang, Budi DP. 1999. Teknik Jaringan Syaraf Tiruan Feedforward Untuk Prediksi Harga Saham Pada Pasar Modal Indonesia. *Jurnal Informatika* Vol. 1, No. 1, Mei 1999 : 11 – 22
- Eliyani. *Pengantar Jaringan Syaraf Tiruan*. 2005.  
<http://materikulia.com/researchreport/article/viewFile/192/076>.  
Diakses : 10 Agustus 2011
- Freeman, J. Skapura, D. 1991. *Neural Networks Algorithms, Applications, and Programming Techniques*. Addison-Wesley. USA.
- Halim, Siana. 2006. *Diklat-Time Series Analysis*. Petra. Surabaya.
- Hermawan, Arif. 2006. *Jaringan Saraf Tiruan: Teori dan Aplikasi*. Andi: Yogyakarta.
- Jawa Timur, BPS. 2001. *Hasil Survei Sosial Ekonomi Nasional Tahun 2001 Propinsi Jawa Timur*. BPS Jawa Timur, Surabaya.
- Kristanto, Andri. 2004. *Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma, dan Aplikasi)*. GAYAMEDIA: Yogyakarta.
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik & Aplikasinya)*. Graha Ilmu. Yogyakarta.
- Kusumadewi, S dan S. Hartati. 2006. *Neuro-Fuzzy*. Graha Ilmu: Yogyakarta.
- Muis, S. 2006. *Teknik Jaringan Syaraf Tiruan*. GRAHA ILMU. Yogyakarta.

Muis, S. 2006. Jaringan Syaraf Tiruan Sebagai Alat Bantu Peramalan Harga Saham. GRAHA ILMU. Yogyakarta.

Suyanto. 2007. *Artificial Intelligence* . Informatika. Bandung

Suyanto. 2008. Soft Computing Membangun Mesin Ber-IQ Tinggi. Informatika Bandung. Bandung.

Sutanto Teguh, Dyan Novita Kartika Sari. 2003. *Implementasi Sistem Neuro-Fuzzy Untuk Prediksi Produksi Air Minum di PDAM Surabaya*.Stikom. Surabaya.

<http://eprints.upnjatim.ac.id/1217/1/teguh.pdf>. Diakses : 11 November 2011

Sutikno. Analisis Peran Sektor Ekonomi Terhadap Penyerapan Tenaga Kerja.UMM. Malang.

<http://research-report.umm.ac.id/index.php/research-report/article/viewFile/153/176>. Diakses : 28 Oktober 2011

Syarifudin. 2007. *Aplikasi Adaptif Neuro Fuzzy Inference System (ANFIS) Dalam Proses Penyulingan Kolom yang Berkelanjutan dan Kolom Batch*.UIN.Malang.

(<http://lib.uin-malang.ac.id/thesis/fullchapter/03510023-syarifudin.ps>. Diakses : 11 November 2011)