



**UNIVERSITAS INDONESIA**

**PENGEMBANGAN METODE JARINGAN SARAF  
TIRUAN DENGAN FUNGSI BASIS RADIAL (FBR)  
FUZZY DAN APLIKASINYA**

**TESIS**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Magister Teknik**

**YOAN ELVIRALITA**

**0906578270**

**FAKULTAS TEKNIK**

**PROGRAM STUDI TEKNIK ELEKTRO**

**DEPOK**

**JULI 2011**

## HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri, dan semua sumber yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Yoan Elviralita

NPM : 0906578270

Tanda Tangan :

Tanggal : Juli 2011

## LEMBAR PENGESAHAN

Tesis ini diajukan oleh :  
Nama : Yoan Elviralita  
NPM : 0906578270  
Program Studi : Teknik Elektro  
Judul Tesis : Pengembangan Jaringan Saraf Tiruan Dengan Fungsi Basis Radial (FBR) Fuzzy dan Aplikasinya

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

### DEWAN PENGUJI

Pembimbing : Prof. Dr.Eng Drs. Benyamin Kusumoputro, M.Eng ( )  
Penguji : Ir. Wahidin Wahab, MSc, PhD ( )  
Penguji : Dr. Ir. Feri Yusivar, M.Eng ( )  
Penguji : Aries Subiantoro, ST, M.Sc ( )

Ditetapkan di : Depok

Tanggal : Juli 2011

## KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas segala kasih sayang dan kemurahan-Nya sehingga saya dapat menyelesaikan tesis ini. Penulisan tesis ini adalah salah satu syarat untuk mendapatkan gelar Magister Teknik di Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dari berbagai pihak sangat sulit bagi saya untuk dapat menyelesaikan penulisan tesis ini. Oleh karena itu saya mengucapkan banyak terima kasih kepada:

- (1) Prof. Dr.Eng Drs. Benyamin Kusumoputro, M.Eng selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikirannya untuk mengarahkan saya dalam penulisan tesis ini;
- (2) Bapak dan Ibu saya yang begitu kuat dalam memberikan dukungan;
- (3) Suami saya tercinta Asrul Hidayat dan Putra-putri saya tercinta Muhammad Ariq Fauzan dan Atiqa Kayla Mecca yang begitu kuat dalam memberikan dukungan dan;
- (4) Keluarga besar saya yang selalu mendoakan saya agar selalu berhasil.

Akhir kata, semoga Allah SWT berkenan membalas kebaikan berlipat-lipat bagi semua pihak yang telah membantu. Semoga tesis ini membawa manfaat sebesar-besarnya.

Depok, Juli 2011

Penulis

(YoanElviralita)

**HALAMAN PERSETUJUAN PUBLIKASI TESIS UNTUK  
KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan  
di bawah ini:

Nama : Yoan Elviralita

NPM : 0906578270

Program Studi : Kontrol Industri

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul : Pengembangan Jaringan Saraf Tiruan Dengan Fungsi Basis Radial (FBR) Fuzzy Dan Aplikasinya. Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juli 2011

Yang Menyatakan

(Yoan Elviralita)

## ABSTRAK

Nama : Yoan Elviralita  
Program Studi : Teknik Elektro  
Judul : Pengembangan Jaringan Saraf Tiruan Dengan Fungsi Basis Radial (FBR) Fuzzy dan Aplikasinya

Dalam beberapa tahun ini, telah banyak penelitian yang berhubungan dengan pengenalan pola dilakukan untuk mengidentifikasi berbagai macam bentuk pola. Tesis ini membahas pengembangan jaringan saraf tiruan fungsi basis radial fuzzy. Dalam penelitian ini dilakukan dua percobaan, yaitu jaringan saraf fungsi basis radial fuzzy menggunakan SOM dan jaringan saraf fungsi basis radial fuzzy tanpa SOM. Hasil yang dicapai dari *recognition rate* menunjukkan jaringan saraf fungsi basis radial fuzzy menggunakan SOM memberikan performa yang baik. Jaringan saraf ini diharapkan dapat dikembangkan oleh peneliti-peneliti yang lain untuk kemajuan keilmuan dalam segala bidang.

Kata kunci:

Jaringan saraf tiruan, Swa-Organisasi, Propagasi balik, Fungsi basis radial, Fuzzy.

## ABSTRACT

Name : Yoan Elviralita

Study Program : Electrical Engineering

Title : Development of Neural Network With Radial Basis Function (RBF) Fuzzy And its Application

In recent years, has been much research related to pattern recognition performed to identify various forms of patterns. This thesis discusses the development of artificial neural networks fuzzy radial basis functions. In this study conducted two experiments, namely radial basis function neural network fuzzy neural network using the SOM and fuzzy radial basis function without SOM. The result of recognition rate shows the radial basis function neural networks using a fuzzy SOM gives a good performance. Neural network is expected to be developed by other researchers for the advancement of knowledge in all fields.

Keyword:

Neural Network, Self-Organizing Maps, *Backpropagation*, Fuzzy.

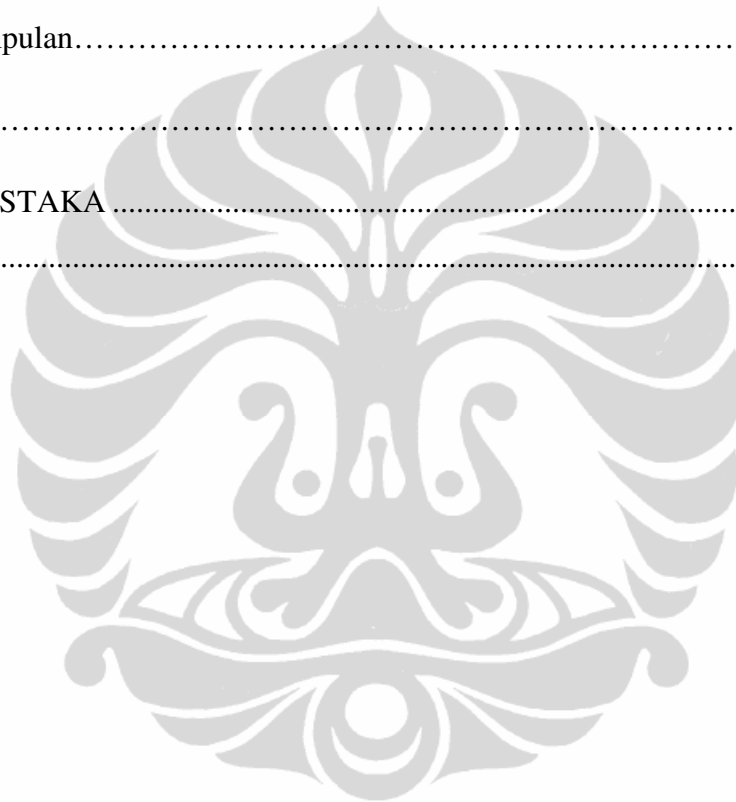
## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	ii
LEMBAR PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	2
1.3. Tujuan Penelitian .....	2
1.4. Manfaat Penelitian .....	2
1.5. Batasan Penelitian.....	2
1.6. Model Operasional Penelitian.....	2
BAB 2 DASAR TEORI .....	4
2.1. Konsep Dasar Jaringan Saraf Tiruan.....	4
2.2 Metode Pelatihan dan Model Jaringan Saraf Tiruan.....	6
2.3 Fungsi Keanggotaan.....	15
BAB 3 ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS RADIAL MENGGUNAKAN ALGORITMA SOM.....	17
3.1. Skema Penelitian Jaringan Saraf FBR Menggunakan SOM.....	17
3.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR dengan Menggunakan SOM.....	18
3.3. Jaringan Saraf FBR Menggunakan SOM.....	19
3.4. Pengujian pada Jaringan Saraf Tiruan FBR Menggunakan SOM .....	24
BAB 4 ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS RADIAL TANPA ALGORITMA SOM. ....	28



4.1.	Skema Jaringan Saraf FBR Tanpa Menggunakan SOM.....	28
4.2.	Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Tanpa Menggunakan SOM.....	29
4.3.	Jaringan Saraf FBR Tanpa Algoritma SOM .....	29
4.4.	Pengujian Pada Jaringan Saraf Tiruan FBR tanpa menggunakan SOM.....	32
<b>BAB 5 ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS</b>		
	<b>RADIAL FUZZY MENGGUNAKAN SOM.....</b>	<b>36</b>
5.1.	Skema Jaringan Saraf FBR Fuzzy Menggunakan SOM.....	36
5.2.	Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Fuzzy SOM.....	37
5.3.	Jaringan Saraf Tiruan FBR Fuzzy SOM .....	38
5.4.	Pengujian Pada Jaringan Saraf Tiruan FBR Fuzzy SOM .....	42
<b>BAB 6 ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS</b>		
	<b>RADIAL FUZZY TANPA SOM .....</b>	<b>46</b>
6.1.	Skema Jaringan Saraf FBR Fuzzy Tanpa SOM.....	46
6.2.	Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Fuzzy Tanpa SOM.....	47
6.3.	Jaringan Saraf Tiruan FBR Fuzzy Tanpa SOM .....	47
6.4.	Pengujian Pada Jaringan Saraf Tiruan FBR Fuzzy Tanpa SOM .....	51
<b>BAB 7 ANALISA PERFORMA JARINGAN SARAF <i>BACKPROPAGATION</i></b>		
7.1.	Skema Penelitian Jaringan Saraf <i>Backpropagation</i> .....	55
7.2.	Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf <i>Backpropagation</i> .....	56
7.3.	Jaringan Saraf Tiruan <i>Backpropagation</i> .....	57
7.4.	Pengujian Pada Jaringan Saraf Tiruan <i>Backpropagation</i> .....	59
<b>BAB 8 PERBANDINGAN PERFORMA JARINGAN SARAF TIRUAN .....</b>		
8.1.	Perbandingan Metode FBR Menggunakan SOM dan FBR Tanpa menggunakan SOM.....	63

8.2. Perbandingan Metode FBR Fuzzy Menggunakan SOM dan FBR Fuzzy Tanpa Menggunakan SOM.....	64
8.3 Perbandingan Metode Fuzzy dan Non Fuzzy.....	65
8.4. Perbandingan Metode FBR Fuzzy Menggunakan SOM dan Metode Propagasi balik.....	65
<b>BAB 9 KESIMPULAN DAN SARAN.....</b>	<b>68</b>
9.1. Kesimpulan.....	68
9.2. Saran.....	68
<b>DAFTAR PUSTAKA.....</b>	<b>69</b>
<b>LAMPIRAN.....</b>	<b>.....</b>



## DAFTAR GAMBAR

Gambar 2.1.Struktur Dasar JST dan Struktur Sederhana Sebuah Neuron [1]..	4
Gambar 2.2.Model Tiruan Sebuah Neuron [1] .....	5
Gambar 2.3.Struktur jaringan Saraf Tiruan <i>Backpropagation</i> [2] .....	6
Gambar 2.4.Struktur jaringan Saraf Tiruan SOM .....	12
Gambar 2.5.Struktur jaringan Saraf Tiruan FBR .....	14
Gambar 3.1. Skema penelitian .....	17
Gambar 3.2. Grafik Recognition Rate Testing Data Original untuk 1000 epoch .....	26
Gambar 3.3. Grafik Recognition Rate Testing Data Normalisasi untuk 1000 epoch.....	27
Gambar 4.1. Grafik Recognition Rate Testing Data Original untuk 1000 epoch .....	34
Gambar 4.2. Grafik Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	35
Gambar 5.1. Grafik Recognition Rate Testing Data Original untuk 1000 epoch .....	44
Gambar 5.2. Grafik Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	44
Gambar 6.1. Grafik Recognition Rate Testing Data Original untuk 1000 epoch .....	53
Gambar 6.2. Grafik Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	53
Gambar 7.1. Grafik Recognition Rate Testing Data Original untuk 1000 epoch.....	61
Gambar 7.2. Grafik Recognition Rate Testing Data Normalisasi 1000 epoch.....	61

## DAFTAR TABEL

Tabel 3.1 Recognition Rate Testing Data Original untuk 1000 epoch .....	25
Tabel 3.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	26
Tabel 4.1 Recognition Rate Testing Data Original untuk 1000 epoch .....	33
Tabel 4.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	34
Tabel 5.1 Recognition Rate Testing Data Original untuk 1000 epoch .....	43
Tabel 5.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	43
Tabel 6.1 Recognition Rate Testing Data Original untuk 1000 epoch .....	52
Tabel 6.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch .....	52
Tabel 7.1 Recognition Rate Testing Data Original untuk 1000 epoch .....	60
Tabel 7.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch.....	60
Tabel 8.1 Perbandingan Recognition rate Testing FBR SOM dan FBR Tanpa SOM.....	63
Tabel 8.2 Perbandingan Recognition Rate Testing FBR Fuzzy SOM dan FBR Fuzzy Tanpa SOM.....	64
Tabel 8.3 Perbandingan Recognition Rate Testing FBR Fuzzy SOM Propagasi Balik.....	66

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Secara umum terdapat dua tipe pelatihan jaringan saraf tiruan, yaitu pelatihan dengan pengarahan (*supervised training*) dan pelatihan tanpa pengarahan (*unsupervised training*). Salah satu jaringan saraf tiruan yang menggunakan algoritma pelatihan dengan pengarahan adalah jaringan saraf propagasi balik (*backpropagation neural network*), sedangkan yang menggunakan algoritma pelatihan tanpa pengarahan adalah jaringan saraf swa-organisasi (*self-organizing neural network*). Jaringan saraf propagasi balik memiliki kemampuan generalisasi yang baik, namun memerlukan waktu pelatihan yang lama. Di lain pihak, jaringan saraf swa-organisasi memiliki kemampuan mengelompokkan data masukan dan memerlukan waktu pelatihan yang singkat. Dengan mengkombinasikan kedua jaringan saraf ini dapat dihasilkan suatu jaringan saraf hibrida Fungsi Basis Radial yang memiliki kemampuan generalisasi yang baik dan waktu pelatihan yang singkat.

Penelitian ini dilakukan untuk mengembangkan jaringan saraf hibrida Fungsi Basis Radial untuk mengatasi permasalahan dalam sistem pengendalian di mana seringkali parameter-parameternya tidak diketahui atau sulit untuk dilakukan serta rumitnya perhitungan model matematik pada sistem control.

Makalah ini menawarkan alternatif lain dari jaringan saraf Fungsi Basis Radial ini dengan mengkombinasikan jaringan saraf Propagasi Balik dan jaringan saraf swa-organisasi. Di samping sebagai alternatif lain, jaringan ini juga diharapkan dapat memperbaiki jaringan saraf hibrida yang telah dikembangkan sebelumnya.

## 1.2. Perumusan Masalah

Secara umum, masalah yang dikaji dan diselesaikan dalam tesis ini adalah bagaimana jaringan saraf tiruan fungsi basis radial dapat digunakan untuk pengenalan pola yang dimensi berbeda.

## 1.3. Tujuan Penelitian

Tujuan penelitian ini adalah menerapkan jaringan saraf tiruan Fungsi basis radial fuzzy dalam sistem pengenalan pola. Aplikasi dilatih dengan memasukkan beberapa jenis pola. Setelah proses pelatihan selesai, pola yang ingin dicari informasinya dimasukan ke dalam aplikasi dan dilihat apakah aplikasi mampu mengenali pola tersebut. Aplikasi diharapkan mampu mengklasifikasikan pola pola yang diujikan dengan baik.

## 1.4. Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Penelitian diharapkan agar pembaca dapat memahami penggunaan jaringan saraf tiruan fungsi basis radial yang menggabungkan jaringan saraf swa-organisasi dengan jaringan saraf *backpropagation*.
2. Penelitian dapat digunakan sebagai bahan pembelajaran untuk peneliti yang ingin mempelajari sistem pengenalan pola secara umum.
3. Penelitian dapat dijadikan acuan atau referensi untuk pengembangan aplikasi sistem pengenalan pola dalam berbagai bidang.

## 1.5. Batasan Penelitian

Berikut ini adalah batasan-batasan masalah yang dilakukan pada penelitian ini :

1. Data Odor 2 campuran

Ada 3 aroma (odor) alami yang diuji yaitu kenanga , jeruk dan mawar

Masing-masing aroma dicampur dengan alkohol dengan 6 persentase yang berbeda sehingga secara keseluruhan didapatkan 18 jenis odor, di mana masing-masing jenis odor diambil 200 data eksperimen

- Eksperimen dengan 16 dimensi (yang mewakili 16 sensor odor yang digunakan)
  - Eksperimen dengan 8 dimensi (yang mewakili 8 sensor odor yang digunakan)
  - Eksperimen dengan 4 dimensi (yang mewakili 4 sensor odor yang digunakan)
2. Data aroma jeruk 0% alkohol, jeruk 15% alkohol dan jeruk 25% alkohol, masing-masing kelas sebanyak 50 data dan 4 dimensi (odor\_4d\_a)
  3. Data aroma jeruk 0% alkohol, mawar 0% dan kenanga 0% alkohol, masing-masing kelas sebanyak 50 data dan 4 dimensi (odor\_4d\_b)
  4. Data Iris 4 dimensi sebagai data pembanding dengan odor 4 dimensi (Iris\_4d)

#### **1.6. Model Operasional Penelitian**

Penelitian dilakukan dengan menggunakan metode studi pustaka, yaitu dengan mengumpulkan data, membaca buku serta artikel yang berhubungan dengan teori dasar matriks, kecerdasan buatan, jaringan saraf tiruan dan bahasa pemrograman MATLAB. Data yang didapatkan kemudian diolah menggunakan algoritma jaringan saraf fungsi basis radial Swa-organisasi – Propagasi balik dengan menggunakan bahasa pemrograman Matlab sebagai aplikasi pengenalan pola.

## BAB 2 DASAR TEORI

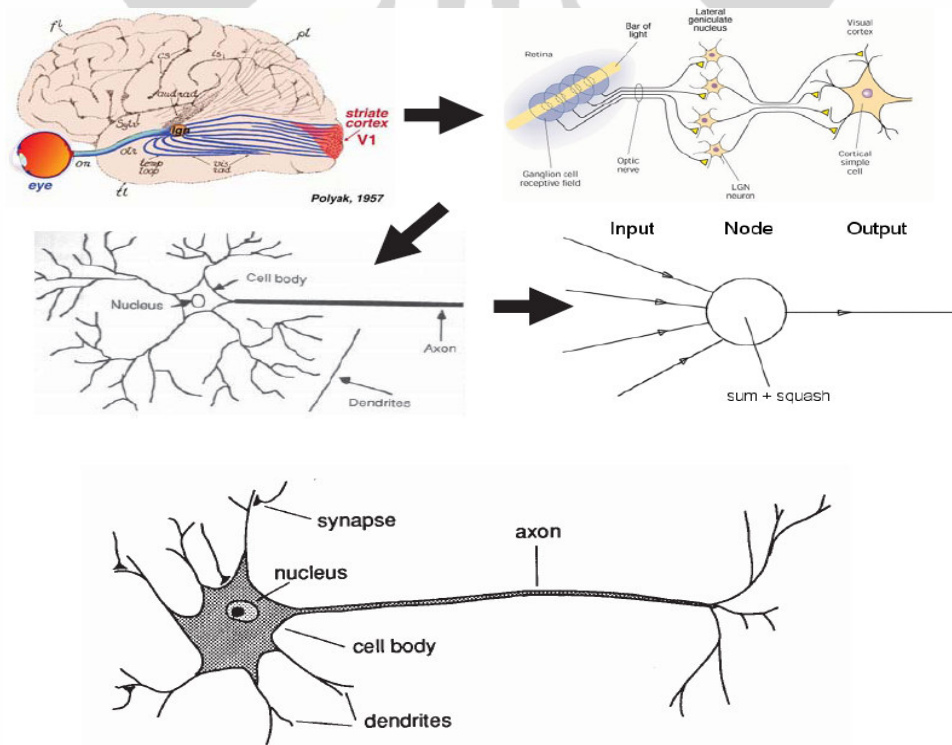
Bab ini menjelaskan tentang teori jaringan saraf tiruan dengan algoritma *backpropagation* dan algoritma Self-organizing Maps.

### 2.1. Konsep Dasar Jaringan Saraf Tiruan

#### 2.1.1 Struktur Dasar Jaringan Biologi

Pembuatan struktur jaringan saraf tiruan diilhami oleh struktur jaringan biologi, khususnya jaringan otak manusia.

Neuron adalah satuan unit pemroses terkecil pada otak, bentuk sederhana sebuah neuron yang oleh para ahli dianggap sebagai satuan unit pemroses tersebut di gambarkan seperti gambar 2.1.

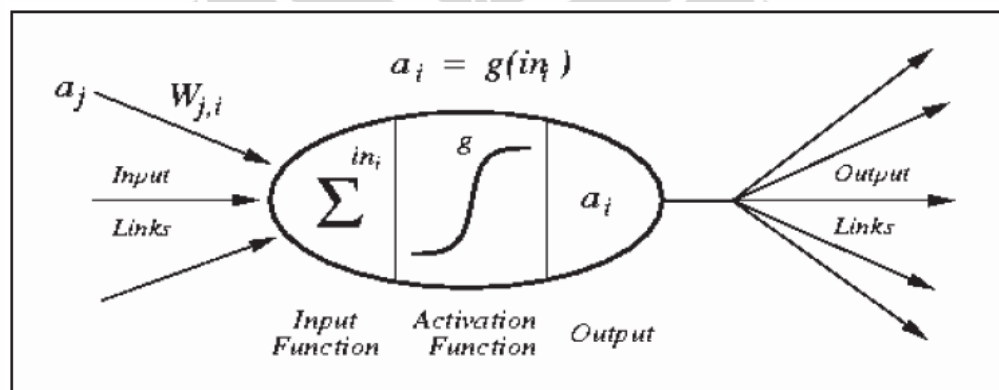


Gambar 2.1 Struktur Dasar JST dan Struktur sederhana sebuah Neuron [1]



### 2.1.2 Konsep Dasar Pemodelan Jaringan Syaraf Tiruan

Tiruan neuron dalam struktur jaringan saraf tiruan adalah sebagai elemen pemroses seperti pada gambar 2.2 yang dapat berfungsi seperti halnya sebuah neuron. Sejumlah sinyal masukan  $a$  dikalikan dengan masing-masing penimbang yang bersesuaian  $w$ . Kemudian dilakukan penjumlahan dari seluruh hasil perkalian tersebut dan keluaran yang dihasilkan dilakukan kedalam fungsi pengaktif untuk mendapatkan tingkatan derajat sinyal keluarannya  $F(a,w)$ . Walaupun masih jauh dari sempurna, namun kinerja dari tiruan neuron ini identik dengan kinerja dari sel biologi yang kita kenal saat ini [1].



Gambar 2.2 Model Tiruan Sebuah Neuron [1]

- $a_j$  : Nilai aktivasi dari unit  $j$
- $w_{j,i}$  : Bobot dari unit  $j$  ke unit  $i$
- $in_i$  : Penjumlahan bobot dan masukan ke unit  $i$
- $g$  : Fungsi aktivasi
- $a_i$  : Nilai aktivasi dari unit  $i$

Jaringan saraf Tiruan ini merupakan generalisasi dari pemodelan matematis dalam proses kognitif berdasarkan asumsi:

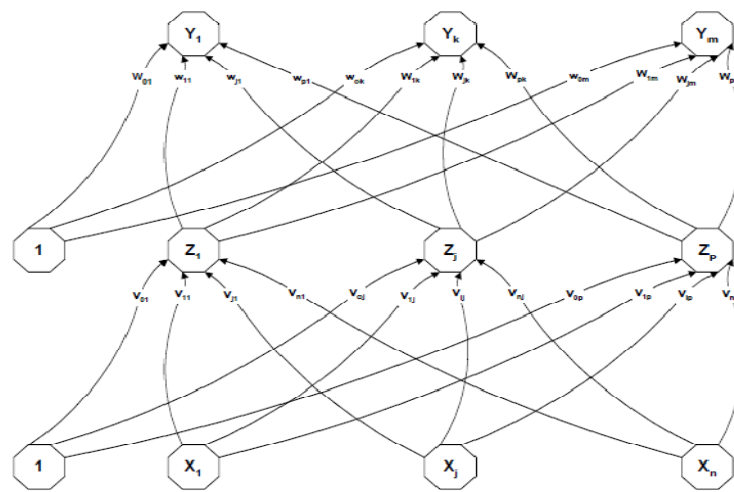
1. Pemrosesan informasi terjadi pada elemen sederhana yang dinamakan neuron.
2. Sinyal antar neuron berhubungan melalui saluran penghubung.
3. Setiap saluran penghubung mempunyai nilai bobot, dan melakukan operasi perkalian dengan sinyal yang ditransmisikan.

4. Setiap neuron memberlakukan fungsi aktivasi (biasanya tidak linier) pada masukan total untuk mendapatkan sinyal keluaran [2].

## 2.2 Metode Pelatihan dan Model Jaringan Saraf Tiruan

### 2.2.1 Jaringan Saraf Tiruan *Backpropagation*

Struktur jaringan saraf tiruan yang digunakan terdiri dari tiga lapisan, yaitu lapisan masukan (input layer), lapisan tersembunyi (hidden layer), dan lapisan keluaran (output layer), seperti terlihat pada gambar 2.3.



Gambar 2.3 Struktur Jaringan Saraf Tiruan *Backpropagation* [2]

Lapisan masukan terdiri dari nilai-nilai matriks yang membentuk pola, sedangkan lapisan keluaran terdiri dari banyaknya kelas yang ingin diklasifikasikan dari semua pola yang ada, yang secara tak langsung terhubung pada apa karakter dibalik pola. Jumlah neuron pada lapisan tersembunyi, bobot-bobot yang menyatakan hubungan lapisan masukan dengan lapisan tersembunyi, dan bobot-bobot lapisan tersembunyi dengan lapisan keluaran akan divariasikan pada saat latihan.

Algoritma pelatihan *backpropagation* merupakan pelatihan yang paling sering digunakan pada jaringan saraf tiruan. Pelatihan *backpropagation* ini terdiri dari tiga tahap, yaitu tahap propagasi maju (*Feedforward*), propagasi balik (backward), dan perubahan bobot. Algoritma *backpropagation*

menggunakan *Error* keluaran untuk mengubah nilai bobot-bobotnya dalam arah balik (*backward*). Untuk mendapatkan *Error* ini, tahap propagasi maju harus dikerjakan terlebih dahulu.

Berdasarkan arsitektur jaringan pada gambar 2.3, selama proses propagasi maju, neuron masukan  $x_i$  ( $i=1,2,\dots,n$ ) menerima sinyal masukan dan mendistribusikannya ke tiap neuron pada lapisan tersembunyi. Pada lapisan tersembunyi, setiap neuron tersembunyi  $z_j$  ( $j=1,2,\dots,p$ ) akan menjumlahkan sinyal terbobot yang masuk dan menghitung sinyal aktivasi  $z_j$  yang dihasilkan. Sinyal aktivasi ini kemudian dikirim sebagai masukan ke tiap neuron pada lapisan berikutnya, entah itu lapisan tersembunyi yang kedua (atau lapisan tersembunyi seterusnya) ataupun lapisan keluaran. Jika dikirim ke lapisan keluaran, maka setiap neuronnya  $Y_k$  ( $k=1,2,\dots,m$ ) akan menjumlahkan sinyal terbobot yang masuk dan menghitung sinyal aktivasi  $Y_k$  yang dihasilkan. Sinyal aktivasi pada lapisan keluaran ini adalah sinyal respon dari jaringan saraf tiruan.

Secara teoritis, jaringan dengan sebuah lapisan tersembunyi sudah cukup bagi *backpropagation* untuk mengenali pola antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah lapisan tersembunyi kadangkala membuat pelatihan lebih mudah [3].

Selama proses pelatihan, tiap keluaran yang dihasilkan oleh neuron keluaran akan dibandingkan dengan target atau respon yang dikehendaki untuk menentukan *Error* yang akan dihasilkan. *Error* yang dihasilkan ini kemudian didistribusikan kembali ke lapisan-lapisan sebelumnya dan digunakan untuk memodifikasi bobot-bobot yang ada. Proses pelatihan melakukan perhitungan secara berulang-ulang sampai dihasilkan respon atau keluaran jaringan saraf tiruan yang sama dengan atau mendekati target dalam batas toleransi yang telah ditentukan atau jumlah iterasi pada proses ini melebihi jumlah iterasi yang diperkenankan.

Algoritma *backpropagation* dapat ditulis sebagai berikut:

Langkah 0:

- Pemberian inisialisasi bobot (diberi nilai kecil secara acak)

Langkah 1 :

- ulangi langkah 2 hingga 9 sampai kondisi akhir iterasi dipenuhi

Langkah 2:

- Untuk masing-masing pasangan data pelatihan lakukan langkah 3 hingga 8

Propagasi maju (*Feedforward*)

Langkah 3:

- Masing-masing unit masukan ( $X_i$ ,  $i = 1, \dots, n$ ) menerima sinyal masukan  $X_i$  dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit-unit lapisan tersembunyi)

Langkah 4:

- Masing-masing unit dilapisan tersembunyi dikalikan dengan factor bobot dan dijumlahkan serta ditambah dengan biasnya:

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.1)$$

Kemudian menghitung sesuai dengan fungsi aktivasi yang digunakan:

$$z_j = f(z_{in_j}) \quad (2.2)$$

bila yang digunakan adalah fungsi sigmoid maka bentuk fungsi tersebut adalah

$$z_j = \frac{1}{1 + e^{-z_{in_j}}} \quad (2.3)$$

Kemudian mengirim sinyal tersebut ke semua unit keluaran (unit keluaran ).

Langkah 5:

- Masing-masing unit keluaran ( $y_k$ ,  $k=1,2,3\dots m$ ) dikalikan dengan faktor bobot dan dijumlahkan:

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.4)$$

Menghitung kembali sesuai dengan fungsi aktivasi

$$y_k = f(y_{in_k}) \quad (2.5)$$

BackPropagasi dan Galatnya

Langkah 6:

- Masing-masing unit keluaran ( $y_k$ ,  $k=1, \dots, m$ ) menerima pola target ( $t_k$ ) sesuai dengan pola masukan dan dihitung galatnya:

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.6)$$

karena  $f'(y_{in_k}) = y_k(1 - y_k)$  menggunakan fungsi sigmoid, maka :

$$\begin{aligned} f'(y_{in_k}) &= f(y_{in_k})(1 - f(y_{in_k})) \\ &= y_k(1 - y_k) \end{aligned} \quad (2.7)$$

Menghitung perbaikan faktor bobot (kemudian untuk memperbaiki wjk).

$$\Delta w_{kj} = \alpha \cdot \delta_k \cdot z_j \quad (2.8)$$

Menghitung perbaikan  $w_{0k}$  koreksi:

$$\Delta w_{0k} = \alpha \cdot \delta_k \quad (2.9)$$

dan menggunakan nilai  $\delta_k$  pada semua unit lapisan sebelumnya.

Langkah 7:

- Masing-masing bobot yang menghubungkan unit-unit lapisan keluaran dengan unit-unit pada lapisan tersembunyi ( $Z_j$ ,  $j=1, \dots, p$ ) dikalikan delta dan dijumlahkan sebagai masukan ke unit-unit lapisan berikutnya.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.10)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktifasinya menghitung galat.

$$\delta_j = \delta_{in_j} \cdot f'(y_{in_j}) \quad (2.11)$$

Kemudian menghitung perbaikan bobot (digunakan untuk memperbaiki  $V_{ij}$ ).

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (2.13)$$

Kemudian menghitung perbaikan bias (untuk memperbaiki  $v_{0j}$ ),

$$\Delta v_{0j} = \alpha \cdot \delta_j \quad (2.14)$$

Memperbaiki bobot dan bias

Langkah 8:

- Masing-masing keluaran unit ( $y_k$ ,  $k=1,\dots,m$ ) diperbaiki bias dan bobotnya ( $j=0,\dots,p$ ),

$$w_{baru_{jk}} = w_{lama_{jk}} + \Delta w_{kj} \quad (2.15)$$

masing-masing unit tersembunyi ( $Z_j$ ,  $j: 1,\dots,p$ ) diperbaiki bias dan bobotnya ( $j=0,\dots,n$ ).

$$v_{baru_{ij}} = v_{lama_{ij}} + \Delta v_{ij} \quad (2.16)$$

Langkah 9:

- Jika jumlah iterasi belum terpenuhi atau Mean Square Error (MSE) belum tercapai, kembali ke langkah 2.

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu dalam standar *backpropagation*, bobot dan bias diisi dengan bilangan acak kecil [4].

Nguyen dan Widrow mengusulkan cara membuat inisialisasi bobot dan bias ke neuron tersembunyi sehingga menghasilkan iterasi lebih cepat. Misal :

$n$  = jumlah neuront masukan

$p$  = jumlah neuron tersembunyi

$$\beta = \text{faktor skala} = 0.7 \sqrt[n]{p} \quad (2.17)$$

Algoritma inisialisasi Nguyen-Widrow adalah sebagai berikut:

1. Inisialisasi semua bobot ( $v_{ij}(\text{lama})$ ) dengan bilangan acak dalam interval  $[-0.5, 0.5]$ .

$$2. \text{ Hitung } \|v_j\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2} \quad (2.18)$$

3. Bobot yang dipakai sebagai inisialisasi,  $v_{ij} = \frac{\beta v_{ij}(\text{lama})}{\|v_j\|}$  (2.19)
4. Bias yang dipakai sebagai inisialisasi,  $v_{0j} =$  bilangan acak antara  $-\beta$  dan  $\beta$

Modifikasi bobot dapat dilakukan dengan penambahan momentum. Jika ada penambahan momentum, bobot baru pada waktu ke-(t+1) didasarkan bobot pada waktu t dan (t-1). Jika  $\mu$  adalah konstanta ( $0 < \mu < 1$ ) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \cdot \delta_k \cdot z_j + \mu(w_{jk}(t) - w_{jk}(t-1)) \quad (2.20)$$

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \cdot \delta_j \cdot x_i + \mu(v_{ij}(t) - v_{ij}(t-1)) \quad (2.21)$$

### 2.2.2. Jaringan Saraf Tiruan Self-Organizing Maps (SOM)

Teknik self-organizing maps (SOM) dikenalkan pertama kali oleh Teuvo Kohonen, merupakan proses unsupervised learning yang mempelajari distribusi himpunan pola-pola tanpa informasi kelas.

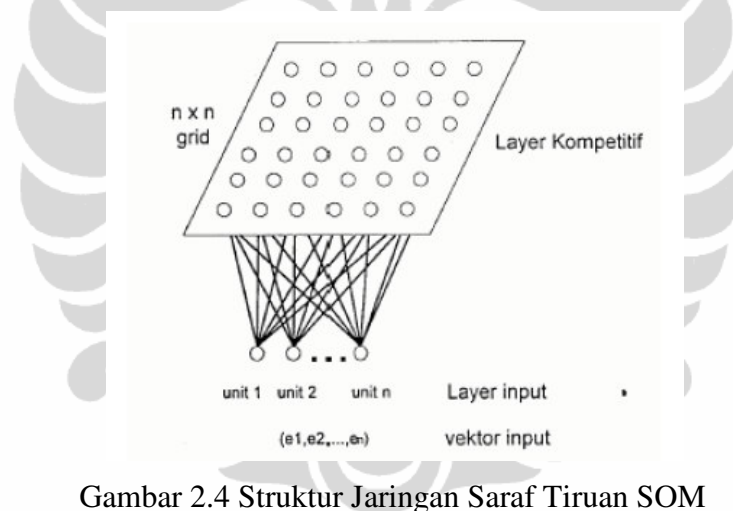
Ide dasar teknik ini diilhami dari bagaimana proses otak manusia menyimpan gambar/pola yang telah dikenalnya melalui mata, kemudian mampu mengungkapkan kembali gambar/pola tersebut. Pada mata kita proses tersebut adalah realisasi pemetaan mapping dari retina menuju cortex. oleh karenanya aplikasi model JST ini banyak digunakan pada pengenalan obyek/citra visual (visual image).

Proses pemetaan terjadi bila sebuah pola berdimensi bebas diproyeksikan dari ruang masukan ke posisi pada array berdimensi satu atau dua. Metoda ekstraksi informasi tersebut dapat dinyatakan sebagai observasi terhadap sebuah pola melalui jendela yang terbentuk oleh lokasi unit-unit luasan pola. pola yang dikenali hanya pola yang batasan unit lokasinya jelas berbeda, biasanya observasi hanya dapat dilakukan bila lokasi pola tersebut mendapat iluminasi/pencahayaan yang cukup/normal.

Meskipun SOM adalah proses klasifikasi, namun tidak seperti teknik klasifikasi atau pengelompokan yang umum digunakan, yang hanya

menyediakan penataan kelas-kelas berdasarkan topologinya. Kemiripan pada pola masukan dipertahankan agar tidak berubah sampai pada keluaran proses. Topologi untuk mempertahankan pola kemiripan pada proses SOM membuatnya berguna sekali, khususnya pada klasifikasi data yang memiliki jumlah kelas yang besar. Pada klasifikasi sampel subcitra, sebagai contoh, mungkin ada sejumlah besar kelas yang perubahannya dari satu kelas ke kelas selanjutnya tidak begitu jauh (membuatnya sulit untuk mendefinisikan batas kelas yang jelas).

Pada jaringan saraf tiruan ini lapisan masukan (pertama) terhubung secara penuh dengan lapisan kompetitif (kedua) seperti gambar 2.4. Jadi setiap unit masukan terhubung ke semua unit keluaran dan pada hubungan tersebut terdapat nilai bobot (weight) tertentu.



Gambar 2.4 Struktur Jaringan Saraf Tiruan SOM

Algoritma Self-Organizing maps dapat ditulis sebagai berikut:

- Langkah 1 : Jika kondisi henti gagal, lakukan langkah 2-7
- Langkah 2 : Untuk setiap vektor masukan  $x$ , lakukan langkah 3 sampai 5
- Langkah 3 : Untuk setiap  $j$ , hitung:

$$D_i = \sqrt{\sum_{j=1}^m (V_{ji} - X_i)^2} \quad (2.22)$$

- Langkah 4 : Temukan indeks  $j$  sehingga  $D(j)$  minimum



- Langkah 5 : Untuk setiap neuron J
  - ✓ meng-update bobotnya

$$v_{ji}(\text{new}) = v_{ji}(\text{lama}) + \alpha (x_i - v_{ji}(\text{lama})) \quad (2.23)$$

- Langkah 6 : Memodifikasi laju pemahaman
- Langkah 7 : melanjutkan iterasi.

### 2.2.3 Jaringan saraf Tiruan Fungsi Basis Radial (Radial Basis Function Networks)

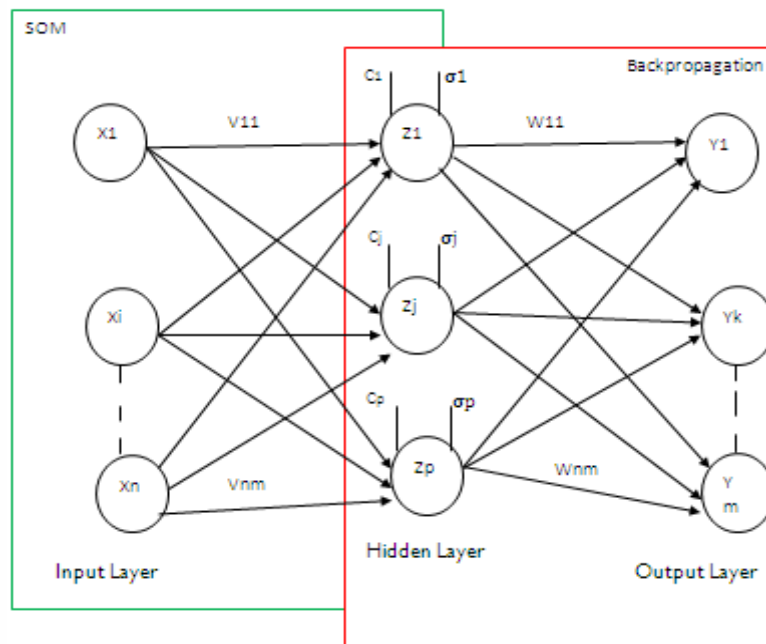
Semakin tinggi tingkat ketidaklinieran suatu sistem, sering kali sulit di selesaikan dengan kedua metoda yang telah diuraikan sebelumnya yaitu metoda pelatihan terbimbing dan tak terbimbing. Untuk mengatasi problema tersebut, banyak periset yang mencoba menggabungkannya dan diperoleh hasil yang lebih baik. Salah satu metoda hibrida yang berhasil dan banyak digunakan adalah metoda radial basis function network (RBFN).

Model jaringan ini, neuron-neuron keluarannya adalah hasil kombinasi linier fungsi basis neuron-neuron pada lapisan tersembunyi. Sebagai fungsi basis yang umum digunakan adalah Gaussian. Perbedaan utama antara jaringan multi lapis perceptron (MLP) dengan jaringan berbasis fungsi radial yaitu penggunaan Gaussian pada lapisan tersembunyi jaringan RBF, sedangkan jaringan MLP menggunakan fungsi sigmoid.

Pada prinsipnya RBF adalah emulasi sifat jaringan biologi yang umumnya sel/neuron yang paling aktif adalah sel/neuron yang paling sensitip menerima rangsangan sinyal masukan. Sehingga orientasi sensitivitas respon tersebut hanya terhadap beberapa daerah (local response) dalam wilayah masukan. JST dengan lapisan tersembunyi tunggal, pada dasarnya lapisan tersebut berisi neuron-neuron (unit-unit) yang sensitip atau aktif secara lokal. Sedangkan keluarannya terdiri dari unit-unit linear.

Pada neuron-neuron dalam lapisan tersembunyi, respon euronnya bersifat lokal dan berkurang sebagai fungsi jarak masukan dari pusat unit penerima rangsangan. Metoda ini menjadi terkenal sejak Broomhead dan Lowe's pada

tahun 1988 menyampaikan makalahnya yang berjudul "Multivariate functional interpolation and adaptive network ". JST-RBF mempunyai kesamaan dasar dengan JST-MLP yang struktur dasarnya ditunjukkan pada gambar 2.5.



Gambar 2.5 Struktur Jaringan Saraf Tiruan FBR

### Fungsi Radial

Fungsi Radial adalah suatu fungsi yang mempunyai karakteristik merespon pengurangan ataupun penambahan secara monoton dengan jarak yang berasal dari nilai tengahnya. Jenis fungsi radial yang banyak digunakan adalah fungsi Gaussian seperti pada persamaan berikut:

$$\varphi_j = \exp\left(-\frac{(x_i - c_j)^T(x_i - c_j)}{2(\sigma_j)^2}\right) \quad (2.24)$$

Dimana:  $\varphi$  adalah jenis fungsi aktivasi yang digunakan dalam RBF

$c$  adalah pusat (nilai tengah).

$(x_i - c_j)^T(x_i - c_j)$  adalah jarak antara input  $x$  dan pusat

$\sigma_j$  adalah spread (penyebaran data)

Beberapa tipe fungsi aktivasi RBF adalah sebagai berikut:

- Fungsi Thinplate-Splane

$$\phi(z, 1) = z^2 \log(z) \quad (2.25)$$

- Fungsi Multiquadratic

$$\phi(z, \sigma) = (z^2 + \sigma^2)^{1/2} \quad (2.26)$$

- Fungsi Inverse Multiquadratic

$$\phi(z, \sigma) = \frac{1}{(z^2 + \sigma^2)^{1/2}} \quad (2.27)$$

- Fungsi Gauss

$$\phi(z, \sigma) = \exp(-z^2/\sigma^2) \quad (2.28)$$

Hasil penelitian menyatakan bahwa seleksi dari keempat fungsi nonlinier tersebut tidak dominan menentukan kinerja RBF. Bila jarak Euclidian antara vector masukan dan unit-unit dalam lapis tersembunyi mempunyai nilai yang berbeda, maka jarak yang sama untuk setiap unitnya cukup untuk pendekatan secara universal. Ini berarti bahwa semua jarak dapat disesuaikan pada sebuah nilai  $\sigma$  untuk menyederhanakan strategi pelatihannya.

### 2.3 Fungsi Keanggotaan

Fungsi keanggotaan (membership function) adalah suatu kurva yang menunjukkan pemetaan vektor data ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan, yaitu:

#### 1. Representasi Linier

Pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

#### 2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier).

3. Representasi Kurva Trapesium

Kurva segitiga pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1

4. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun.

5. Representasi Kurva-S

Kurva pertumbuhan dan penyusutan merupakan kurva-S atau sigmoid yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linier.

6. Representasi Kurva Bentuk Lonceng

Untuk mempresentasikan bilangan fuzzy, biasanya digunakan kurva berbentuk lonceng. Kurva ini terbagi atas 3 kelas, yaitu: himpunan fuzzy PI, beta, dan Gauss. Perbedaan ketiga kurva ini terletak pada gradiennya.

### BAB 3

## ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS RADIAL MENGGUNAKAN ALGORITMA SOM

Pada bab ini penulis akan memaparkan skema penelitian jaringan saraf fungsi basis radial (FBR), parameter-parameter yang digunakan, bagaimana jaringan saraf FBR dengan menggunakan SOM dapat digunakan untuk sistem pengenalan pola, dan hasil pengujian.

### 3.1. Skema Penelitian Jaringan Saraf FBR Menggunakan SOM

Sistem pengenalan pola iris yang dikerjakan pada penelitian ini terbagi menjadi 3 tahapan, yang dijabarkan seperti gambar 3.1.



Gambar 3.1. Skema penelitian

#### 3.1.1. Data Percobaan

Uji coba dilakukan pada komputer dengan prosesor Pentium 1.66 Ghz dan Memory 2 GB. Perangkat lunak yang digunakan dalam percobaan ini adalah bahasa pemrograman MATLAB dan dijalankan pada sebuah komputer. Percobaan dilakukan terhadap data Odor 2 campuran yang terdiri dari 3 aroma (odor) alami yang diuji yaitu kenanga, jeruk dan mawar.

Masing-masing aroma dicampur dengan alkohol dengan 6 persentase yang berbeda sehingga secara keseluruhan didapatkan 18 jenis odor, di mana masing-masing jenis odor diambil 200 data eksperimen.

- a. Eksperimen dengan 16 dimensi (yang mewakili 16 sensor odor yang digunakan)
- b. Eksperimen dengan 8 dimensi (yang mewakili 8 sensor odor yang digunakan)
- c. Eksperimen dengan 4 dimensi (yang mewakili 4 sensor odor yang digunakan)

### 3.1.2. Pelatihan atau Training

Setelah data percobaan didapatkan, tahap selanjutnya yang merupakan bagian utama sistem pengenalan pola adalah pelatihan pola atau training. Sistem dilatih untuk mengenali pola masukan agar menghasilkan keluaran yang sesuai dengan target tertentu. Awal pelatihan digunakan pelatihan tanpa arahan untuk melakukan pengelompokan data, kemudian setelah itu digunakan pelatihan dengan arahan untuk menggeneralisasi nilai masukan. Jaringan ini hanya mempunyai satu lapisan tersembunyi. Nilai bobot, center, dan spread akan dimodifikasi terus menerus sampai sistem ini menemukan nilai akhir yang konvergen. Dalam penelitian ini, satu kali pelatihan dinyatakan sebagai satu epoch

### 3.1.3. Pengujian atau Testing

Tahap pengujian ini berupa verifikasi suatu pola akan ditempatkan dalam kelas yang mana. Pada penelitian, tahap pengujian ini dilakukan dengan menggunakan pola yang serupa namun tidak sama dengan pola yang digunakan pada pelatihan.

Dalam proses pengujian, setiap pola dijalankan ke dalam jaringan *backpropagation* tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang disimpan sebelumnya.

Untuk satu kali pengujian, persentase kesesuaian terhadap target yang disebut recognition rate pun dihitung.

## 3.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Menggunakan SOM

Parameter yang digunakan dalam penelitian jaringan saraf FBR menggunakan SOM ini dan sekaligus menjadi variable bebas, antara lain :

### 1. Inisialisasi bobot

Dalam penelitian ini ada dua macam inisialisasi bobot, yaitu inisialisasi bobot pada jaringan saraf swa organisasi dengan memilih salah satu dari

vektor data dan inisialisasi bobot pada jaringan saraf propagasi balik yang ditentukan secara random dan metode Nguyen-Widrow

2. Banyaknya neuron yang digunakan pada lapisan masukan, lapisan tersembunyi, dan lapisan keluaran disesuaikan dengan pola yang digunakan.
3. Besarnya epoch  
Banyaknya epoch yang digunakan adalah 1000 epoch.
4. Besarnya laju pembelajaran ( $\alpha$ )  
Laju pembelajaran untuk algoritma SOM sebesar 0.1 dan algoritma propagasi balik sebesar 0.2, laju pembelajaran untuk pembaharuan *center* sebesar 0.5 dan pembaharuan *spread* sebesar 0.8. Sementara itu, variable terikat yang digunakan pada penelitian ini adalah nilai recognition rate testing.
5. Nilai kesalahan Uji  
Range nilai kesalahan uji yang digunakan dalam penelitian ini adalah 0.49 untuk nilai 0 dan range 0.51 untuk nilai 1

### 3.3. Jaringan Saraf FBR Menggunakan SOM

Seperti yang telah dijelaskan Bab II, jaringan saraf tiruan adalah jaringan yang prinsip kerjanya menyerupai jaringan saraf manusia, di mana jaringan ini dapat mempelajari suatu pola yang diberikan dengan target tertentu. Dari pola-pola tersebut, proses pembelajaran dilakukan dengan bobot-bobot atau koefisien-koefisien di antara neuron-neuronnya secara berulang-ulang sehingga pada akhirnya akan mendapatkan suatu tetapan yang *default*, yang dapat digunakan untuk mengidentifikasi suatu pola baru.

Jaringan saraf tiruan yang digunakan pada penelitian ini terdiri dari tiga lapisan yaitu : lapisan masukan  $X_i$  (*input layer*), lapisan tersembunyi  $Z_j$  (*hidden layer*), dan lapisan keluaran  $Y_k$  (*output layer*). Lapisan masukan  $X_i$  ( $i = 1, 2, \dots, n$ ) berisi nilai-nilai atau atribut-atribut atau informasi suatu pola

yang digunakan pada pelatihan. Oleh karena itu, jumlah neuron pada lapisan ini sama dengan jumlah atribut yang akan dikenali.

Dari gambar 2.5, dapat dilihat bahwa setiap neuron pada lapisan masukan terhubung dengan setiap neuron pada lapisan tersembunyi, dan setiap neuron pada lapisan tersembunyi terhubung dengan setiap neuron pada lapisan keluaran. Karena lapisan keluaran merupakan lapisan paling kanan, maka neuron-neuronnya tidak terhubung dengan neuron apapun lagi selain neuron-neuron pada lapisan di sebelah kirinya. Setiap hubungan antar neuron ini berisi suatu nilai atau konstanta yang disebut bobot. Bobot-bobot yang menghubungkan lapisan masukan dengan lapisan tersembunyi dinotasikan dalam  $V_{ij}$  sedangkan bobot-bobot yang menghubungkan lapisan tersembunyi dengan lapisan keluaran dinotasikan dengan  $W_{jk}$ . Pada awal-awal, bobot-bobot tersebut diberi nilai acak kemudian untuk algoritma propagasi balik dilanjutkan dengan metode Nguyen-Widrow sedangkan untuk algoritma SOM tidak menggunakan metode Nguyen-Widrow.

Jaringan saraf tiruan yang digunakan pada penelitian ini adalah jaringan yang hanya memiliki sebuah lapisan tersembunyi, karena jaringan saraf fungsi basis radial hanya memiliki sebuah lapisan tersembunyi. Pada penelitian ini jaringan saraf tiruannya merupakan gabungan dua algoritma, yaitu algoritma SOM dan algoritma Propagasi balik.

Proses pelatihan pada jaringan saraf SOM merupakan proses pengelompokan (*clustering*) vektor data yang mana setiap kelompok atau *cluster* berisi vektor-vektor data yang memiliki kemiripan yang paling dekat dengan vektor masukan. Ukuran kemiripan yang digunakan dalam penelitian ini adalah jarak Euclidean yang paling minimum. Banyaknya kelompok atau *cluster* menentukan banyaknya neuron yang digunakan dalam lapisan tersembunyi. Setelah proses pelatihan secara berulang-ulang selesai dan memiliki bobot akhir, maka dilakukan proses pengelompokan vektor yang gunanya untuk mencari nilai awal *center* dan *spread*, caranya setiap kelompok dihitung nilai rata-ratanya dan setelah itu dihitung nilai *spread* (penyebaran) dengan cara mencari jarak maksimum antara dua



kelompok atau *cluster* dibagi dengan akar dari banyaknya kelompok atau *cluster*.

Pada proses selanjutnya yaitu proses pelatihan propagasi balik, setiap vektor data melakukan proses propagasi maju, propagasi balik dan perubahan bobot, *center* dan *spread*. Pada proses propagasi maju dihitung nilai  $Z_j$  ( $j = 1, 2, \dots, p$ ) berdasarkan nilai *center* dengan menggunakan fungsi aktivasi Gaussian. Hasil penghitungan proses Gaussian akan dikomputasi dengan bobot-bobot yang berkesesuaian dengannya ( $W_{jk}$ ) sehingga menghasilkan nilai pada neuron-neuron lapisan keluaran  $Y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan fungsi aktivasi Sigmoid Biner. Proses propagasi balik adalah proses yang dilakukan setelah proses propagasi maju dalam bagian ini dihitung informasi *Error* nya ( $\delta_k$ ) yang nilainya akan digunakan dalam menghitung besarnya koreksi bobot neuron keluaran dan besarnya koreksi bias keluaran. Setelah proses propagasi maju dan propagasi balik selesai maka dihitung nilai perubahan bobot, *center* yang kedua dan *spread* berdasarkan informasi *Error*nya. Setelah proses pelatihan kemudian dilakukan proses pengujian untuk pola yang lama atau pola yang baru. Proses yang dilakukan untuk pengenalan pola ini hanyalah pengelompokan data dan propagasi maju.

Pada penelitian ini, untuk memudahkan perhitungan pada program, nilai sebuah target dinyatakan dalam matriks yang ukurannya disesuaikan dengan neuron keluaran.

Agar jaringan dapat mempelajari pola dengan baik, pola yang mempunyai target yang sama sebaiknya tidak dilatih secara berurutan, melainkan silih berganti dengan pola yang mempunyai target berbeda, misalnya pola dengan target ke-1 dilanjutkan dengan pola target ke-2, dan seterusnya. Setelah itu, barulah pelatihan dilanjutkan dengan pola dengan target ke-1 kembali hingga seluruh pola selesai dilatih.

Pada pelatihan ini algoritma secara detail yang digunakan sebagai berikut:

#### **Langkah 0 : Inisialisasi**

- SOM
  - Bobot awal  $V_{ji}$  di ambil dr salah satu vektor dalam setiap kelas
  - Set parameter laju pembelajaran (Learning Rate ( $\alpha$ ))
- *Backpropagation*
  - Bobot awal  $w_{kj}$  menggunakan nguyen widrow.
  - Bias awal  $w_{0k}$
  - Set parameter laju pembelajaran (Learning rate)

**Proses SOM:**

**Langkah 1 :** Clustering Data, menggunakan SOM

- Langkah a : Jika kondisi henti gagal, lakukan langkah b-g
- Langkah b : Untuk setiap vektor masukan  $x$ , lakukan langkah c sampai e
- Langkah c : Untuk setiap  $j$ , hitung:

$$D_i = \sqrt{\sum_{j=1}^m (V_{ji} - X_i)^2} \quad (3.1)$$

- Langkah d : Temukan indeks  $j$  sehingga  $D(j)$  minimum
- Langkah e : Untuk setiap neuron  $J$ 
  - ✓ meng-update bobotnya

$$v_{ji} (new) = v_{ji} (lama) + \alpha (x_i - v_{ji} (lama)) \quad (3.2)$$

- Langkah f : Memodifikasi laju pemahaman
- Langkah g : melanjutkan iterasi

**Langkah 2 :** Inisialisasi center dan Spread

- Langkah a : menghitung jumlah/ banyaknya vektor data dalam kelas
- Langkah b : menghitung rata-rata vektor /center (c)

$$c_j = \frac{\sum_{i=1}^p x_{pj}}{N_j} \quad (3.3)$$

- Langkah c : menghitung nilai spread ( $\sigma$ )

$$\sigma = \frac{\text{jarak maksimum antara 2 center}}{\sqrt{\text{banyaknya center}}} = \frac{d_{maks}}{\sqrt{k}} \quad (3.4)$$

**Proses Backpropagation:****Langkah 3 :** Untuk masing masing vektor kelas j**Feedforward :**

- Langkah a : menghitung nilai  $z_j$

$$z_j = \exp\left(-\frac{(x_i - c_j)^T(x_i - c_j)}{2(\sigma_j)^2}\right) = \exp(-z_{in}) \quad (3.5)$$

- Langkah b : menghitung nilai  $y_{ink}$

$$y_{ink} = w_0(k) + \sum z_j \cdot w_{kj} \quad (3.6)$$

- Langkah c : menghitung nilai  $Y_k$

$$Y_k = f(-y_{ink})$$

$$Y_k = 1/(1 + \exp(-y_{ink})) \quad (3.7)$$

**Backward :**

- Langkah d : menghitung nilai  $\delta_k$

$$\delta_k = (T_k - Y_k) f'(y_{ink})$$

$$\delta_k = (T_k - Y_k) (Y_k)(1 - Y_k) \quad (3.8)$$

- Langkah e : Menghitung besarnya koreksi bobot unit output

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (3.9)$$

- Langkah f : Menghitung besarnya koreksi bias output

$$\Delta w_0_k = \alpha \delta_k \quad (3.10)$$

**Perubahan Bobot**

- Langkah g : Setiap unit output mengup-date bobotnya

$$w_{kj}(\text{new}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (3.11)$$

- Langkah h : Setiap unit output mengup-date biasnya

$$w_0_k(\text{new}) = w_0_k(\text{lama}) + \Delta w_0_k \quad (3.12)$$

**Perubahan center**

- Langkah i : mengup-date center

$$c_j(\text{new}) = c_j(\text{lama}) + \alpha \left( \frac{do(k) \cdot w_{kj}}{\sigma_j^2} \right) \cdot z_j \cdot [X_i - c_j] \quad (3.13)$$

**Perubahan nilai spread**

- Langkah j : mengup-date nilai spread

$$\sigma_j(\text{new}) = \sigma_j(\text{lama}) + \alpha \left( \frac{do(k) \cdot w_{kj}}{\sigma_j^3} \right) \cdot z_j \cdot ((x_i - c_j)^T(x_i - c_j)) \quad (3.14)$$

- Langkah k : kondisi stopping

$$E_{\min} < 0.01 \quad (3.15)$$

Setelah melakukan pelatihan, dilakukan pengenalan pola dengan, agar seragam diambil pola dengan jumlah yang sama untuk setiap target. Pada proses pengenalan pola ini, sistem akan memberikan jawaban terhadap setiap pola, masuk ke *cluster* yang mana dari *cluster* yang ada.

Apabila jawaban yang diberikan sesuai target, maka jaringan dianggap telah mengenali pola dengan benar. Namun bila jawaban yang diberikan tidak sesuai dengan target, maka jaringan dinilai salah dalam mengenali pola. Sistem tidak dapat menghasilkan jawaban bahwa suatu pola berada di luar target yang ditentukan. Dengan demikian, hanya ada dua pilihan, yaitu benar dan salah. Persentase jawaban yang benar terhadap seluruh pertanyaan mengacu kepada *recognition rate*.

### 3.4. Pengujian pada Jaringan Saraf Tiruan FBR Menggunakan SOM

Pada metode FBR menggunakan SOM, pengujian dilakukan dengan menggunakan tiga jenis data dengan :

1. Rasio perbandingan data training dan testing 70:30
2. Data yang diuji yaitu data original dan data normalisasi

Adapun data yang dilihat sebagai hasil pengujian adalah tingkat kemampuan pengenalan pola (Recognition Rate) dan waktu yang diperlukan untuk mengenali pola tersebut.

Perbandingan data yang digunakan 70 : 30 berarti 70% data dijadikan sebagai data pembelajaran jaringan dan 30% data sisanya digunakan sebagai data pengujian.

#### 3.4.1. Hasil Pengujian dan Analisa

Berikut ini tabel 3.1 menunjukkan hasil ujicoba untuk data original dalam mendeteksi data yang berdimensi 4, 8, dan 16 dan tabel 3.2 menunjukkan hasil ujicoba untuk data normalisasi dalam mendeteksi data yang berdimensi 8, dan 16.

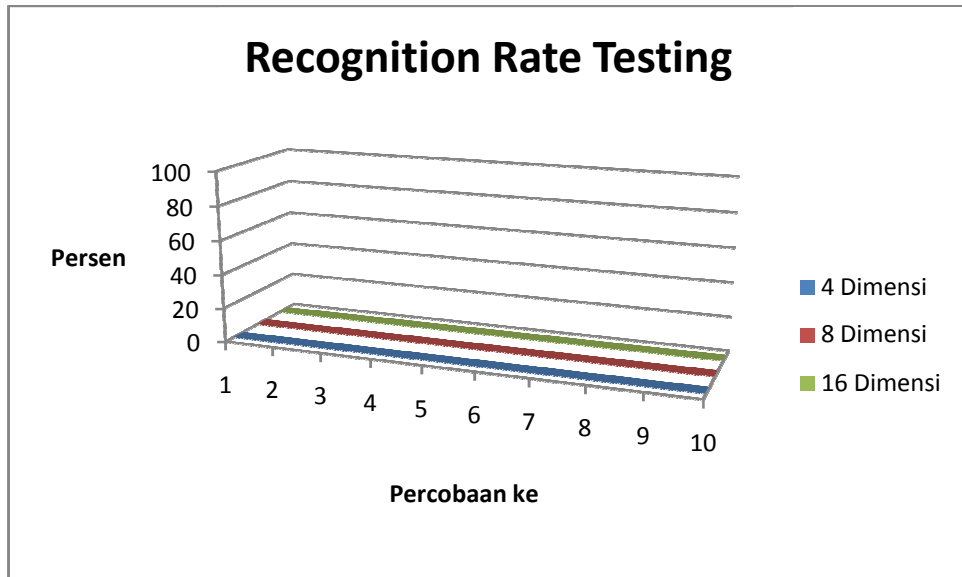
Tabel 3.1 Recognition Rate Testing Data Original untuk 1000 epoch

Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.00	0.00	0.00
10	0.00	0.00	0.00
Rata-rata	0.00	0.00	0.00
Waktu	6339.22	6230.05	6411.73

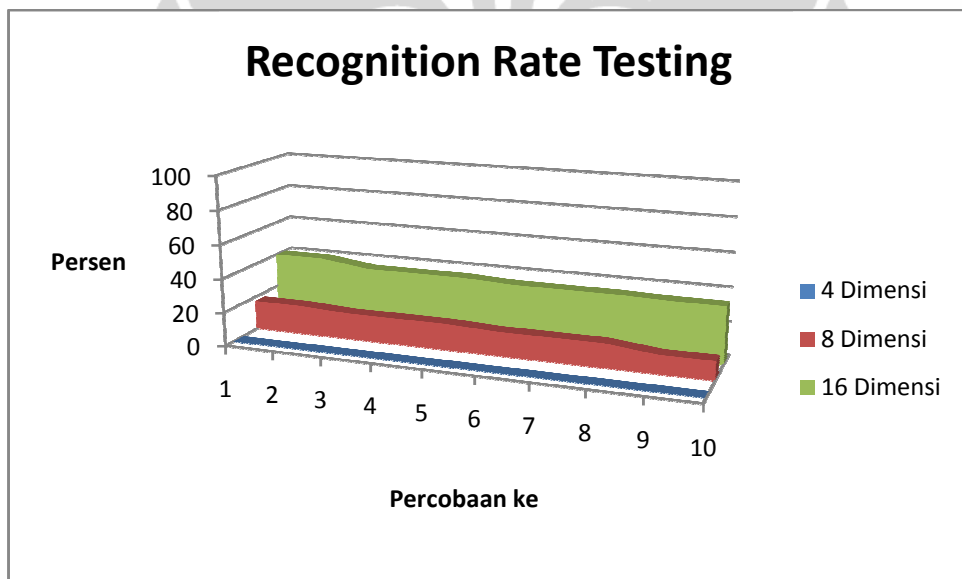
Tabel 3.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch

Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	1.78	17.22	39.89
2	0.00	17.22	39.89
3	0.00	15.67	36
4	0.00	15.67	36
5	0.00	15.67	36
6	0.00	14.33	34.56
7	0.00	14.33	34.56
8	0.00	14.33	34.56
9	0.00	11.11	33.78
10	0.00	11.11	33.78
Rata-rata	0.59	14.67	35.90
Waktu	6800.81	6929.14	6112.1

Untuk mempermudah proses analisa, data direpresentasikan kembali dalam bentuk grafik dapat dilihat seperti di bawah ini



Gambar 3.2 Grafik Recognition Rate Testing Data Original



Gambar 3.3. Recognition Rate Testing Data Normalisasi untuk 1000 epoch

Dari gambar 3.2 dan gambar 3.3 untuk data 4 dimensi, data 8 dimensi dan data 16 dimensi terlihat bahwa dalam metode FBR data normalisasi dapat diklasifikasikan lebih baik dari pada data aslinya, hal ini dikarenakan data hasil normalisasi memiliki tingkat independensi yang lebih baik. Pada

proses FBR metode pengklasifikasi sangat dipengaruhi oleh perubahan bobot, center dan spread.

Pada data normalisasi, komponen dari perubahan bobot, center dan spread memiliki tingkat perbedaan yang lebih jelas jika dibandingkan dengan tingkat perbedaan pada data aslinya. Sehingga proses perubahan tiga parameter tersebut yang dihasilkan akan semakin baik, akibatnya proses klasifikasi juga akan menjadi semakin baik.



## BAB 4

### ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS RADIAL TANPA ALGORITMA SOM.

#### 4.1. Skema Jaringan Saraf FBR Tanpa Menggunakan SOM

Sama seperti pada penelitian jaringan Saraf tiruan FBR menggunakan SOM, jaringan saraf FBR tanpa menggunakan SOM juga terdiri dari tiga tahapan, yaitu: data percobaan, pelatihan atau training, dan pengujian atau testing.

##### 4.1.1. Data Percobaan

Data percobaan yang digunakan dalam penelitian ini adalah sama dengan data percobaan yang digunakan pada jaringan Saraf tiruan FBR menggunakan SOM.

##### 4.1.2. Pelatihan atau Training

Sistem dilatih untuk mengenali pola masukan agar menghasilkan keluaran yang sesuai dengan target tertentu. Dalam penelitian ini tidak digunakan pelatihan tanpa arahan untuk menentukan kelompok data, namun pengelompokan data dilakukan menurut kelas data yang sudah ada, setelah itu digunakan pelatihan dengan arahan untuk menggeneralisasi nilai masukan. Jaringan ini hanya mempunyai satu lapisan tersembunyi. Nilai bobot, center, dan spread akan dimodifikasi terus menerus sampai sistem ini menemukan nilai akhir yang konvergen. Dalam penelitian ini, satu kali pelatihan dinyatakan sebagai satu epoch

##### 4.1.3. Pengujian atau Testing

Tahap pengujian ini berupa verifikasi suatu pola akan ditempatkan dalam kelas yang mana. Pada penelitian, tahap pengujian ini dilakukan dengan menggunakan pola yang sama namun tidak sama dengan pola yang digunakan pada pelatihan. Dalam proses pengujian, proses pengelompokan data tidak menggunakan algoritma SOM, setelah itu proses dilanjutkan dengan proses *backpropagation*, setiap pola dijalankan ke dalam jaringan



*backpropagation*, tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang disimpan sebelumnya.

Untuk satu kali pengujian, persentase kesesuaian terhadap target yang disebut recognition rate pun dihitung.

#### **4.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Tanpa SOM**

Parameter yang digunakan dalam penelitian jaringan saraf FBR tanpa menggunakan SOM yang terdiri dari inisialisasi bobot, neuron yang digunakan pada lapisan masukan, lapisan tersembunyi dan lapisan keluaran, banyaknya epoch, besarnya laju pembelajaran namun tanpa laju pembelajaran untuk SOM dan nilai kesalahan uji yang digunakan. Nilai parameter yang digunakan pada bab ini sama dengan nilai parameter yang digunakan pada bab tiga.

#### **4.3. Jaringan Saraf FBR Tanpa Algoritma SOM**

Pada penelitian ini jaringan saraf tiruan yang digunakan sama dengan saraf tiruan yang digunakan pada bab ke-3, yaitu jaringan saraf fungsi basis radial, namun proses SOM tidak dilakukan untuk pengelompokan vektor data dalam bab ini. Proses pengelompokan vektor data dilakukan secara manual dengan memasukan vektor data secara langsung ke cluster yang sesuai dengan cluster vektor data, Setiap cluster terdapat beberapa vektor data untuk proses pelatihan, pengelompokan ini digunakan untuk mencari nilai awal *center* dan *spread*, caranya setiap kelompok dihitung nilai rata-ratanya dan setelah itu dihitung nilai *spread* (penyebaran) dengan cara mencari jarak maksimum antara dua kelompok atau *cluster* dibagi dengan akar dari banyaknya kelompok atau *cluster*.

Pada proses selanjutnya yaitu proses pelatihan propagasi balik, setiap vektor data melakukan proses propagasi maju, propagasi balik dan perubahan bobot, *center* dan *spread*. Pada proses propagasi maju dihitung nilai  $Z_j$  ( $j = 1, 2, \dots, p$ ) berdasarkan nilai *center* dengan menggunakan fungsi aktivasi

Gaussian. Hasil penghitungan proses Gaussian akan dikomputasi dengan bobot-bobot yang berkesesuaian dengannya ( $W_{jk}$ ) sehingga menghasilkan nilai pada neuron-neuron lapisan keluaran  $Y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan fungsi aktivasi Sigmoid Biner. Proses propagasi balik adalah proses yang dilakukan setelah proses propagasi maju dalam bagian ini dihitung informasi *Error* nya ( $\delta_k$ ) yang nilainya akan digunakan dalam menghitung besarnya koreksi bobot neuron keluaran dan besarnya koreksi bias keluaran. Setelah proses propagasi maju dan propagasi balik selesai maka dihitung nilai perubahan bobot, *center* dan *spread* berdasarkan informasi *Error*nya. Setelah proses pelatihan kemudian dilakukan proses pengujian untuk pola yang lama atau pola yang baru. Proses yang dilakukan untuk pengenalan pola ini hanyalah propagasi maju.

Pada penelitian ini, untuk memudahkan perhitungan pada program, nilai sebuah target dinyatakan dalam matriks yang ukurannya disesuaikan dengan neuron keluaran.

Agar jaringan dapat mempelajari pola dengan baik, pola yang mempunyai target yang sama sebaiknya tidak dilatih secara berurutan, melainkan silih berganti dengan pola yang mempunyai target berbeda, misalnya pola dengan target ke-1 dilanjutkan dengan pola target ke-2, dan seterusnya. Setelah itu, barulah pelatihan dilanjutkan dengan pola dengan target ke-1 kembali hingga seluruh pola selesai dilatih.

Detail algoritma pelatihan jaringan ini adalah sebagai berikut:

**Langkah 0** : Inisialisasi

- *Backpropagation*
  - ✓ Bobot awal  $w_{kj}$  menggunakan nguyen widrow.
  - ✓ Bias awal  $w_{0k}$
  - ✓ Set parameter laju pembelajaran (Learning rate)

**Langkah 1** : Clustering data dilakukan manual

**Langkah 2** : Inisialisasi center dan Spread

- Langkah a : menghitung jumlah/ banyaknya vector data dalam kelas
- Langkah b : menghitung rata-rata vektor /center (c)

$$c_j = \frac{\sum_{i=1}^p x_{pj}}{N_j} \quad (4.1)$$

- Langkah c : menghitung nilai spread ( $\sigma$ )

$$\sigma = \frac{\text{jarak maksimum antara 2 center}}{\sqrt{\text{banyaknya center}}} = \frac{d_{maks}}{\sqrt{k}} \quad (4.2)$$

### Proses Backpropagation:

**Langkah 3 :** Untuk masing masing vektor kelas j

#### Feedforward :

- Langkah a : menghitung nilai  $z_j$

$$z_j = \exp\left(-\frac{(x_i - c_j)^T (x_i - c_j)}{2(\sigma_j)^2}\right) = \exp(-z_{in}) \quad (4.3)$$

- Langkah b : menghitung nilai  $y_{ink}$

$$y_{ink} = w_0(k) + \sum z_j \cdot w_{kj} \quad (4.4)$$

- Langkah c : menghitung nilai  $Y_k$

$$Y_k = f(-y_{ink})$$

$$Y_k = 1/(1 + \exp(-y_{ink})) \quad (4.5)$$

#### Backward :

- Langkah d : menghitung nilai  $\delta_k$

$$\delta_k = (T_k - Y_k) f'(y_{ink})$$

$$\delta_k = (T_k - Y_k) (Y_k)(1 - Y_k) \quad (4.6)$$

- Langkah e : Menghitung besarnya koreksi bobot unit output

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (4.7)$$

- Langkah f : Menghitung besarnya koreksi bias output

$$\Delta w_0_k = \alpha \delta_k \quad (4.8)$$

#### Perubahan Bobot

- Langkah g : Setiap unit output mengup-date bobotnya

$$w_{kj}(\text{new}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (4.9)$$

- Langkah h : Setiap unit output mengup-date biasnya

$$w_0_k(\text{new}) = w_0_k(\text{lama}) + \Delta w_0_k \quad (4.10)$$

### Perubahan center

- Langkah i : mengup-date center

$$c_j(\text{new}) = c_j(\text{lama}) + \alpha \left( \frac{do(k).w_{kj}}{\sigma_j^2} \right) \cdot z_j \cdot [X_i - c_j] \quad (4.11)$$

### Perubahan nilai spread

- Langkah j : mengup-date nilai spread

$$\sigma_j(\text{new}) = \sigma_j(\text{lama}) + \alpha \left( \frac{do(k).w_{kj}}{\sigma_j^3} \right) \cdot z_j \cdot ((x_i - c_j)^T (x_i - c_j)) \quad (4.12)$$

- Langkah k : kondisi stopping

$$E_{\min} < 0.01 \quad (4.13)$$

Setelah melakukan pelatihan, dilakukan pengenalan pola, agar seragam diambil pola dengan jumlah yang sama untuk setiap target. Pada proses pengenalan pola ini, sistem akan memberi jawaban terhadap setiap pola, masuk ke *cluster* yang mana dari tiga *cluster* yang ada.

Apabila jawaban yang diberikan sesuai target, maka jaringan dianggap telah mengenali pola dengan benar. Namun bila jawaban yang diberikan tidak sesuai dengan target, maka jaringan dinilai salah dalam mengenali pola. Sistem tidak dapat menghasilkan jawaban bahwa suatu pola berada di luar target yang ditentukan. Dengan demikian, hanya ada dua pilihan, yaitu benar dan salah. Persentase jawaban yang benar terhadap seluruh pertanyaan mengacu kepada *recognition rate*.

#### 4.4. Pengujian Pada Jaringan Saraf Tiruan FBR tanpa menggunakan SOM

Pada metode FBR tanpa menggunakan SOM, pengujian dilakukan dengan menggunakan tiga jenis data dengan :

1. Rasio perbandingan data training dan testing 70:30
2. Data yang diuji yaitu data original dan data normalisasi

Adapun data yang dilihat sebagai hasil pengujian adalah tingkat kemampuan pengenalan pola (Recognition Rate) dan waktu yang diperlukan untuk mengenali pola tersebut.

Perbandingan data yang digunakan 70 : 30 berarti 70% data dijadikan sebagai data pembelajaran jaringan dan 30% data sisanya digunakan sebagai data pengujian.

#### 4.4.1. Hasil Pengujian dan Analisa

Berikut ini tabel 4.1 menunjukkan hasil ujicoba untuk data original dalam mendeteksi data yang berdimensi 4, 8, dan 16 dan tabel 4.2 menunjukkan hasil ujicoba untuk data normalisasi dalam mendeteksi data yang berdimensi 8, dan 16.

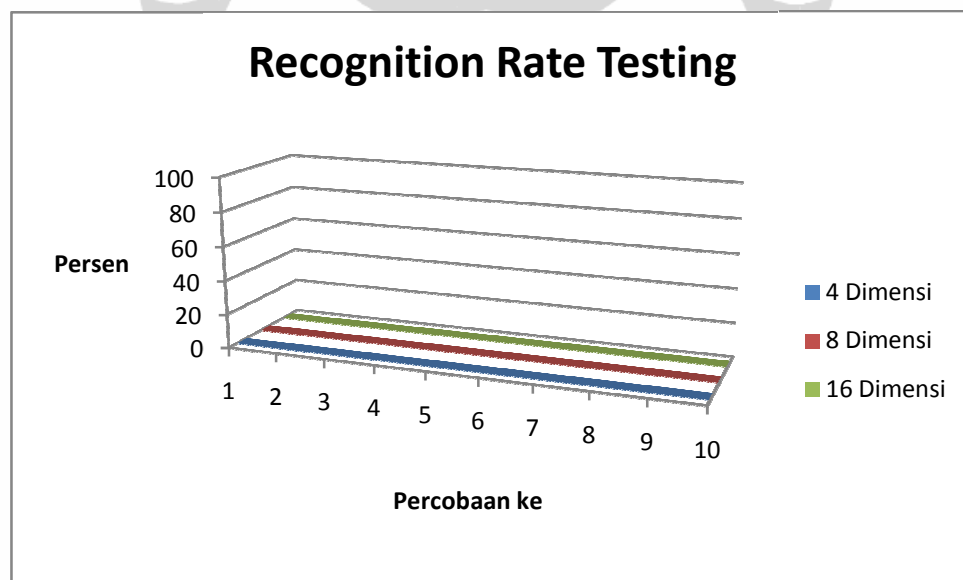
Tabel 4.1 Recognition Rate Testing Data Original untuk 1000 epoch

Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.00	0.00	0.00
10	0.00	0.00	0.00
Rata-rata	0.00	0.00	0.00
Waktu	6304.00	6058.30	5953.3

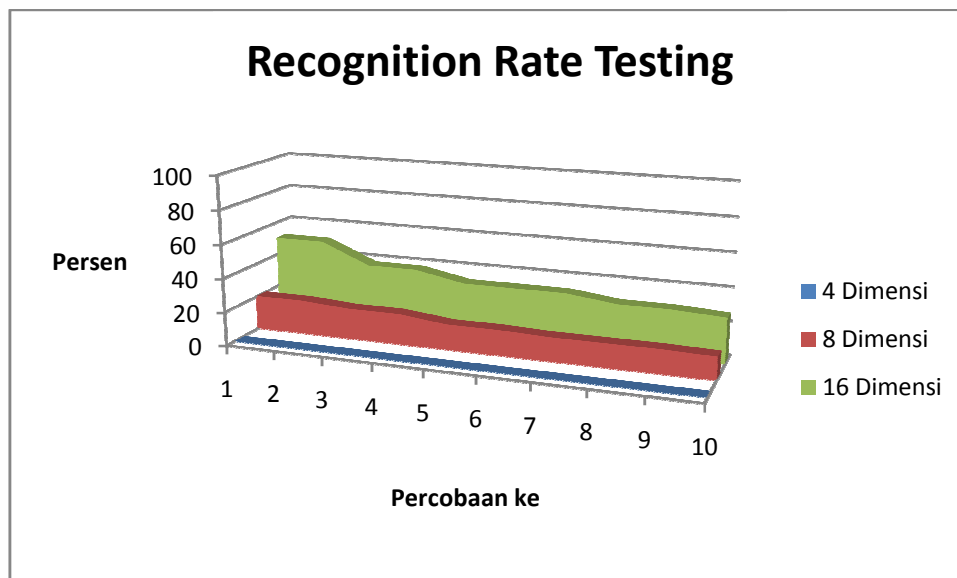
Tabel 4.2 Recognition Rate Testing Data Normalisasi untuk  
1000 epoch

Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0.00	20.33	50.33
2	0.00	20.33	50.33
3	0.00	18.44	38
4	0.00	18.44	38
5	0.00	15.22	32.33
6	0.00	15.22	32.33
7	0.00	14.22	32.33
8	0.00	14.22	28.78
9	0.00	14.22	28.78
10	0.00	13.44	27.11
Rata-rata	0.00	16.41	35.83
Waktu	6176.44	5796..26	6218.64

Untuk mempermudah proses analisa, data direpresentasikan kembali dalam bentuk grafik dapat dilihat seperti di bawah ini



Gambar 4.1 Grafik Recognition Rate Testing Data Original untuk  
1000 epoch



Gambar 4.2. Recognition Rate Testing data Normalisasi untuk 1000 epoch

Dari gambar 4.1 dan gambar 4.2 untuk data 4 dimensi, data 8 dimensi dan data 16 dimensi terlihat bahwa dalam metode FBR data normalisasi dapat diklasifikasikan lebih baik dari pada data aslinya, hal ini dikarenakan data hasil normalisasi memiliki tingkat independensi yang lebih baik. Pada proses FBR metode pengklasifikasi sangat dipengaruhi oleh perubahan bobot, center dan spread.

Pada data normalisasi, komponen dari perubahan bobot, center dan spread memiliki tingkat perbedaan yang lebih jelas jika dibandingkan dengan tingkat perbedaan pada data aslinya. Sehingga proses perubahan tiga parameter tersebut yang dihasilkan akan semakin baik, akibatnya proses klasifikasi juga akan menjadi semakin baik.

**BAB 5**  
**ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS**  
**RADIAL FUZZY MENGGUNAKAN SOM**

**5.1. Skema Jaringan Saraf FBR Fuzzy Menggunakan SOM**

Sama seperti pada penelitian jaringan Saraf FBR menggunakan SOM, jaringan saraf FBR fuzzy ini juga terdiri dari tiga tahapan, yaitu: data percobaan, pelatihan atau training, dan pengujian atau testing.

**5.1.1. Data Percobaan**

Data percobaan yang digunakan dalam penelitian ini adalah :

1. Data Odor 2 campuran

Ada 3 aroma (odor) alami yang diuji yaitu kenanga , jeruk dan mawar Masing-masing aroma dicampur dengan alkohol dengan 6 persentase yang berbeda sehingga secara keseluruhan didapatkan 18 jenis odor, di mana masing-masing jenis odor diambil 200 data eksperimen

- Eksperimen dengan 16 dimensi (yang mewakili 16 sensor odor yang digunakan)
- Eksperimen dengan 8 dimensi (yang mewakili 8 sensor odor yang digunakan)
- Eksperimen dengan 4 dimensi (yang mewakili 4 sensor odor yang digunakan)

2. Data aroma jeruk 0% alkohol, jeruk 15% alkohol dan jeruk 25% alkohol, masing- masing kelas sebanyak 50 data dan 4 dimensi (odor\_4d\_a)

3. Data aroma jeruk 0% alkohol, mawar 0% dan kenanga 0% alkohol, masing-masing kelas sebanyak 50 data dan 4 dimensi (odor\_4d\_b)

4. Data Iris 4 dimensi sebagai data pembanding dengan odor 4 dimensi ( Iris\_4d).



### 5.1.2. Pelatihan atau Training

Sistem dilatih untuk mengenali pola masukan agar menghasilkan keluaran yang sesuai dengan target tertentu. Awal pelatihan digunakan pelatihan tanpa arahan untuk melakukan pengelompokan data, kemudian setelah itu digunakan pelatihan dengan arahan untuk menggeneralisasi nilai masukan. Jaringan ini hanya mempunyai satu lapisan tersembunyi. Nilai bobot, center, dan spread akan dimodifikasi terus menerus sampai sistem ini menemukan nilai akhir yang konvergen. Dalam penelitian ini, satu kali pelatihan dinyatakan sebagai satu epoch

### 5.1.3. Pengujian atau Testing

Tahap pengujian ini berupa verifikasi suatu pola akan ditempatkan dalam kelas yang mana. Pada penelitian, tahap pengujian ini dilakukan dengan menggunakan pola yang sama namun tidak sama dengan pola yang digunakan pada pelatihan.

Dalam proses pengujian, setiap pola dijalankan ke dalam jaringan SOM dengan menggunakan nilai bobot akhir atau vektor perwakilan akhir yang telah disimpan sebelumnya untuk melakukan pengelompokan data, setelah itu dijalankan dalam jaringan *backpropagation* tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang disimpan sebelumnya.

Untuk satu kali pengujian, persentase kesesuaian terhadap target yang disebut recognition rate pun dihitung.

## 5.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Fuzzy Menggunakan SOM

Parameter yang digunakan dalam penelitian jaringan saraf FBR Fuzzy menggunakan SOM sama dengan bab tiga terdiri dari inisialisasi bobot, neuron yang digunakan pada lapisan masukan, lapisan tersembunyi dan lapisan keluaran, banyaknya epoch, dan besarnya laju pembelajaran, serta nilai kesalahan uji.

### 5.3. Jaringan Saraf Tiruan FBR Fuzzy Menggunakan SOM

Pada penelitian sistem pengenalan pola ini, penulis menggunakan jaringan saraf tiruan fungsi basis radial menggunakan membership function. Secara garis besar jaringan ini sama dengan jaringan yang digunakan pada bab tiga, namun ada penambahan algoritma membership function pada algoritmanya. Proses pelatihan pada jaringan saraf SOM merupakan proses pengelompokan (*clustering*) vektor data yang mana setiap kelompok atau *cluster* berisi vektor-vektor data yang memiliki kemiripan yang paling dekat dengan vektor masukan. Ukuran kemiripan yang digunakan dalam penelitian ini adalah jarak Euclidean yang paling minimum. Banyaknya kelompok atau *cluster* menentukan banyaknya neuron yang digunakan dalam lapisan tersembunyi. Setelah proses pelatihan secara berulang-ulang selesai dan memiliki bobot akhir, maka dilakukan proses pengelompokan vektor yang gunanya untuk mencari nilai awal *center* dan *spread*, caranya setiap kelompok dihitung nilai rata-ratanya dan setelah itu dihitung nilai *spread* (penyebaran) dengan cara mencari jarak maksimum antara dua kelompok atau *cluster* dibagi dengan akar dari banyaknya kelompok atau *cluster*.

Pada proses selanjutnya yaitu proses pelatihan propagasi balik, untuk setiap iterasi dihitung nilai derajat keanggotaan atau *membership function* setiap vektor data pada setiap kelompok, dihitung nilai pembaharuan *center* yang pertama menggunakan derajat keanggotaan atau *membership function*. Kemudian untuk setiap vektor data melakukan proses propagasi maju, propagasi balik dan perubahan bobot, *center* dan *spread*. Pada proses propagasi maju dihitung nilai  $Z_j$  ( $j = 1, 2, \dots, p$ ) berdasarkan nilai *center* yang pertama dengan menggunakan fungsi aktivasi Gaussian. Hasil penghitungan proses Gaussian akan dikomputasi dengan bobot-bobot yang berkesesuaian dengannya ( $W_{jk}$ ) sehingga menghasilkan nilai pada neuron-neuron lapisan keluaran  $Y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan fungsi aktivasi Sigmoid Biner. Proses propagasi balik adalah proses yang dilakukan setelah proses propagasi maju dalam bagian ini dihitung informasi *Error* nya ( $\delta_k$ ) yang nilainya akan digunakan dalam menghitung besarnya koreksi bobot neuron

keluaran dan besarnya koreksi bias keluaran. Setelah proses propagasi maju dan propagasi balik selesai maka dihitung nilai perubahan bobot, *center* yang kedua dan *spread* berdasarkan informasi *Error*nya. Setelah proses pelatihan kemudian dilakukan proses pengujian untuk pola yang lama atau pola yang baru. Proses yang dilakukan untuk pengenalan pola ini hanyalah pengelompokan data dan propagasi maju.

Pada penelitian ini, untuk memudahkan perhitungan pada program, nilai sebuah target dinyatakan dalam matriks yang ukurannya disesuaikan dengan neuron keluaran.

Agar jaringan dapat mempelajari pola dengan baik, pola yang mempunyai target yang sama sebaiknya tidak dilatih secara berurutan, melainkan silih berganti dengan pola yang mempunyai target berbeda, misalnya pola dengan target ke-1 dilanjutkan dengan pola target ke-2, dan seterusnya. Setelah itu, barulah pelatihan dilanjutkan dengan pola dengan target ke-1 kembali hingga seluruh pola selesai dilatih.

Detail algoritma pelatihan jaringan ini adalah sebagai berikut:

**Langkah 0 : Inisialisasi**

- SOM
  - Bobot awal  $V_{ji}$  di ambil dr salah satu vektor dalam setiap kelas
  - Set parameter laju pembelajaran (Learning Rate ( $\alpha$ ))
- *Backpropagation*
  - Bobot awal  $w_{kj}$  menggunakan nguyen widrow.
  - Bias awal  $w_{0k}$
  - Set parameter laju pembelajaran (Learning rate)

**Proses SOM:**

**Langkah 1 : Clustering Data, menggunakan SOM**

- Langkah a : Jika kondisi henti gagal, lakukan langkah b-g
- Langkah b : Untuk setiap vektor masukan  $x$ , lakukan langkah c sampai e
- Langkah c : Untuk setiap  $j$ , hitung:

$$D_i = \sqrt{\sum_{j=1}^m (V_{ji} - X_i)^2} \quad (5.1)$$

- Langkah d : Temukan indeks j sehingga  $D(j)$  minimum
- Langkah e : Untuk setiap neuron J
  - meng-update bobotnya

$$v_{ji}(\text{new}) = v_{ji}(\text{lama}) + \alpha (x_i - v_{ji}(\text{lama})) \quad (5.2)$$

- Langkah f : Memodifikasi laju pemahaman
- Langkah g : melanjutkan iterasi

### Langkah 2 : Inisialisasi center dan Spread

- Langkah a : menghitung jumlah/ banyaknya vector data dalam kelas
- Langkah b : menghitung rata-rata vektor /center (c)

$$c_j = \frac{\sum_{i=1}^p x_{pj}}{N_j} \quad (5.3)$$

- Langkah c : menghitung nilai spread ( $\sigma$ )

$$\sigma = \frac{\text{jarak maksimum antara 2 center}}{\sqrt{\text{banyaknya center}}} = \frac{d_{maks}}{\sqrt{k}} \quad (5.4)$$

### Proses Backpropagation:

#### Langkah 3 : Untuk setiap epoch

- Langkah a : Menghitung membership function dari tiap data ke tiap kelas j

$$\mu_{ij} = \frac{[\sum_{j=1}^m (\bar{x}_{ij} - \bar{c}_j)^2]^{-1/2}}{\sum_{n=1}^N [\sum_{j=1}^m (\bar{x}_{ij} - \bar{c}_j)^2]^{-1/2}} \quad (5.5)$$

Dengan  $n=1,2,\dots,N$

$N$ = jumlah kelas

$\mu_{ij}$ = membership function data ke i di cluster ke j

- Langkah b : Meng-update center dengan menggunakan membership function

$$\bar{c}_{j\text{new}} = \frac{\sum_{i=1}^m (\mu_{ij})^2 \cdot \bar{c}_j}{\sum_{i=1}^m \mu_{ij}^2} \quad (5.6)$$

**Langkah 4 :** Untuk masing masing vektor kelas j

**Feedforward :**

- Langkah a : menghitung nilai  $z_j$

$$z_j = \exp\left(-\frac{(x_i - c_j)^T (x_i - c_j)}{2(\sigma_j)^2}\right) = \exp(-z_{in}) \quad (5.7)$$

- Langkah b : menghitung nilai  $y_{ink}$

$$y_{ink} = w_0(k) + \sum z_j \cdot w_{kj} \quad (5.8)$$

- Langkah c : menghitung nilai  $Y_k$

$$Y_k = f(-y_{ink})$$

$$Y_k = 1/(1 + \exp(-y_{ink})) \quad (5.9)$$

**Backward :**

- Langkah d : menghitung nilai  $\delta_k$

$$\delta_k = (T_k - Y_k) f'(y_{ink})$$

$$\delta_k = (T_k - Y_k) (Y_k)(1 - Y_k) \quad (5.10)$$

- Langkah e : Menghitung besarnya koreksi bobot unit output

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (5.11)$$

- Langkah f : Menghitung besarnya koreksi bias output

$$\Delta w_0_k = \alpha \delta_k \quad (5.12)$$

**Perubahan Bobot**

- Langkah g : Setiap unit output mengup-date bobotnya

$$w_{kj}(\text{new}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (5.13)$$

- Langkah h : Setiap unit output mengup-date biasnya

$$w_0_k(\text{new}) = w_0_k(\text{lama}) + \Delta w_0_k \quad (5.14)$$

**Perubahan center**

- Langkah i : mengup-date center

$$c_j(\text{new}) = c_j(\text{lama}) + \alpha \left( \frac{do(k) \cdot w_{kj}}{\sigma_j^2} \right) \cdot z_j \cdot [X_i - c_j] \quad (5.15)$$

**Perubahan nilai spread**

- Langkah j : mengup-date nilai spread

$$\sigma_j(\text{new}) = \sigma_j(\text{lama}) + \alpha \left( \frac{do(k) \cdot w_{kj}}{\sigma_j^3} \right) \cdot z_j \cdot ((x_i - c_j)^T (x_i - c_j)) \quad (5.16)$$

- Langkah k : kondisi stopping

$$E_{\min} < 0.01 \quad (5.17)$$

Setelah melakukan pelatihan, dilakukan pengenalan pola, kemudian agar seragam diambil pola dengan jumlah yang sama untuk setiap target. Pada proses pengenalan pola ini, sistem akan memberi jawaban terhadap setiap pola, masuk ke *cluster* yang mana dari *cluster* yang ada.

Apabila jawaban yang diberikan sesuai target, maka jaringan dianggap telah mengenali pola dengan benar. Namun bila jawaban yang diberikan tidak sesuai dengan target, maka jaringan dinilai salah dalam mengenali pola. Sistem tidak dapat menghasilkan jawaban bahwa suatu pola berada di luar target yang ditentukan. Dengan demikian, hanya ada dua pilihan, yaitu benar dan salah. Persentase jawaban yang benar terhadap seluruh pertanyaan mengacu kepada *recognition rate*.

#### 5.4. Pengujian Pada Jaringan Saraf Tiruan FBR Fuzzy menggunakan SOM

Pada metode FBR Fuzzy menggunakan SOM, pengujian dilakukan dengan menggunakan tiga jenis data dengan :

- 1 . Rasio perbandingan data training dan testing 70:30
- 2 . Data yang diuji yaitu data original dan data normalisasi

Adapun data yang dilihat sebagai hasil pengujian adalah tingkat kemampuan pengenalan pola (Recognition Rate) dan waktu yang diperlukan untuk mengenali pola tersebut.

Perbandingan data yang digunakan 70 : 30 berarti 70% data dijadikan sebagai data pembelajaran jaringan dan 30% data sisanya digunakan sebagai data pengujian.

##### 5.4.1. Hasil Pengujian dan Analisa

Berikut ini tabel 5.1 menunjukkan hasil ujicoba untuk data original dalam mendeteksi data yang berdimensi 4, 8, dan 16 dan tabel 5.2 menunjukkan hasil ujicoba untuk data normalisasi dalam mendeteksi data yang berdimensi 8, dan 16.

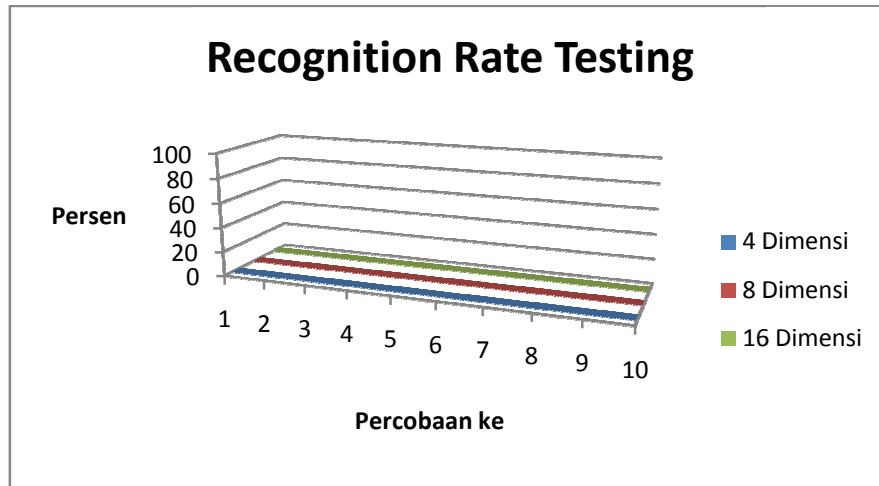
Tabel 5.1 Recognition Rate Testing Data Original untuk 1000 epoch

Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.00	0.00	0.00
10	0.00	0.00	0.00
Rata-rata	0.00	0.00	0.00
Waktu	6944.19	7110.02	6497.47

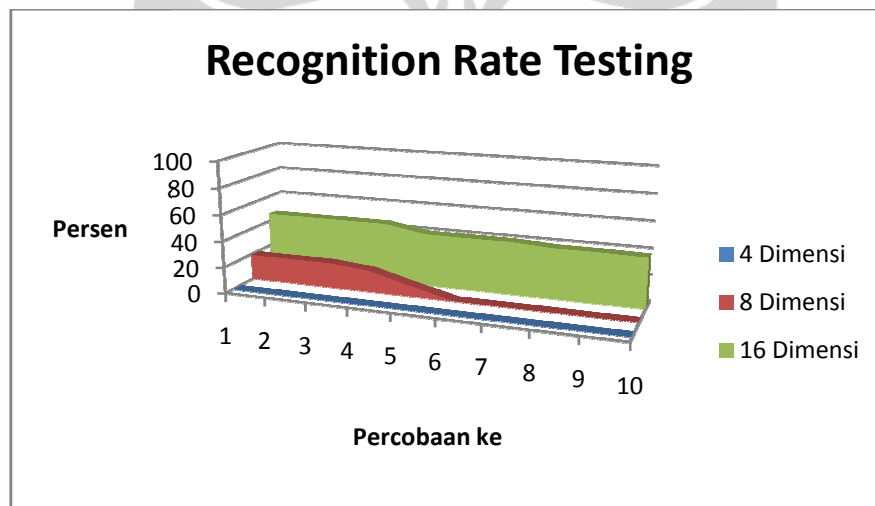
Tabel 5.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch

Percobaan ke-	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0	19.22	45.67
2	0	19.22	45.67
3	0	19.22	45.67
4	0	16.33	45.67
5	0	8	40.33
6	0	0.22	40.33
7	0	0.22	40.33
8	0	0.22	38.56
9	0	0	38.56
10	0	0	38
Rata-rata	0.00	8.27	41.88
Waktu	7484.48	6970.66	6465.04

Untuk mempermudah proses analisa, data direpresentasikan kembali dalam bentuk grafik dapat dilihat seperti di bawah ini



Gambar 5.1 Grafik Recognition Rate Testing Data Original untuk 1000 epoch



Gambar 5.2 Grafik Recognition Rate Testing Data Normalisasi untuk 1000 epoch

Dari gambar 5.1 dan gambar 5.2 untuk data 8 dimensi dan data 16 dimensi terlihat bahwa dalam metode FBR data normalisasi dapat diklasifikasikan lebih baik dari pada data aslinya, hal ini dikarenakan data hasil normalisasi memiliki tingkat independensi yang lebih baik., sedangkan untuk data 4 dimensi tidak menggunakan data normalisasi karena hasil uji coba dari data original sudah menunjukkan hasil yang baik. Pada proses FBR metode



pengklasifikasi sangat dipengaruhi oleh perubahan membership function bobot, center yang mengalami perubahan sebanyak dua kali dan spread.

Pada data normalisasi, komponen dari perubahan membership function, bobot, center dan spread memiliki tingkat perbedaan yang lebih jelas jika dibandingkan dengan tingkat perbedaan pada data aslinya. Sehingga proses perubahan lima parameter tersebut yang dihasilkan akan semakin baik, akibatnya proses klasifikasi juga akan menjadi semakin baik.

## **BAB 6**

### **ANALISA PERFORMA JARINGAN SARAF FUNGSI BASIS RADIAL FUZZY TANPA SOM**

#### **6.1. Skema Jaringan Saraf FBR Fuzzy Tanpa SOM**

Sama seperti pada penelitian bab tiga, jaringan saraf FBR fuzzy tanpa SOM juga terdiri dari tiga tahapan, yaitu: data percobaan, pelatihan atau training, dan pengujian atau testing. Hanya saja pada penelitian ini pada proses pelatihan dan pengujian, algoritma SOM tidak digunakan untuk menentukan cluster atau kelompok data.

##### **6.1.1. Data Percobaan**

Data percobaan yang digunakan dalam penelitian ini adalah sama dengan data percobaan yang digunakan pada jaringan Saraf tiruan FBR fuzzy menggunakan SOM..

##### **6.1.2. Pelatihan atau Training**

Sistem dilatih untuk mengenali pola masukan agar menghasilkan keluaran yang sesuai dengan target tertentu. Dalam penelitian ini tidak digunakan pelatihan tanpa arahan untuk menentukan kelompok data, namun pengelompokan data dilakukan secara manual, setelah itu digunakan pelatihan dengan arahan untuk mengeneralisasi nilai masukan. Jaringan ini hanya mempunyai satu lapisan tersembunyi. Nilai bobot, center, dan spread akan dimodifikasi terus menerus sampai sistem ini menemukan nilai akhir yang konvergen. Dalam penelitian ini, satu kali pelatihan dinyatakan sebagai satu epoch.

##### **6.1.3. Pengujian atau Testing**

Tahap pengujian ini berupa verifikasi suatu pola akan ditempatkan dalam kelas yang mana. Pada penelitian, tahap pengujian ini dilakukan dengan menggunakan pola yang sama namun tidak sama dengan pola yang digunakan pada pelatihan.

Dalam proese pengujian, setiap pola dijalankan dalam jaringan *backpropagation* tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang disimpan sebelumnya.

Untuk satu kali pengujian, persentase kesesuaian terhadap target yang disebut recognition rate pun dihitung.

## **6.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf FBR Fuzzy Tanpa SOM**

Parameter yang digunakan dalam penelitian jaringan saraf FBR menggunakan membership function dan tanpa algoritma SOM sama dengan bab empat terdiri dari inisialisasi bobot, neuron yang digunakan pada lapisan masukan, lapisan tersembunyi dan lapisan keluaran, banyaknya epoch, dan besarnya laju pembelajaran.

## **6.3. Jaringan Saraf Tiruan FBR Fuzzy Tanpa SOM**

Pada penelitian ini jaringan saraf tiruan yang digunakan sama dengan saraf tiruan yang digunakan pada bab ke-5, yaitu jaringan saraf fungsi basis radial menggunakan membership function, namun proses SOM tidak dilakukan untuk pengelompokan vektor data dalam bab ini. Proses pengelompokan vektor data dilakukan secara manual dengan memasukan vektor data secara langsung ke cluster yang sesuai dengan cluster vektor data, Setiap cluster terdapat beberapa vektor data untuk proses pelatihan, pengelompokan ini digunakan untuk mencari nilai awal *center* dan *spread*, caranya setiap kelompok dihitung nilai rata-ratanya dan setelah itu dihitung nilai *spread* (penyebaran) dengan cara mencari jarak maksimum antara dua kelompok atau *cluster* dibagi dengan akar dari banyaknya kelompok atau *cluster*.

Pada proses selanjutnya yaitu proses pelatihan propagasi balik, untuk setiap iterasi dihitung nilai derajat keanggotaan atau *membership function* setiap vektor data pada setiap kelompok, dihitung nilai pembaharuan *center* yang pertama menggunakan derajat keanggotaan atau *membership function*. Kemudian untuk setiap vektor data melakukan proses propagasi maju, propagasi balik dan perubahan bobot, *center* dan *spread*. Pada proses

propagasi maju dihitung nilai  $Z_j$  ( $j = 1, 2, \dots, p$ ) berdasarkan nilai *center* dengan menggunakan fungsi aktivasi Gaussian. Hasil penghitungan proses Gaussian akan dikomputasi dengan bobot-bobot yang berkesesuaian dengannya ( $W_{jk}$ ) sehingga menghasilkan nilai pada neuron-neuron lapisan keluaran  $Y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan fungsi aktivasi Sigmoid Biner. Proses propagasi balik adalah proses yang dilakukan setelah proses propagasi maju dalam bagian ini dihitung informasi *Error* nya ( $\delta_k$ ) yang nilainya akan digunakan dalam menghitung besarnya koreksi bobot neuron keluaran dan besarnya koreksi bias keluaran. Setelah proses propagasi maju dan propagasi balik selesai maka dihitung nilai perubahan bobot, *center* yang kedua dan *spread* berdasarkan informasi *Error*nya. Setelah proses pelatihan kemudian dilakukan proses pengujian untuk pola yang lama atau pola yang baru. Proses yang dilakukan untuk pengenalan pola ini hanyalah propagasi maju.

Pada penelitian ini, untuk memudahkan perhitungan pada program, nilai sebuah target dinyatakan dalam matriks yang ukurannya disesuaikan dengan neuron keluaran.

Agar jaringan dapat mempelajari pola dengan baik, pola yang mempunyai target yang sama sebaiknya tidak dilatih secara berurutan, melainkan silih berganti dengan pola yang mempunyai target berbeda, misalnya pola dengan target ke-1 dilanjutkan dengan pola target ke-2, dan seterusnya. Setelah itu, barulah pelatihan dilanjutkan dengan pola dengan target ke-1 kembali hingga seluruh pola selesai dilatih.

Detail algoritma pelatihan jaringan ini adalah sebagai berikut:

**Langkah 0** : Inisialisasi

- *Backpropagation*
  - Bobot awal  $w_{kj}$  menggunakan nguyen widrow.
  - Bias awal  $w_{0k}$
  - Set parameter laju pembelajaran (Learning rate)

**Langkah 1** : Clustering Data secara manual.

**Langkah 2** : Inisialisasi center dan Spread

- Langkah a : menghitung jumlah/ banyaknya vector data dalam kelas
- Langkah b : menghitung rata-rata vektor /center (c)

$$c_j = \frac{\sum_{i=1}^p x_{pj}}{N_j} \quad (6.1)$$

- Langkah c : menghitung nilai spread ( $\sigma$ )

$$\sigma = \frac{\text{jarak maksimum antara 2 center}}{\sqrt{\text{banyaknya center}}} = \frac{d_{maks}}{\sqrt{k}} \quad (6.2)$$

**Proses Backpropagation:**

**Langkah 3 :** Untuk setiap iterasi

- Langkah a : Menghitung membership function dari tiap data ke tiap kelas j

$$\mu_{ij} = \frac{[\sum_{j=1}^m (\bar{x}_{ij} - \bar{c}_j)^2]^{-1/2}}{\sum_{n=1}^N [\sum_{j=1}^m (\bar{x}_{ij} - \bar{c}_j)^2]^{-1/2}} \quad (6.3)$$

Dengan  $n=1,2,\dots,N$

$N$ = jumlah kelas

$\mu_{ij}$ = membership function data ke i di cluster ke j

- Langkah b : Meng-update center dengan menggunakan membership function

$$\bar{c}_{j\text{new}} = \frac{\sum_{i=1}^m (\mu_{ij})^2 \cdot \bar{c}_j}{\sum_{i=1}^m \mu_{ij}^2} \quad (6.4)$$

**Langkah 4 :** Untuk masing masing vektor kelas j

**Feedforward :**

- Langkah a : menghitung nilai  $z_j$

$$z_j = \exp\left(-\frac{(x_i - c_j)^T (x_i - c_j)}{2(\sigma_j)^2}\right) = \exp(-z_{in}) \quad (6.5)$$

- Langkah b : menghitung nilai yink

$$y_{ink} = w_0(k) + \sum z_j \cdot w_{kj} \quad (6.6)$$

- Langkah c : menghitung nilai  $Y_k$

$$Y_k = f(-y_{ink})$$

$$Y_k = 1/(1 + \exp(-y_{ink})) \quad (6.7)$$

#### Backward :

- Langkah d : menghitung nilai  $\delta_k$

$$\delta_k = (T_k - Y_k) f'(y_{ink})$$

$$\delta_k = (T_k - Y_k) (Y_k)(1 - Y_k) \quad (6.8)$$

- Langkah e : Menghitung besarnya koreksi bobot unit output

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (6.9)$$

- Langkah f : Menghitung besarnya koreksi bias output

$$\Delta w_{0k} = \alpha \delta_k \quad (6.10)$$

#### Perubahan Bobot

- Langkah g : Setiap unit output mengup-date bobotnya

$$w_{kj}(\text{new}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (6.11)$$

- Langkah h : Setiap unit output mengup-date biasnya

$$w_{0k}(\text{new}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (6.12)$$

#### Perubahan center

- Langkah i : mengup-date center

$$c_j(\text{new}) = c_j(\text{lama}) + \alpha \left( \frac{do(k) \cdot w_{kj}}{\sigma_j^2} \right) \cdot z_j \cdot [X_i - c_j] \quad (6.13)$$

#### Perubahan nilai spread

- Langkah j : mengup-date nilai spread

$$\sigma_j(\text{new}) = \sigma_j(\text{lama}) + \alpha \left( \frac{do(k) \cdot w_{kj}}{\sigma_j^3} \right) \cdot z_j \cdot ((x_i - c_j)^T (x_i - c_j)) \quad (6.14)$$

- Langkah k : kondisi stopping

$$E_{\min} < 0.01 \quad (6.15)$$

Setelah melakukan pelatihan, dilakukan pengenalan pola, agar seragam diambil pola dengan jumlah yang sama untuk setiap target. Pada proses pengenalan pola ini, sistem akan memberi jawaban terhadap setiap pola, masuk ke *cluster* yang mana dari tiga *cluster* yang ada.

Apabila jawaban yang diberikan sesuai target, maka jaringan dianggap telah mengenali pola dengan benar. Namun bila jawaban yang diberikan tidak sesuai dengan target, maka jaringan dinilai salah dalam mengenali pola. Sistem tidak dapat menghasilkan jawaban bahwa suatu pola berada di luar target yang ditentukan. Dengan demikian, hanya ada dua pilihan, yaitu benar dan salah. Persentase jawaban yang benar terhadap seluruh pertanyaan mengacu kepada *recognition rate*.

#### **6.4. Pengujian Pada Jaringan Saraf Tiruan FBR Fuzzy Tanpa SOM**

Pada metode FBR Membership Function tanpa menggunakan SOM, pengujian dilakukan dengan menggunakan tiga jenis data dengan :

1. Rasio perbandingan data training dan testing 70:30
2. Data yang diuji yaitu data original dan data normalisasi

Adapun data yang dilihat sebagai hasil pengujian adalah tingkat kemampuan pengenalan pola (Recognition Rate) dan waktu yang diperlukan untuk mengenali pola tersebut.

Perbandingan data yang digunakan 70 : 30 berarti 70% data dijadikan sebagai data pembelajaran jaringan dan 30% data sisanya digunakan sebagai data pengujian.

##### **6.4.1. Hasil Pengujian dan Analisa**

Berikut ini tabel 6.1 menunjukkan hasil ujicoba untuk data original dalam mendeteksi data yang berdimensi 4, 8, dan 16 dan tabel 6.2 menunjukkan hasil ujicoba untuk data normalisasi dalam mendeteksi data yang berdimensi 8, dan 16.

Tabel 6.1 Recognition Rate Testing Data Original untuk 1000 epoch

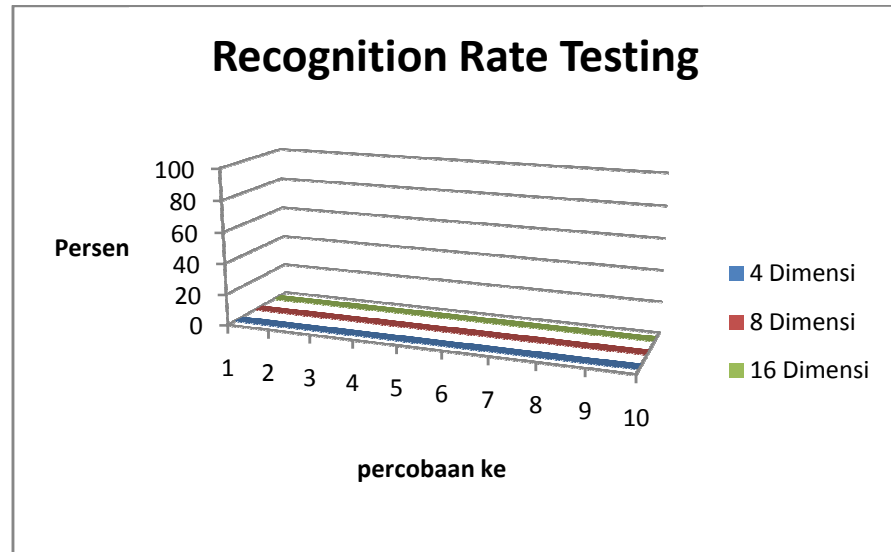
Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.00	0.00	0.00
10	0.00	0.00	0.00
Rata-rata	0.00	0.00	0.00
Waktu	6456.51	7160.17	6538.09

Tabel 6.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch

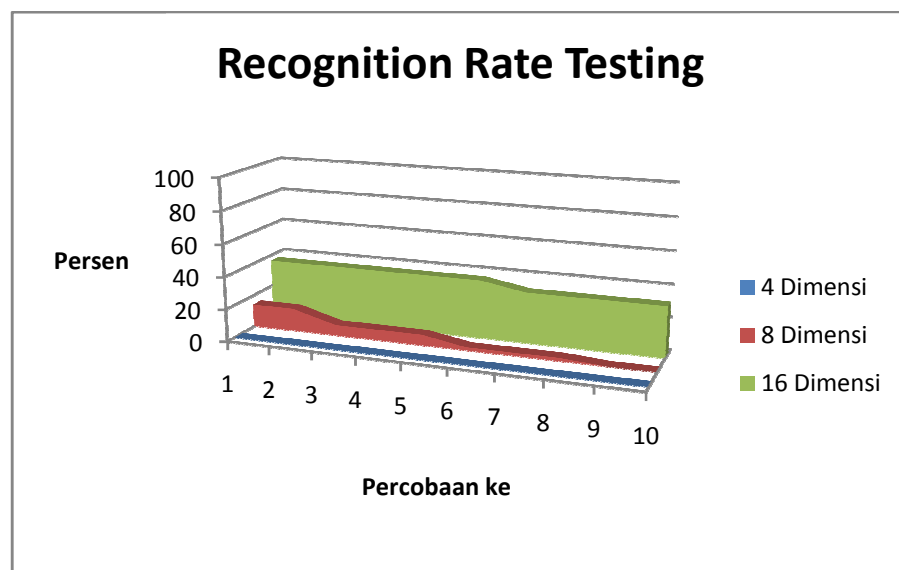
Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	0.00	14	36.22
2	0.00	14	36.22
3	0.00	6.56	36.22
4	0.00	6.56	36.11
5	0.00	6.56	36.11
6	0.00	1.89	36.11
7	0.00	1.89	31.33
8	0.00	1.89	31.33
9	0.00	0.00	31.33
10	0.00	0.00	31.33
Rata-rata	0.00	5.33	34.23
Waktu	6587.80	7020.57	6950.09

Untuk mempermudah proses analisa, data direpresentasikan kembali dalam bentuk grafik dapat dilihat seperti di bawah ini.





Gambar 6.1. Grafik Recognition Rate Testing Data Original untuk 1000 epoch



Gambar 6.2. Recognition Rate Testing Data Normalisasi untuk 1000 epoch

Dari gambar 6.1 dan gambar 6.2 untuk data 8 dimensi dan data 16 dimensi terlihat bahwa dalam metode FBR data normalisasi dapat diklasifikasikan lebih baik dari pada data aslinya, hal ini dikarenakan data hasil normalisasi memiliki tingkat independensi yang lebih baik., sedangkan untuk data 4 dimensi tidak menggunakan data normmalisasi karena hasil uji coba dari data original sudah menunjukkan hasil yang baik. Pada proses FBR metode

pengklasifikasi sangat dipengaruhi oleh perubahan membership function bobot, center yang mengalami perubahan sebanyak dua kali dan spread.

Pada data normalisasi, komponen dari perubahan membership function, bobot, center dan spread memiliki tingkat perbedaan yang lebih jelas jika dibandingkan dengan tingkat perbedaan pada data aslinya. Sehingga proses perubahan lima parameter tersebut yang dihasilkan akan semakin baik, akibatnya proses klasifikasi juga akan menjadi semakin baik.

## BAB 7

### ANALISA PERFORMA JARINGAN SARAF BACKPROPAGATION

Pada bab ini penulis akan memaparkan skema penelitian jaringan saraf *backpropagation*, parameter-parameter yang digunakan, bagaimana jaringan saraf *backpropagation* dapat digunakan untuk sistem pengenalan pola, dan bagaimana parameter-parameter pada jaringan *backpropagation* berpengaruh terhadap hasil pelatihan dan pengujian.

#### 7.1. Skema Penelitian Jaringan Saraf *Backpropagation*

Sama seperti pada penelitian bab tiga, jaringan saraf *backpropagation* terdiri dari tiga tahapan, yaitu: data percobaan, pelatihan atau training, dan pengujian atau testing.

##### 7.1.1. Data Percobaan

Data percobaan yang digunakan dalam penelitian ini adalah sama dengan data percobaan yang digunakan pada jaringan Saraf tiruan FBR menggunakan SOM.

##### 7.1.2. Pelatihan atau Training

Setelah data percobaan didapatkan, tahap selanjutnya yang merupakan bagian utama sistem pengenalan pola adalah pelatihan pola atau training. Sistem dilatih untuk mengenali pola masukan agar menghasilkan keluaran yang sesuai dengan target tertentu. Pola dilatih dengan menggunakan pelatihan dengan arahan untuk menggeneralisasi nilai masukan. Jaringan ini hanya mempunyai satu lapisan tersembunyi. Nilai bobot akan dimodifikasi terus menerus sampai sistem ini menemukan nilai akhir yang konvergen. Dalam penelitian ini, satu kali pelatihan dinyatakan sebagai satu epoch

##### 7.1.3. Pengujian atau Testing

Tahap pengujian ini berupa verifikasi suatu pola akan ditempatkan dalam kelas yang mana. Pada penelitian, tahap pengujian ini dilakukan dengan

menggunakan pola yang sama juga namun tidak sama dengan pola yang digunakan pada pelatihan.

Dalam proses pengujian, setiap pola dijalankan dalam jaringan *backpropagation* tetapi hanya sampai pada tahap propagasi maju saja dengan menggunakan nilai bobot akhir yang disimpan sebelumnya.

Untuk satu kali pengujian, persentase kesesuaian terhadap target yang disebut recognition rate pun dihitung.

## **7.2. Parameter yang Digunakan sebagai Variabel pada Jaringan Saraf *Backpropagation***

Parameter yang digunakan dalam penelitian jaringan saraf backpropagation dan sekaligus menjadi variabel bebas, antara lain :

### 1. Inisialisasi bobot

Dalam penelitian ini ada 2 macam inisialisasi bobot, inisialisasi bobot antara lapisan masukan dengan lapisan tersembunyi dan inisialisasi antara lapisan tersembunyi dengan lapisan keluaran. Kedua-duanya pada jaringan saraf propagasi balik ditentukan secara random dan metode Nguyen-Widrow

### 2. Banyaknya neuron yang digunakan pada lapisan masukan, lapisan tersembunyi, dan lapisan keluaran disesuaikan dengan pola yang digunakan.

### 3. Besarnya epoch yang digunakan

Banyaknya epoch yang digunakan adalah 1000 epoch.

### 4. Besarnya laju pembelajaran ( $\alpha$ )

Laju pembelajaran untuk algoritma propagasi balik sebesar 0.2. Sementara itu, variabel terikat yang digunakan pada penelitian ini adalah nilai recognition rate pada tahap pengujian dan grafik *Error* terhadap epoch. Berdasarkan hasil tersebut, akan ditentukan parameter-parameter yang dapat mengoptimalkan performa jaringan, yang selanjutnya disebut parameter optimalisasi.

5. Momentum ( $\mu$ )  
Peneliti menggunakan momentum sebesar 0.5
6. Nilai Kesalahan Uji  
Range nilai kesalahan uji yang digunakan dalam penelitian ini adalah 0.49 untuk nilai 0 dan range 0.51 untuk nilai 1

### 7.3. Jaringan Saraf Tiruan *Backpropagation*

Seperti yang telah dijelaskan bab II, jaringan saraf tiruan adalah jaringan yang prinsip kerjanya menyerupai jaringan saraf manusia, di mana jaringan ini dapat mempelajari suatu pola yang diberikan dengan target tertentu. Dari pola-pola tersebut, proses pembelajaran dilakukan dengan bobot-bobot atau koefisien-koefisien di antara neuron-neuronnya secara berulang-ulang sehingga pada akhirnya akan mendapatkan suatu tetapan yang *default*, yang dapat digunakan untuk mengidentifikasi suatu pola baru.

Jaringan saraf tiruan yang digunakan pada penelitian ini terdiri dari tiga lapisan yaitu : lapisan masukan  $X_i$  (*input layer*), lapisan tersembunyi  $Z_j$  (*hidden layer*), dan lapisan keluaran  $Y_k$  (*output layer*). Lapisan masukan  $X_i$  ( $i = 1, 2, \dots, n$ ) berisi nilai-nilai atau atribut-atribut atau informasi suatu pola.

Dari gambar 2.5, dapat dilihat bahwa setiap neuron pada lapisan masukan terhubung dengan setiap neuron pada lapisan tersembunyi, dan setiap neuron pada lapisan tersembunyi terhubung dengan setiap neuron pada lapisan keluaran. Karena lapisan keluaran merupakan lapisan paling kanan, maka neuron-neuronnya tidak terhubung dengan neuron apapun lagi selain neuron-neuron pada lapisan di sebelah kirinya. Setiap hubungan antar neuron ini berisi suatu nilai atau konstanta yang disebut bobot. Bobot-bobot yang menghubungkan lapisan masukan dengan lapisan tersembunyi dinotasikan dalam  $V_{ij}$  sedangkan bobot-bobot yang menghubungkan lapisan tersembunyi dengan lapisan keluaran dinotasikan dengan  $W_{jk}$ . Pada awal-awal, bobot-bobot tersebut diberi nilai acak kemudian dilanjutkan dengan metode Nguyen-Widrow. Jaringan saraf tiruan yang digunakan pada

penelitian ini adalah jaringan yang hanya memiliki sebuah lapisan tersembunyi.

Pada proses pelatihan propagasi balik, setiap vektor data melakukan proses propagasi maju, propagasi balik dan perubahan bobot,. Pada proses propagasi maju dihitung nilai  $Z_j$  ( $j = 1, 2, \dots, p$ ) menggunakan fungsi aktivasi Sigmoid biner.. Hasil penghitungan proses Sigmoid akan dikomputasi dengan bobot-bobot yang berkesesuaian dengannya ( $W_{jk}$ ) sehingga menghasilkan nilai pada neuron-neuron lapisan keluaran  $Y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan fungsi aktivasi Sigmoid biner. Proses propagasi balik adalah proses yang dilakukan setelah proses propagasi maju dalam bagian ini dihitung informasi *Error* nya ( $\delta_k$ ) yang nilainya akan digunakan dalam menghitung besarnya koreksi bobot neuron keluaran dan besarnya koreksi bias keluaran. Setelah proses propagasi maju dan propagasi balik selesai maka dihitung nilai perubahan bobot berdasarkan informasi *Error*nya. Setelah proses pelatihan kemudian dilakukan proses pengujian untuk pola yang lama atau pola yang baru. Proses yang dilakukan untuk pengenalan pola ini hanyalah propagasi maju.

Pada penelitian ini, untuk memudahkan perhitungan pada program, nilai sebuah target dinyatakan dalam matriks yang ukurannya disesuaikan dengan neuron keluaran.

Agar jaringan dapat mempelajari pola dengan baik, pola yang mempunyai target yang sama sebaiknya tidak dilatih secara berurutan, melainkan silih berganti dengan pola yang mempunyai target berbeda, misalnya pola dengan target ke-1 dilanjutkan dengan pola target ke-2, dan seterusnya. Setelah itu, barulah pelatihan dilanjutkan dengan pola dengan target ke-1 kembali

Setelah melakukan pelatihan, dilakukan pengenalan pola dengan menggunakan 30 % dari data keseluruhan, agar seragam diambil pola dengan jumlah yang sama untuk setiap target. Pada proses pengenalan pola ini, sistem akan memberi jawaban terhadap setiap pola, masuk ke *cluster* yang mana dari tiga *cluster* yang ada.

Apabila jawaban yang diberikan sesuai target, maka jaringan dianggap telah mengenali pola dengan benar. Namun bila jawaban yang diberikan tidak sesuai dengan target, maka jaringan dinilai salah dalam mengenali pola. Sistem tidak dapat menghasilkan jawaban bahwa suatu pola berada di luar target yang ditentukan. Dengan demikian, hanya ada dua pilihan, yaitu benar dan salah. Persentase jawaban yang benar terhadap seluruh pertanyaan mengacu kepada *recognition rate*.

#### **7.4. Pengujian Pada Jaringan Saraf Tiruan *Backpropagation***

Pada metode FBR Membership Function tanpa menggunakan SOM, pengujian dilakukan dengan menggunakan tiga jenis data dengan :

2. Rasio perbandingan data training dan testing 70:30
2. Data yang diuji yaitu data original dan data normalisasi

Adapun data yang dilihat sebagai hasil pengujian adalah tingkat kemampuan pengenalan pola (Recognition Rate) dan waktu yang diperlukan untuk mengenali pola tersebut.

Perbandingan data yang digunakan 70 : 30 berarti 70% data dijadikan sebagai data pembelajaran jaringan dan 30% data sisanya digunakan sebagai data pengujian.

##### **7.4.1. Hasil Pengujian dan Analisa**

Berikut ini tabel 7.1 menunjukkan hasil ujicoba untuk data original dalam mendeteksi data yang berdimensi 4, 8, dan 16 dan tabel 7.2 menunjukkan hasil ujicoba untuk data normalisasi dalam mendeteksi data yang berdimensi 8, dan 16.

Tabel 7.1 Recognition Rate Testing Data Original 1000 epoch

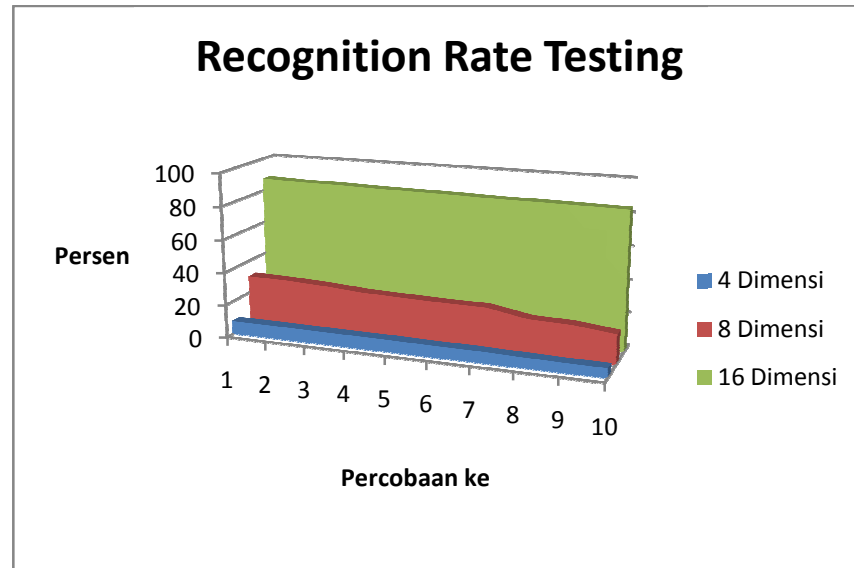
Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	7.96	18.06	84.91
2	7.13	25.93	84.26
3	7.59	21.39	86.76
4	7.78	27.13	84.63
5	7.50	20.74	85.00
6	5.83	29.91	86.11
7	7.04	25.74	86.67
8	6.48	26.48	85.56
9	7.96	28.80	87.50
10	6.02	30.28	85.83
Rata-rata	7.13	25.44	85.72
Waktu	380.63	409.16	462.44

Tabel 7.2 Recognition Rate Testing Data Normalisasi untuk 1000 epoch

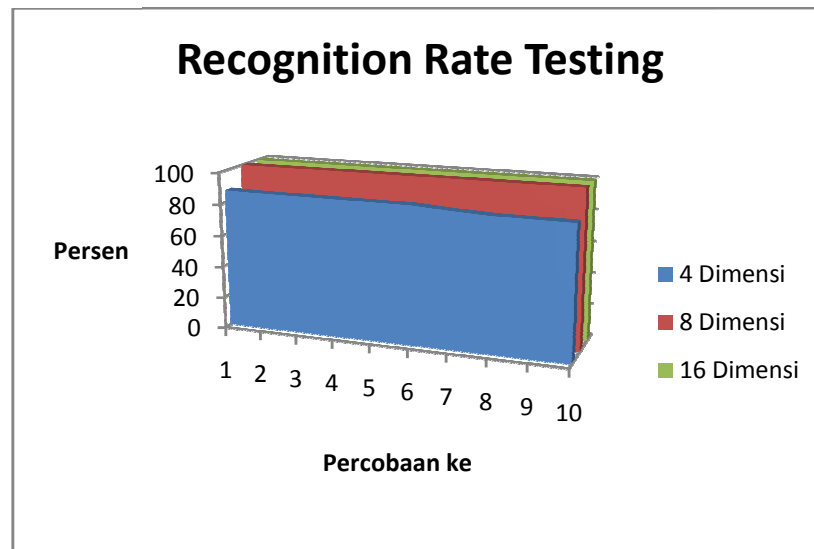
Percobaan ke	Pola		
	4 Dimensi	8 Dimensi	16 Dimensi
1	87.22	99.91	99.72
2	87.69	100.00	99.72
3	83.89	100.00	99.72
4	86.94	100.00	99.72
5	84.17	99.91	99.72
6	85.28	100.00	99.72
7	86.85	100.00	99.72
8	87.13	99.91	99.72
9	87.04	100.00	99.72
10	83.98	100.00	99.72
Rata-rata	86.02	99.97	99.72
Waktu	382.17	221.39	149.11

Untuk mempermudah proses analisa, data direpresentasikan kembali dalam bentuk grafik dapat dilihat seperti di bawah ini.





Gambar 7.1 Grafik Recognition Rate Testing Data Original dengan 1000 epoch



Gambar 7.2 Grafik Recognition Rate Testing Data Normalisasi dengan 1000 epoch

Dari gambar 7.1 dan gambar 7.2 untuk data 8 dimensi dan data 16 dimensi terlihat bahwa dalam metode *Backpropagation* data normalisasi dapat diklasifikasikan lebih baik dari pada data aslinya, hal ini dikarenakan data hasil normalisasi memiliki tingkat independensi yang lebih baik., sedangkan untuk data 4 dimensi tidak menggunakan data normalisasi karena hasil uji coba dari data original sudah menunjukkan hasil yang baik. Pada proses

Backpropagation metode pengklasifikasi sangat dipengaruhi oleh perubahan bobot.

Pada data normalisasi, komponen dari perubahan bobot, center dan spread memiliki tingkat perbedaan yang lebih jelas jika dibandingkan dengan tingkat perbedaan pada data aslinya. Sehingga proses perubahan bobot tersebut yang dihasilkan akan semakin baik, akibatnya proses klasifikasi juga akan menjadi semakin baik.

## BAB 8

### PERBANDINGAN PERFORMA JARINGAN SARAF TIRUAN

Bab ini menerangkan tentang perbandingan antara metode-metode yang telah diuji dan dianalisa pada bab sebelumnya. Performa utama yang dibandingkan adalah kemampuan pengklasifikasiannya (Recognition Rate). Selain itu juga perbandingan dalam hal waktu proses yang diperlukan masing-masing metode, hal ini penting karena pada beberapa kasus terkadang lebih memilih untuk menggunakan metode dengan waktu proses yang lebih cepat dan RR yang tidak begitu sempurna.

#### 8.1. Perbandingan Metode FBR Menggunakan SOM dan FBR Tanpa Menggunakan SOM

Sebagai bahan perbandingan berikut data ujicoba FBR menggunakan SOM dan FBR tanpa SOM dengan variasi 70:30

Tabel 8.1 Perbandingan Recognition Rate Testing FBR SOM dengan FBR tanpa SOM

Percobaan ke-	FBR SOM		FBR Tanpa SOM	
	Original	Normalisasi	Original	Normalisasi
1	0.00	39.89	0.00	50.33
2	0.00	39.89	0.00	50.33
3	0.00	36	0.00	38
4	0.00	36	0.00	38
5	0.00	36	0.00	32.33
6	0.00	34.56	0.00	32.33
7	0.00	34.56	0.00	32.33
8	0.00	34.56	0.00	28.78
9	0.00	33.78	0.00	28.78
10	0.00	33.78	0.00	27.11
Rata-rata	0.00	35.90	0.00	35.83
Waktu	6411.73	6112.1	5953.30	6218.64

Dari data pada tabel 8.1 terlihat unjuk kerja FBR menggunakan SOM lebih baik daripada dengan FBR tanpa SOM. Waktu proses untuk FBR menggunakan SOM dan FBR tanpa SOM ditentukan oleh seberapa lama

waktu yang dibutuhkan untuk mencapai kondisi konvergen, atau dengan kata lain hingga iterasi ke berapa kondisi henti terpenuhi.

Meskipun FBR menggunakan SOM memiliki algoritma yang lebih panjang daripada FBR tanpa SOM namun dalam prakteknya untuk stoping epoch yang sama (antara FBR menggunakan SOM dan FBR tanpa SOM normalisasi) waktu yang digunakan tidak jauh berbeda hal ini karena perbedaannya sangat kecil. Sehingga dalam hal ini penentuan waktu proses lebih dipengaruhi oleh banyaknya stoping epoch saja.

## 8.2. Perbandingan Metode FBR Fuzzy Menggunakan SOM dan FBR Fuzzy Tanpa Menggunakan SOM

Sebagai bahan perbandingan berikut data ujicoba FBR fuzzy menggunakan SOM dan FBR fuzzy tanpa SOM dengan variasi 70:30

Tabel 8.2 Perbandingan Recognition Rate Testing FBR Fuzzy SOM dengan FBR Fuzzy tanpa SOM

Percobaan ke-	FBR FUZZY SOM		FBR FUZZY Tanpa SOM	
	Original	Normalisasi	Original	Normalisasi
1	0.00	45.67	0.00	36.22
2	0.00	45.67	0.00	36.22
3	0.00	45.67	0.00	36.22
4	0.00	45.67	0.00	36.11
5	0.00	40.33	0.00	36.11
6	0.00	40.33	0.00	36.11
7	0.00	40.33	0.00	31.33
8	0.00	38.56	0.00	31.33
9	0.00	38.56	0.00	31.33
10	0.00	38	0.00	31.33
Rata-rata	0.00	41.88	0.00	34.23
Waktu	6497.47	6465.04	6538.09	6950.09

Dari data pada tabel 8.2 terlihat unjuk kerja FBR Fuzzy menggunakan SOM lebih baik daripada dengan FBR Fuzzy tanpa SOM, terlihat dari hasil recognition rate yang dihasil oleh metode FBR Fuzzy menggunakan SOM sebesar 41.88% sedangkan FBR Fuzzy tanpa SOM menghasilkan 34.23%.

Waktu proses untuk FBR Fuzzy menggunakan SOM dan FBR Fuzzy tanpa SOM ditentukan oleh seberapa lama waktu yang dibutuhkan untuk mencapai kondisi konvergen, atau dengan kata lain hingga iterasi ke berapa kondisi henti terpenuhi.

Meskipun FBR Fuzzy menggunakan SOM memiliki algoritma yang lebih panjang daripada FBR Fuzzy tanpa SOM namun dalam prakteknya untuk stopping epoch yang sama (antara FBR Fuzzy menggunakan SOM dan FBR Fuzzy tanpa SOM normalisasi) waktu yang digunakan tidak jauh berbeda hal ini karena perbedaannya sangat kecil. Sehingga dalam hal ini penentuan waktu proses lebih dipengaruhi oleh banyaknya stopping epoch saja.

### **8.3 Perbandingan Metode Fuzzy dan Non-Fuzzy**

Dari dua sub bab sebelumnya, secara umum dapat dilihat adanya keunggulan dan kelemahan dari sistem fuzzy (FBR Fuzzy menggunakan SOM dan FBR Fuzzy tanpa SOM) jika dibandingkan dengan sistem non fuzzy (FBR menggunakan SOM dan FBR tanpa SOM).

Keunggulan Metode Fuzzy

- Recognition rate lebih baik daripada metode biasa
- Kekurangan Metode Fuzzy
- Memerlukan waktu proses yang lebih lama dibanding dengan metode biasa.
  - Lebih kompleks dalam pengembangannya karena proses di dalam sebuah iterasi lebih panjang karena menyesuaikan dengan algoritma dan aturan-aturan khusus yang berlaku pada bilangan fuzzy.

### **8.4 Perbandingan Metode FBR Fuzzy Menggunakan SOM dan Propagasi Balik**

Sebagai bahan perbandingan berikut data ujicoba FBR fuzzy menggunakan SOM dan propagasi Balik dengan variasi 70:30

Tabel 8.3 Perbandingan Recognition Rate Testing FBR Fuzzy SOM dengan propagasi Balik

Percobaan ke-	FBR FUZZY SOM		PROPAGASI BALIK	
	Original	Normalisasi	Original	Normalisasi
1	0.00	45.67	84.91	99.72
2	0.00	45.67	84.26	99.72
3	0.00	45.67	86.76	99.72
4	0.00	45.67	84.63	99.72
5	0.00	40.33	85.00	99.72
6	0.00	40.33	86.11	99.72
7	0.00	40.33	86.67	99.72
8	0.00	38.56	85.56	99.72
9	0.00	38.56	87.50	99.72
10	0.00	38	85.83	99.72
Rata-rata	0.00	41.88	85.72	99.72
Waktu	6497.47	6465.04	462.44	149.11

Dari data pada tabel 8.3 terlihat unjuk kerja FBR Fuzzy menggunakan SOM tidak lebih baik daripada propagasi balik hal ini dapat dilihat dari recognition rate yang dihasilkan FBR Fuzzy menggunakan SOM hanya sebesar 41.88% sedangkan metode propagasi balik menghasilkan 99.72% untuk data odor normalisasi sedangkan untuk data original, FBR Fuzzy menggunakan SOM menghasilkan 0% sedangkan metode Propagasi balik menghasilkan 85.72%, hal ini dikarenakan metode FBR Fuzzy SOM, data original yang digunakan mempunyai tingkat perbedaan data tiap dimensi tidak jelas atau dalam arti data setiap dimensinya mendekati sama, epoch yang digunakan masih sangat sedikit, dan parameter-parameter yang digunakan belum optimal. Waktu proses untuk FBR Fuzzy menggunakan SOM dan propagasi balik ditentukan oleh seberapa lama waktu yang dibutuhkan untuk mencapai kondisi konvergen, atau dengan kata lain hingga iterasi ke berapa kondisi henti terpenuhi. Dari tabel 8.3 dapat dilihat bahwa waktu yang dibutuhkan pada jaringan saraf tiruan FBR Fuzzy menggunakan SOM lebih lama dibandingkan pada jaringan saraf tiruan propagasi balik, hal ini dikarenakan adanya penambahan algoritma yang

panjang pada metode propagasi balik dalam metode jaringan saraf tiruan FBR Fuzzy.

## **BAB 9**

### **KESIMPULAN DAN SARAN**

#### **9.1. Kesimpulan**

Setelah melakukan serangkaian penelitian yang dimulai dari studi literatur, pengembangan program dan pengujian serta analisa hasil pengujian maka ada beberapa hal yang dapat disimpulkan yaitu :

1. Jaringan saraf tiruan FBR Fuzzy dapat diterapkan dalam sistem pengenalan pola, namun sangat dipengaruhi oleh parameter-parameter yang digunakan antara lain :
  - a. Banyaknya epoch yang digunakan.
  - b. Laju pembelajaran SOM, laju pembelajaran Propagasi balik, laju pembelajaran center dan laju pembelajaran spread.
  - c. Tingkat perbedaan data pada masing-masing dimensi. Jika tingkat perbedaan data semakin besar , maka recognition rate yang dihasilkan akan semakin bagus dan sebaliknya.
2. Jaringan saraf tiruan FBR Fuzzy dapat menghasilkan recognition rate yang lebih bagus dari jaringan saraf tiruan FBR dan Jaringan saraf tiruan propagasi balik, namun waktu yang dibutuhkan masih sangat lama.
3. Logika fuzzy terbukti bisa diaplikasikan dalam jaringan syaraf tiruan FBR Fuzzy

#### **9.2. Saran**

Saran ke depan dapat dilakukan dengan mencari parameter-parameter yang optimal terlebih dahulu untuk setiap data yang berbeda agar didapatkan pula hasil yang optimal, sehingga metode ini bisa digunakan oleh semua jenis data yang berbeda



## DAFTAR PUSTAKA

1. Mauridhi Hery Purnomo (2002), “Dasar Algoritma Cerdas“, PENS-ITS.
2. Benyamin Kusumoputro, “Jaringan Neural Network”, (Depok: Universitas Indonesia), hal 3-4.
3. Jong Jek Siang, (2005)“Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab”, ANDI, Yogyakarta.
4. Ibid, hal 109.
5. Dae Yup Lee;,Byung Man Kim, Hyung Suck Cho,(1999),” A self-organizing radial basis function network combined with ART II”, Neural Networks, 1999. IJCNN '99. International Joint Conference on, Volume: 3
6. Jianming Lian, Yonggon Lee,Sudhoff, S.D, Zak, S.H.(2008),”Self-Organizing Radial Basis Function Network For Real-Time Approximation of Continuous Time Dynamical Systems”, Neural Networks, IEEE Transactions on, Volume: 19
7. Wahyudi, Hariyanto dan Iwan Setiawan, (2007), “Aplikasi jaringan Saraf Tiruan RBF pada Sistem Kontrol Valve untuk Pengendalian Tinggi Muka Air”,SNATI, Yogyakarta.
8. Brodjol Sutijo, Subanar dan Suryo Guritno, (2006), “Pemilihan Hubungan Input-Node Pada Jaringan Saraf ungsi Radial Basis”, Berkala MIPA
9. Bastos,L.C,Bastos,R.C,Nishida,W,(1999),”Radial basis Function For Classification of remote Sensing Images", Neural Networks, 1999. IJCNN '99. International Joint Conference on, Volume:3.
10. Puneet Singla, Subbarao, K., Junkins, J.L., (2007), “Direction-Dependent Learning Approach For Radial Basis Function networks”, Neural Networks,IEEETransactionson,Volume:18.
11. Subuh Isnur Haryudo, Adi Soeprijanto dan Mauridhi Hery Purnomo, (2005),” Penggunaan Radial Basis Function (RBF) Pada Generator Tunggal Untuk Optimasi Kinerja Power System Stabilizer”, SNATI, Yogyakarta