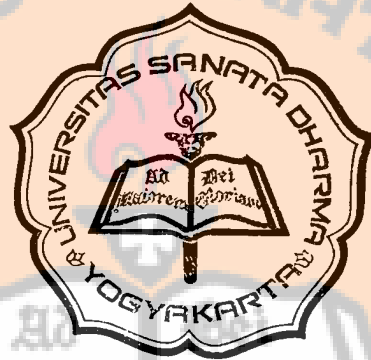


**Pengenalan Pola Angka 0 sampai 9
Menggunakan Jaringan Syaraf Tiruan dengan
Model Jaringan Kohonen**
SKRIPSI

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains (S.Si)
Program Studi Ilmu Komputer



Oleh :

Clara Wastiunamsih

NIM : 033124002

PROGRAM STUDI ILMU KOMPUTER

JURUSAN MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

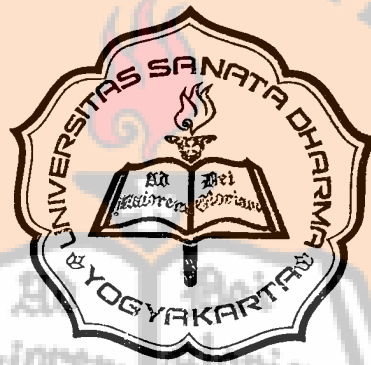
UNIVERSITAS SANATA DHARMA

YOGYAKARTA

2007

**THE PATTERN RECOGNITION OF NUMBER 0 TO 9
USING THE KOHONEN NETWORK MODEL OF
ARTIFICIAL NEURAL NETWORK
A THESIS**

Presented as Partial Fullfilment of the Requirements
To Obtain *Sarjana Sains* (S.Si) Degree
Computer Scince Study Program



By :
Clara Wastiunamsih
Student ID : 033124002

**COMPUTER SCIENCE STUDY PROGRAM
MATHEMATICS DEPARTMENT
SAINS AND TEKNOLOGY FACULTY
SANATA DHARMA UNIVERSITY**

YOGYAKARTA

2007

SKRIPSI

**Pengenalan Pola Angka 0 sampai 9
Menggunakan Jaringan Syaraf Tiruan Dengan
Model Jaringan Kohonen**

Oleh:

Clara Wastiunamsih

NIM : 033124002

Telah disetujui Oleh :

Dosen Pembimbing



Yanuarius Joko Nugroho, S.Si

Tanggal : 29 Agustus 2007

SKRIPSI

**Pengenalan Pola Angka 0 sampai 9
Menggunakan Jaringan Syaraf Tiruan Dengan
Model Jaringan Kohonen**

Dipersiapkan dan ditulis oleh

Clara Wastiunamsih

NIM : 033124002

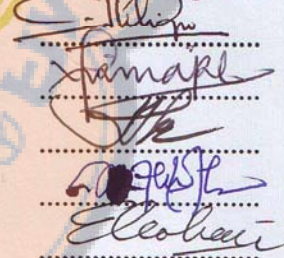
Telah dipertahankan di depan Panitia Penguji
pada tanggal 29 Agustus 2007
dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Nama Lengkap

Ketua : Ir. Greg. Heliarko, S.J., S.S., B.S.T., M.Sc., M.A.
Sekretaris : P.H. Prima Rosa, S.Si., M.Sc
Anggota : Yanuarius Joko Nugroho, S.Si
Anggota : Anastasia Rita Widiarti, S.Si., M.Kom.
Anggota : St. Eko Hari Parmadi, S.Si., M.Kom.

Tanda tangan

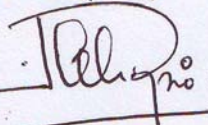


Yogyakarta, 29 Agustus 2007


Fakultas Sains dan Teknologi

Universitas Sanata Dharma

Dekan,



(Ir. Greg. Heliarko, S.J., S.S., B.S.T., M.Sc., M.A.)



"Pagawean senjata sepele
tanpa aji, yen katindakake
kanthi katresnan lan saapik-apike
dadi dalaning kasucian."

mbah Martowiratmo

Skripsi ini Kupersembahkan kepada:

- ♥ Sahabat sejatiku dalam setiap tapak hidupku, Yesus Sang Sumber Pengharapan Sejati.
- ♥ Kakekku, mbah Martowiratmo dengan semangat hidup tak berkesudahan.
- ♥ Bapak, Ibu, dan Prisca yang slalu mendukungku.
- ♥ Silas, adik yang penuh inspirasi.
- ♥ Yulius Wahyu Jatmiko yang slalu mendampingiku dalam suka dan duka dengan kasih cintanya.
- ♥ Semua orang yang peduli terhadap ilmu pengetahuan dan slalu berjuang untuk menggalinya lebih dalam.

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Yogyakarta, 29 Agustus 2007

Penulis

Clara Wastiunamsih



PENGENALAN POLA ANGKA 0 SAMPAI 9 MENGUNAKAN JARINGAN SYARAF TIRUAN DENGAN MODEL JARINGAN KOHONEN

INTISARI

Mengenali pola sebuah angka bukan menjadi hal yang sulit bagi manusia, akan tetapi tidak demikian halnya dengan komputer. Komputer harus memiliki algoritma atau cara tersendiri untuk dapat mengenali pola angka.

Jaringan Syaraf Tiruan dengan model Jaringan Kohonen memiliki kemampuan untuk mengelompokkan pola yang mirip ke dalam satu kelompok. Kemampuan tersebut dapat digunakan untuk menyelesaikan masalah pengenalan pola, misalnya pengenalan pola angka 0 sampai 9.

Program pengenalan pola angka 0 sampai 9 telah berhasil diimplementasikan menggunakan *software Matlab 6.5.1*. Berdasarkan hasil penelitian dan percobaan yang telah dilakukan dapat diperoleh kesimpulan bahwa pengenalan pola angka 0 sampai 9 menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen memiliki tingkat keberhasilan 100%.

Kata kunci : Pengenalan Pola, Jaringan Syaraf Tiruan, Kohonen.

THE PATTERN RECOGNITION OF NUMBER 0 TO 9 USING THE KOHONEN NETWORK MODEL OF ARTIFICIAL NEURAL NETWORK

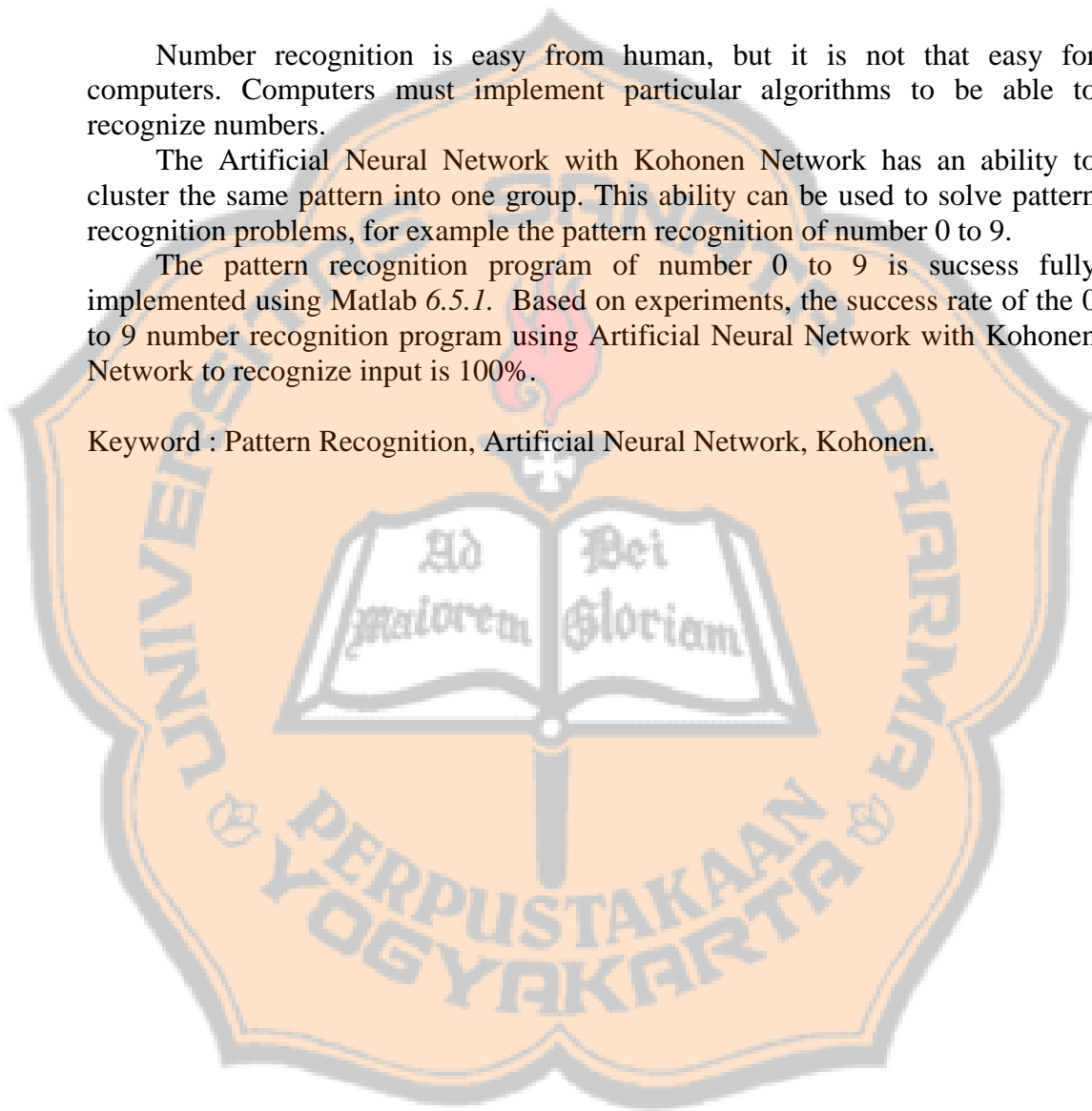
ABSTRACT

Number recognition is easy from human, but it is not that easy for computers. Computers must implement particular algorithms to be able to recognize numbers.

The Artificial Neural Network with Kohonen Network has an ability to cluster the same pattern into one group. This ability can be used to solve pattern recognition problems, for example the pattern recognition of number 0 to 9.

The pattern recognition program of number 0 to 9 is success fully implemented using Matlab 6.5.1. Based on experiments, the success rate of the 0 to 9 number recognition program using Artificial Neural Network with Kohonen Network to recognize input is 100%.

Keyword : Pattern Recognition, Artificial Neural Network, Kohonen.



KATA PENGANTAR

Puji syukur kepada Allah Bapa yang maha baik sehingga penulis dapat menyelesaikan skripsi ini dengan lancar. Atas segala kekuatan dan kesehatan yang diberikan-Nya, baik sebelum ataupun selama penulis mengerjakan skripsi ini.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains (S.Si) Program Studi Ilmu Komputer di Fakultas Sains dan Teknologi Universitas Sanata Dharma Yogyakarta.

Penulis menyadari banyak pihak yang telah memberikan sumbangan baik pikiran, waktu, tenaga, bimbingan dan dorongan pada penulis sehingga akhirnya skripsi ini dapat selesai. Oleh karena itu pada kesempatan ini, tidak lupa penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

- ✚ Pak Yanuarius Joko Nugroho, S.Si selaku dosen pembimbing penulisan tugas akhir ini. Terimakasih atas bimbingan, semangat, dan inspirasi yang diberikan kepada penulis dalam penulisan skripsi ini.
- ✚ Bapak St. Eko Hari Parmadi, S.Si., M.Kom dan Ibu Rita Widiarti, S.Si., M.Kom selaku dosen penguji yang telah memberikan, masukan, saran dan kritik yang membangun untuk skripsi ini.
- ✚ Pak JJ Siang yang telah mengenalkan penulis pada Jaringan Syaraf Tiruan.
- ✚ Bapak Haris Sriwindono selaku dosen pembimbing akademik.
- ✚ Ibu P.H. Prima Rosa, M.Sc. selaku Kepala Program Studi Ilmu Komputer atas kesabaran dan bimbingannya.
- ✚ Romo Gregorius Heliarko selaku Dekan Fakultas Sains dan Teknologi.

- ✚ Seluruh dosen-dosen di Fakultas Sains dan Teknologi. Terima kasih atas semua bimbingan, kesabaran, pengetahuan dan kesempatan yang penulis terima sejak menjadi mahasiswi di Fakultas Sains dan Teknologi.
- ✚ Pejabat Struktural di Fakultas MIPA. Terima kasih atas segala kemudahan dan fasilitas yang diberikan kepada penulis selama belajar di Fakultas MIPA.
- ✚ Seluruh Karyawan Fakultas MIPA. Terima kasih atas kemudahan-kemudahan yang selama ini saya terima.
- ✚ Temanku belajar mengarungi kehidupan, Yulius Wahyu Jatmiko, ST yang kucintai. Hidup ini jadi penuh makna.
- ✚ Inspirasi dan penyemangat hidup tiada henti bagiku : Silas (*sang Galileo*), Prisca, Bapak Purwatmo dengan semangat yang terus menyala, Ibu Jumarsih, Mbah Martowiratmo dengan Doa dan berkatnya untuk Penulis, serta semua orang yang dikasihi dan mengasihi Penulis.
- ✚ Sahabat-sahabatku : Anjar (teman special yang slalu memberiku semangat), Hosea, Vitri dan Kadek dengan kebersamaan yang menyenangkan, Aris (cerita-cerita kita kan abadi), mbak Nana (impianmu kan jadi nyata, *I believe in you*), Yosi, Iie, mbak Utin (*I miss u*), mbak Sisca, mbak Yanti, mbak Ferli, kak Elen, mbak Wulan, mbak Erni, mbak Nunuk, mbak Rita, mbak Yeni, dek Ana kecil, teman-teman KKN, dan temen-temen Mudika Ganjuran.
- ✚ Teman-teman Ilmu Komputer angkatan 2003. Terima kasih atas kebersamaannya selama ini, kalian membuat hidupku lebih indah.
- ✚ Teman masa kecilku : Om Ari, Om Andi, mbak Kristin (percayalah bahwa hidup ini indah), mas Andre, mas Duwix, mas Antok, dan *olimpus*.

✚ Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Walaupun demikian penulis berharap semoga skripsi ini dapat bermanfaat bagi masyarakat dan perkembangan ilmu pengetahuan alam.

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERSEMBAHAN	v
HALAMAN KEASLIAN	vi
INTISARI.....	vii
ABSTRACT.....	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xix
DAFTAR LAMPIRAN.....	xx
BAB I. PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan Penulisan.....	3
1.5. Metodologi	4
1.6. Sistematika Pembahasan.....	5

BAB II. LANDASAN TEORI

2.1. Pengenalan Pola7

2.2. Jaringan Syaraf Biologi.....10

2.3. Jaringan Syaraf Tiruan11

 2.3.1. Arsitektur Jaringan13

 2.3.2. Metode Pelatihan.....15

 2.3.3. Fungsi Aktivasi.....15

2.4. Pelatihan *Supervised* dan *Unsupervised*.....16

2.5. Jaringan Kohonen17

 2.5.1. Arsitektur Jaringan17

 2.5.2. Algoritma Jaringan Kohonen19

BAB III. ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisa Sistem.....21

3.2. Perancangan23

 3.2.1. Perancangan Proses Pengenalan Pola.....23

 3.2.2. Perancangan Menu38

 3.2.3. Perancangan *Interface* Sistem Pengenalan Pola.....41

BAB IV. IMPLEMENTASI DAN PEMBAHASAN

4.1. Implementasi Proses Pengenalan Pola51

 4.2.1. Proses Memasukkan Data Pola Angka51

 4.2.2. Proses Menampilkan Data Pola Angka.....52

 4.2.3. Proses Menghapus Data Pola Angka52

 4.2.4. *Preprocessing*.....53

4.2.5. Pelatihan.....	55
4.2.6. <i>Clustering</i>	68
4.2.7. Pengujian.....	70
4.2. Implentasi Interface Sistem Pengenalan Pola	71
4.2.1. Form <i>enter</i>	71
4.2.2. Form Home	72
4.2.3. Form Input Pola	73
4.2.4. Form Tampil Pola	74
4.2.5. Form Edit Pola	75
4.2.6. Form Pelatihan	76
4.2.7. Form Hasil Pelatihan.....	79
4.2.8. Form Hasil Pengujian.....	80
4.2.9. Form ProgramHelp	81
4.2.10. Form About	81
4.3. Hasil Percobaan	83
4.4. Pembahasan.....	96
BAB V. KESIMPULAN DAN SARAN	
5.1. Kesimpulan	99
5.2. Saran.....	100
 DAFTAR PUSTAKA	 xxi

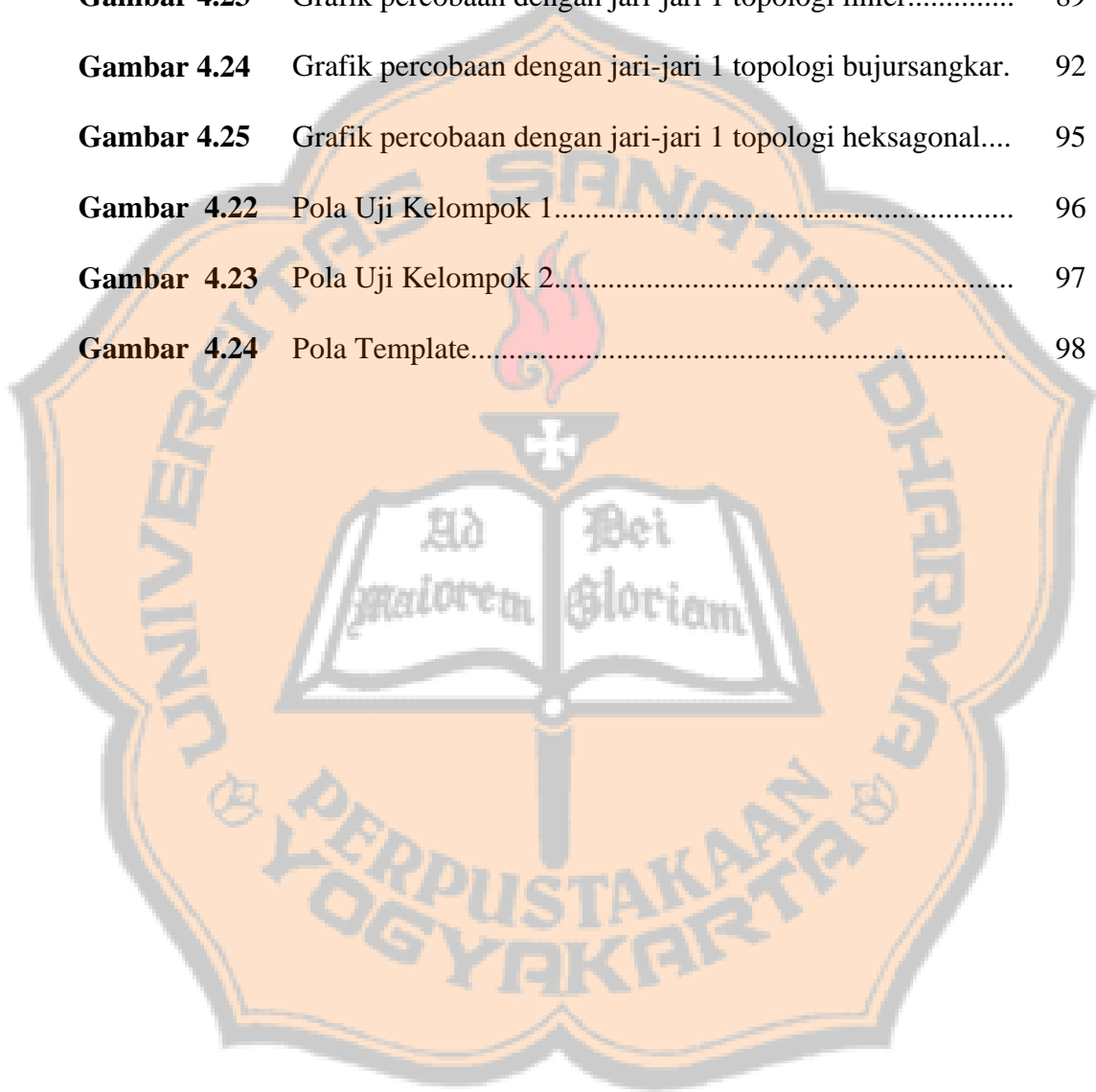
DAFTAR GAMBAR

Gambar 2.1	Grafik fungsi kerapatan dari ciri diameter jeruk dan apel....	8
Gambar 2.2	Grammar penyusun gambar rumah yang merupakan penggabungan dari <i>Primitif</i> penyusunnya.....	9
Gambar 2.3	Neuron.....	11
Gambar 2.4	Jaringan Syaraf Tiruan terdiri dari input (x_1 , x_2 , dan x_3), bobot masing-masing input (w_1 , w_2 , dan w_3), dan keluaran (y).....	12
Gambar 2.5	Jaringan Layer Tunggal.....	13
Gambar 2.6	Jaringan Layer Jamak.....	14
Gambar 2.7	Jaringan Reccurent Sederhana.....	14
Gambar 2.8	Representasi satu dimensi (linear) Model Jaringan Kohonen Gambar di atas menunjukkan kasus untuk vector w berupa vector satu dimensi (dengan jarak $R=2$).....	18
Gambar 2.9	Representasi 2 dimensi Model Jaringan Kohonen dengan $R = 1$ dan $R = 2$	19
Gambar 3.1	Flowchart proses pelatihan.....	26
Gambar 3.2	Arsitektur Jaringan Kohonen Topologi Linear.....	29
Gambar 3.3	Matriks Bobot.....	30
Gambar 3.4	Kelas yang dimodifikasi pada Topologi Linear dengan jari-jari 1.....	30
Gambar 3.5	Baris Vektor Bobot yang dimodifikasi.....	31
Gambar 3.6	Arsitektur Jaringan Kohonen Topologi Bujursangkar.....	32

Gambar 3.7	Matriks Bobot.....	32
Gambar 3.8	Kelas yang dimodifikasi pada Topologi Bujursangkar dengan jari-jari 1.....	33
Gambar 3.9	Baris vektor bobot yang dimodifikasi.....	33
Gambar 3.10	Arsitektur Jaringan Kohonen Topologi Heksagonal.....	34
Gambar 3.11	Matriks Bobot.....	35
Gambar 3.12	Kelas yang dimodifikasi pada Topologi Heksagonal dengan jari-jari 1.....	35
Gambar 3.13	Baris Vektor Bobot yang dimodifikasi.....	36
Gambar 3.14	Menu File beserta submenu di dalamnya.....	38
Gambar 3.15	Menu View.....	39
Gambar 3.16	Menu Edit beserta submenu di dalamnya.....	39
Gambar 3.17	Menu Run beserta submenu di dalamnya.....	40
Gambar 3.18	Menu Help beserta submenu di dalamnya.....	40
Gambar 3.19	Form Enter.....	41
Gambar 3.20	Form Home.....	42
Gambar 3.21	Form Input Pola.....	42
Gambar 3.22	Form Tampil Pola.....	43
Gambar 3.23	Waitbar.....	44
Gambar 3.24	Form Edit Pola.....	44
Gambar 3.25	Form Pelatihan.....	45
Gambar 3.26	Form Hasil Pelatihan.....	47
Gambar 3.27	Form Hasil Pengujian.....	48

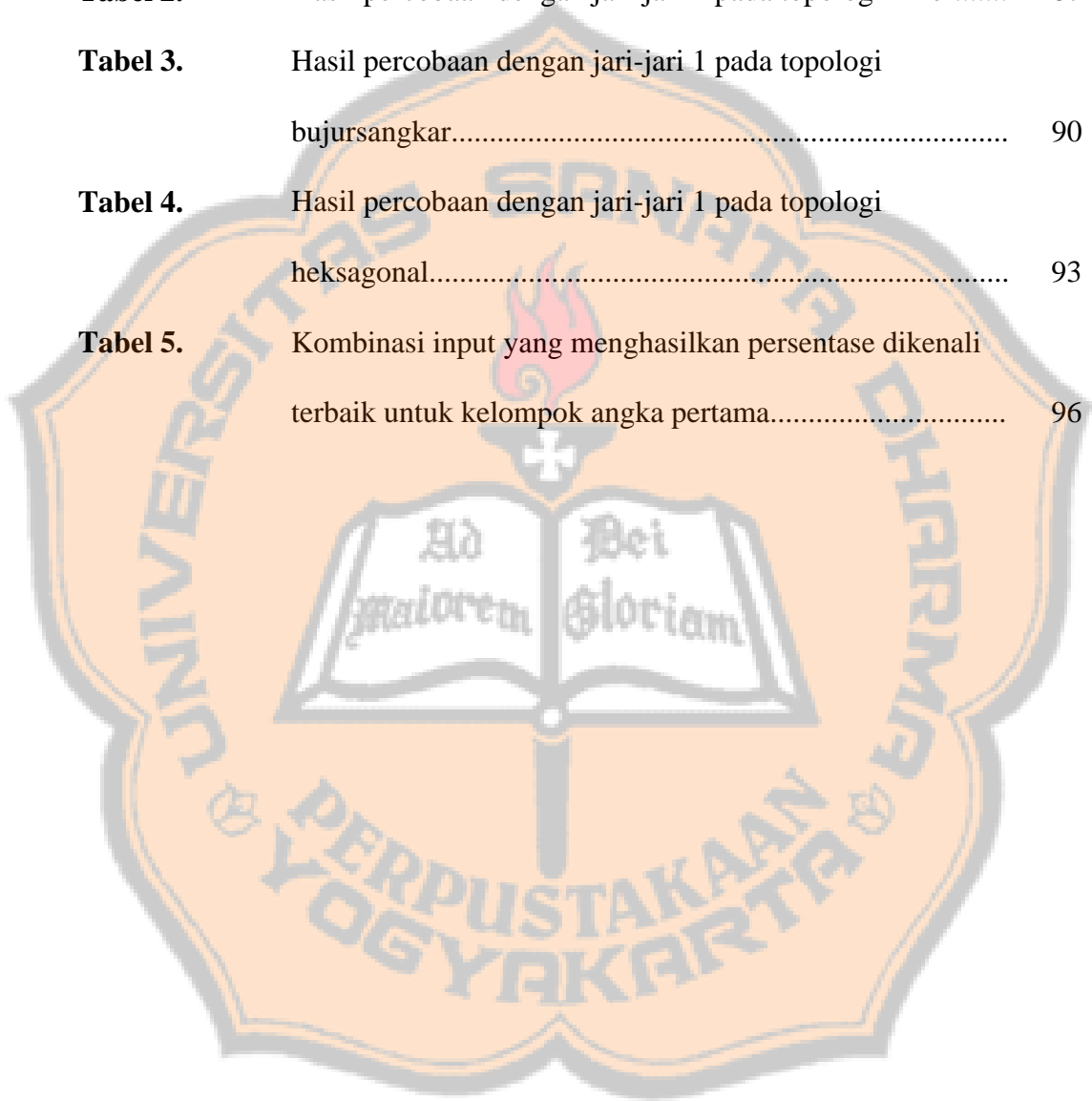
Gambar 3.28	Form ProgramHelp.....	49
Gambar 3.29	Form About.....	50
Gambar 4.1	Flowchart proses pelatihan.....	56
Gambar 4.2	Topologi linear.....	62
Gambar 4.3	Matriks pembantu pada topologi bujursangkar.....	63
Gambar 4.4	Susunan kelompok pada topologi Heksagonal.....	64
Gambar 4.5	Arsitektur jaringan kohonen topologi heksagonal.....	64
Gambar 4.6	Matriks bobot.....	65
Gambar 4.7	Kelas yang dimodifikasi pada topologi heksagonal dengan jari-jari 1.....	65
Gambar 4.8	Baris vektor bobot yang dimodifikasi.....	66
Gambar 4.9	Form enter.....	71
Gambar 4.10	Form home.....	72
Gambar 4.11	Form Input Pola.....	73
Gambar 4.12	Form Tampil Pola.....	74
Gambar 4.13	Waitbar.....	74
Gambar 4.14	Form Edit Pola.....	75
Gambar 4.15	Form Pelatihan.....	76
Gambar 4.16	Form Pelatihan.....	77
Gambar 4.17	Form Pelatihan.....	77
Gambar 4.18	Form Hasil Pelatihan.....	79
Gambar 4.19	Form Hasil Pengujian.....	80

Gambar 4.20	Form ProgramHelp.....	81
Gambar 4.21	Form About.....	82
Gambar 4.22	Grafik percobaan dengan jari-jari 0 semua topologi.....	86
Gambar 4.23	Grafik percobaan dengan jari-jari 1 topologi linier.....	89
Gambar 4.24	Grafik percobaan dengan jari-jari 1 topologi bujursangkar.	92
Gambar 4.25	Grafik percobaan dengan jari-jari 1 topologi heksagonal....	95
Gambar 4.22	Pola Uji Kelompok 1.....	96
Gambar 4.23	Pola Uji Kelompok 2.....	97
Gambar 4.24	Pola Template.....	98



DAFTAR TABEL

Tabel 1.	Hasil percobaan dengan jari-jari 0 semua topologi.....	84
Tabel 2.	Hasil percobaan dengan jari-jari 1 pada topologi linier.....	87
Tabel 3.	Hasil percobaan dengan jari-jari 1 pada topologi bujursangkar.....	90
Tabel 4.	Hasil percobaan dengan jari-jari 1 pada topologi heksagonal.....	93
Tabel 5.	Kombinasi input yang menghasilkan persentase dikenali terbaik untuk kelompok angka pertama.....	96



DAFTAR LAMPIRAN

Lampiran 1 Hasil Percobaan Dengan Pola Uji Kelompok 2



BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Mengenali pola angka tulisan tangan bukan menjadi hal yang sulit bagi manusia, akan tetapi berbeda halnya dengan komputer. Komputer harus memiliki algoritma atau cara tersendiri untuk dapat mengenali pola angka.

Kemampuan komputer dalam mengenali suatu pola mulai banyak dibutuhkan untuk menyelesaikan berbagai persoalan, contohnya pengenalan pola sidik jari pada bidang keamanan informasi, pengenalan pola *Kardiograf* pada bidang kedokteran, pengenalan pola wajah, dan pengenalan pola karakter.

Metode pengenalan pola dilakukan dengan beberapa pendekatan, antara lain *Geometric / Statistical Approach*, *Sruktural / Syntactic Approach*, dan *Computational Intelligent Approach*. *Computational Intelligent Approach* terdiri dari *Fuzzy Logic Approach*, dan *Neural Network Approach* (Jaringan Syaraf Tiruan).

Jaringan Syaraf Tiruan adalah salah satu cabang ilmu dari bidang ilmu Kecerdasan Buatan (*Artificial Intelligence*) dan merupakan cara untuk memecahkan masalah terutama di bidang pengelompokan dan pengenalan pola. Pada pengenalan pola angka 0 sampai 9, dibutuhkan sebuah metode yang dapat mengelompokkan angka masukan ke dalam kelompok angka yang mirip satu sama lain. Metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen dapat

bekerja seperti cara kerja otak manusia yang dapat secara otomatis mengelompokkan angka berdasarkan bentuknya.

Pola angka 0 sampai 9 adalah penyederhanaan dari pola karakter. Pola tersebut akan digunakan untuk menguji sejauh mana metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen dapat digunakan untuk pengenalan pola.

1.2. RUMUSAN MASALAH

Masalah yang akan diselesaikan di dalam penelitian ini adalah :

1. Bagaimana cara kerja pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen?
2. Bagaimana mengimplementasikan pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen dalam program?
3. Bagaimana unjuk kerja sistem pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen?

1.3. BATASAN MASALAH

Permasalahan yang akan dipecahkan pada penelitian ini akan dibatasi oleh beberapa hal sebagai berikut :

1. Pola sederhana yang akan digunakan sebagai input adalah tulisan tangan angka 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9.

2. Input berupa file dengan format grafik BMP-1 bit (*Biner*). File grafik tersebut adalah hasil *Scan* dari tulisan angka yang dituliskan pada kertas dengan area penulisan masing-masing angka dibatasi pada area ukuran 3 x 3 cm, memiliki posisi tegak yang ditulis dengan spidol Snowman warna hitam ukuran sedang.
3. Input Pola *Template* ada 50 buah, terdiri dari semua jenis angka yang telah disebutkan pada batasan nomor 1 dengan masing-masing jenis 5 buah. Sedangkan untuk pola uji 20 buah yang terdiri dari dua kelompok angka 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9. Jadi, jumlah keseluruhan pola yang digunakan dalam penelitian ini ada 70 buah pola.
4. Pola input akan disamakan ukurannya yaitu 30 X 30 pixel.
5. Metode yang digunakan untuk menyelesaikan permasalahan pengenalan pola sederhana tersebut adalah metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.
6. Bahasa pemrograman yang digunakan untuk pembentukan aplikasi sebagai representasi dari permasalahan yang akan diteliti menggunakan *Software* Matlab 6.5.1.

1.4. TUJUAN PENULISAN

Tujuan dari penelitian yang dilakukan dalam tugas akhir ini adalah :

1. Mempelajari cara kerja pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

2. Membuat sistem pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.
3. Mengetahui unjuk kerja metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen yang digunakan untuk pengenalan pola angka 0 sampai 9.

1.5. METODOLOGI

Metode dalam pengembangan system menggunakan Paradigma *Waterfall* :

1. Tahap Rekayasa Sistem dan Analisis

Pada tahap ini dilakukan perincian kebutuhan, diagnosa masalah, dan penentuan tujuan pembuatan sistem pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

2. Tahap Perancangan

Pada tahap ini akan dilakukan spesifikasi yang lebih detail dari komponen-komponen sistem, struktur, dan *Feature* sistem.

3. Tahap Penulisan Program

Pada tahap ini dilakukan pengkodean dari hasil perancangan pada tahap sebelumnya menjadi bentuk yang bisa dibaca oleh mesin.

Pengkodean yang dilakukan pada tugas akhir ini adalah menggunakan *Software* Matlab 6.5.1.

4. Tahap Pengujian

Pada tahap ini dilakukan analisis program dan hasil percobaan yang dilakukan. Hasil percobaan pada tugas akhir ini akan digunakan untuk mengetahui tingkat keberhasilan pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

5. Tahap Revisi

Pada tahap ini dilakukan pengecekan sistem agar sistem selalu pada kondisi yang seharusnya dan selalu dapat beroperasi dengan baik. Pada pengembangan sistem pengenalan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen ini hanya dibatasi sampai pada tahap pengujian.

1.6. SISTEMATIKA PEMBAHASAN

BAB I PENDAHULUAN

Bab ini berisi gambaran umum yang menjelaskan latar belakang penulisan, rumusan masalah, tujuan penulisan, metodologi, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini akan menjelaskan dasar teori dalam pembuatan *Software*, yang meliputi penjelasan tentang Pengenalan Pola, Jaringan Syaraf Tiruan, dan Model Jaringan Kohonen.

BAB III ANALISIS DAN PERANCANGAN SISTEM

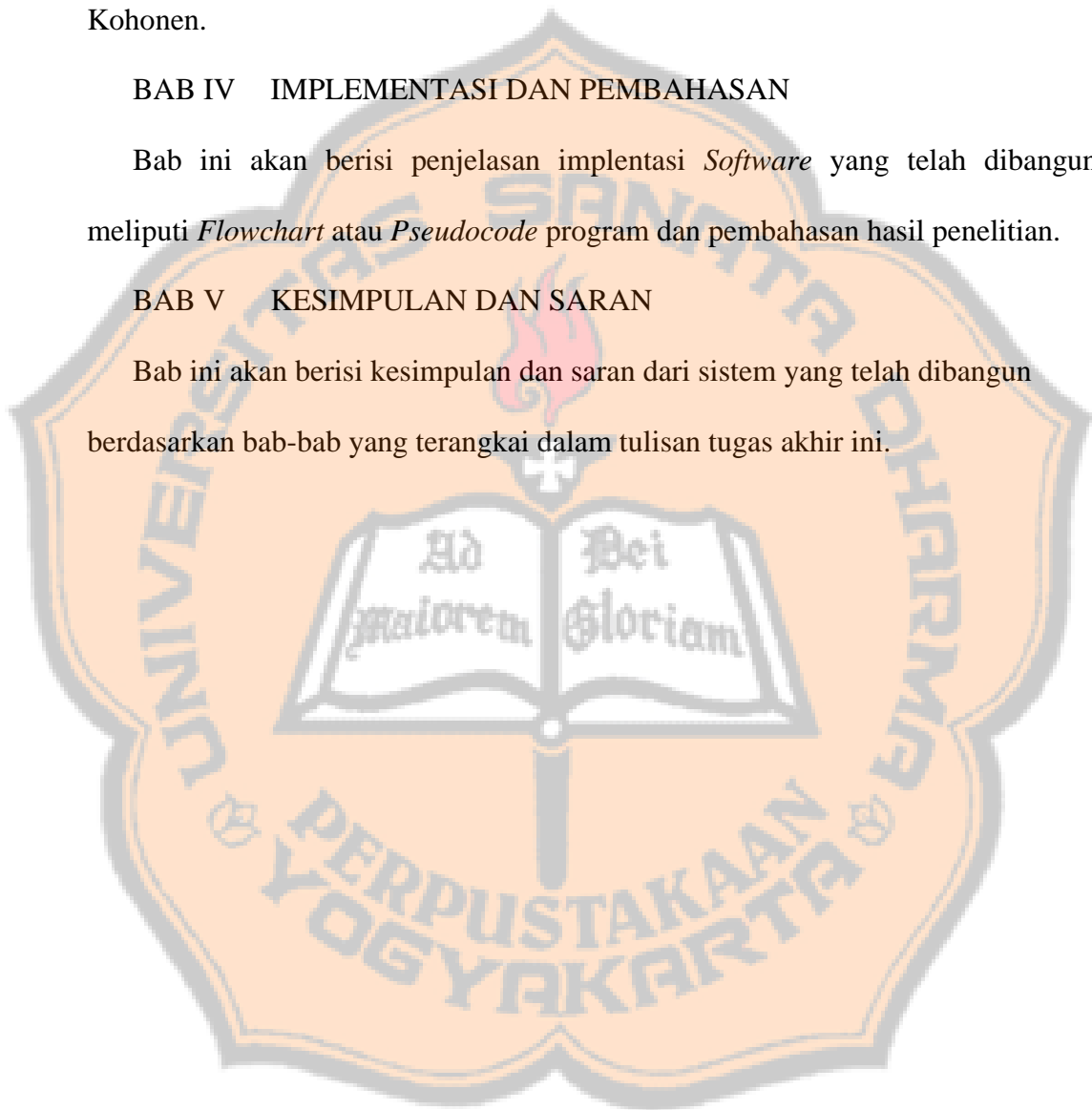
Bab ini akan berisi analisis dan perancangan desain *Software* Pengenalan Pola Angka 0 sampai 9 menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini akan berisi penjelasan implentasi *Software* yang telah dibangun meliputi *Flowchart* atau *Pseudocode* program dan pembahasan hasil penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini akan berisi kesimpulan dan saran dari sistem yang telah dibangun berdasarkan bab-bab yang terangkai dalam tulisan tugas akhir ini.



BAB II

LANDASAN TEORI

2.1. Pengenalan Pola

Pola adalah entitas yang dapat terdefinisi dan dapat diidentifikasi melalui ciri-cirinya¹. Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Pengenalan pola bertujuan untuk menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut.

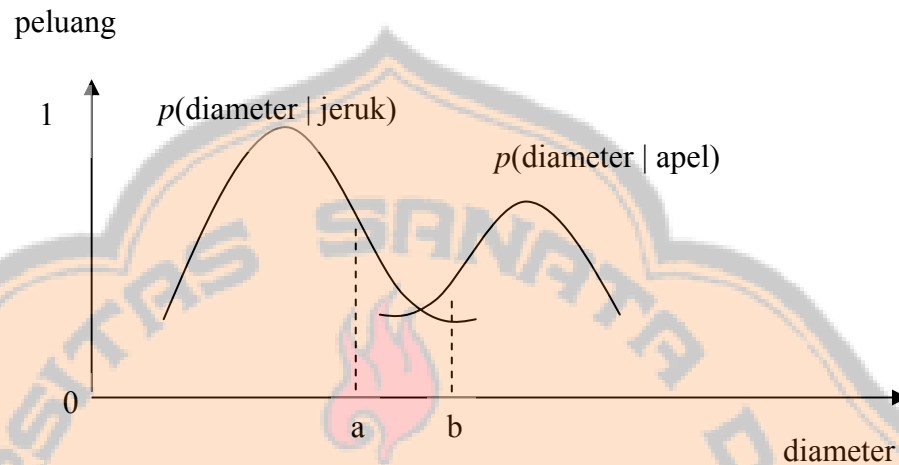
Ada dua fase dalam sistem pengenalan pola, yaitu fase pelatihan dan fase pengenalan. Pada fase pelatihan, beberapa contoh pola dipelajari untuk menentukan ciri yang akan digunakan dalam proses pengenalan serta prosedur klasifikasinya. Pada fase pengenalan, pola diambil cirinya kemudian ditentukan kelas kelompoknya.

Metode pengenalan pola dilakukan dengan beberapa pendekatan, yaitu pendekatan *Geometric / Statistik*, *Sruktural / Sintaktik*, dan pendekatan *Computational Intelligent*. Pada pendekatan *Computational Intelligent* terbagi menjadi 2 bagian yaitu dengan menggunakan pendekatan Logika Kabur dan Jaringan Syaraf Tiruan.

Pada pengenalan pola dengan Pendekatan Statistik digunakan teori-teori ilmu peluang dan statistik. Ciri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya. Pola yang berbeda memiliki distribusi yang berbeda pula. Dengan menggunakan teori keputusan di dalam Pendekatan Statistik digunakan distribusi

¹ Munir, Rinaldi. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*; hal 241-248

ciri untuk mengklasifikasikan pola. Sebagai contoh, diketahui fungsi kerapatan dari diameter buah jeruk dan apel dengan grafik seperti Gambar 2.1.



Gambar 2.1 Grafik fungsi kerapatan dari ciri diameter jeruk dan apel

Apabila diperoleh hasil pengukuran diameter adalah a cm, maka objek diklasifikasikan sebagai “jeruk”, karena $p(a \mid \text{Jeruk}) > p(a \mid \text{Apel})$.

Sedangkan pada pengenalan pola menggunakan Pendekatan *Sintaktik* digunakan teori bahasa formal. Ciri-ciri yang terdapat pada suatu pola ditentukan *Primitif* dan hubungan struktural antara *Primitif*-nya kemudian disusun tata bahasanya. Aturan produksi pada tata bahasa tersebut digunakan untuk menentukan kelompok pola.

Contoh pengenalan pola dengan pendekatan *Sintaktik* adalah pengenalan bentuk “rumah”. Tata bahasa yang digunakan untuk mengenali pola “rumah” adalah $G=(N,\Sigma,P,S)$, dengan :

$$\Sigma = \{a \nearrow, b \searrow, c \rightarrow, d \downarrow\}$$

$$N = \{S, A_1, A_2, A_3, A_4, A_5\}$$

$$S = \{S\}$$

Himpunan aturan produksi P :

$$S \rightarrow d + A_1$$

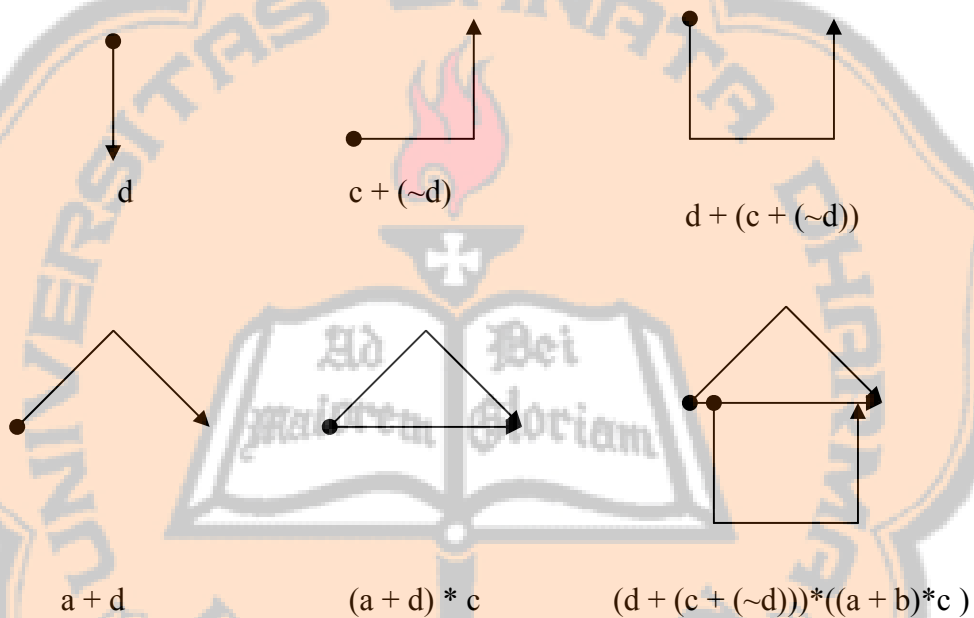
$$A_3 \rightarrow a + A_4$$

$$A_1 \rightarrow c + A_2$$

$$A_4 \rightarrow b * A_5$$

$$A_2 \rightarrow \sim d * A_2$$

$$A_5 \rightarrow c$$



Gambar 2.2 Grammar penyusun gambar rumah yang merupakan penggabungan dari *Primitif* penyusunnya.

Computational intelligent approach merupakan solusi untuk memecahkan permasalahan yang menggunakan proses pengembangan secara iteratif berdasarkan pada data tertentu dan proses yang mengandalkan algoritma *heuristik* (pencarian yang tak menentu).

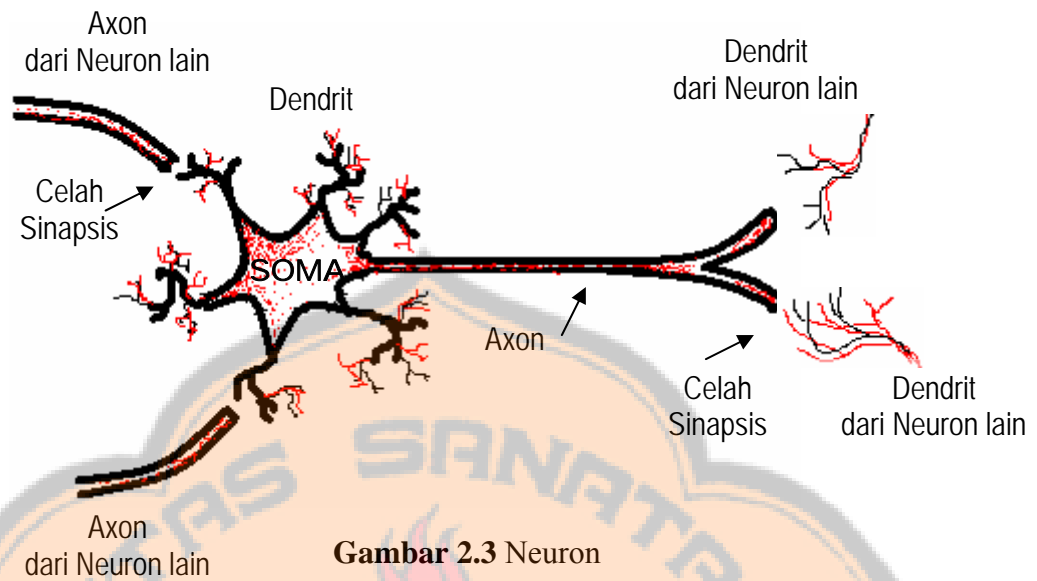
2.2. Jaringan Syaraf Biologi

Otak manusia terdiri dari neuron-neuron dan penghubung yang disebut *Sinapsis*². Neuron bekerja berdasarkan *Impuls* / sinyal yang diberikan pada neuron. Neuron meneruskannya pada neuron lain. Diperkirakan manusia memiliki 10^{12} neuron dan $6 \cdot 10^{18}$ *Sinapsis*. Dengan jumlah yang sebanyak itu otak mampu untuk mengenali pola, melakukan perhitungan, dan mengontrol organ-organ tubuh dengan kecepatan melebihi kecepatan komputer digital.

Otak memiliki struktur yang menakjubkan karena kemampuannya membentuk sendiri aturan-aturan / pola berdasarkan pengalaman yang diterima. Jumlah dan kemampuan neuron berkembang seiring dengan pertumbuhan fisik manusia.

Neuron memiliki 3 komponen penting yaitu *Dendrite*, *Soma*, dan *Axon*. *Dendrit* menerima sinyal dari neuron lain. Sinyal tersebut berupa *Impuls* elektrik yang dikirim melalui celah *Sinaptik* melalui proses kimiawi. Sinyal tersebut dimodifikasi (diperkuat / diperlemah) di celah *Sinaptik*. Berikutnya, *Soma* menjumlahkan semua sinyal yang masuk. Kalau jumlahan itu kuat dan melebihi batas ambang (*Threshold*), maka sinyal tersebut akan diteruskan ke sel lain melalui *Axon*. Frekuensi penerimaan sinyal berbeda-beda antara satu sel dengan sel lainnya.

² Siang, JJ.2005.*Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*.hal 1-2



Gambar 2.3 Neuron

Neuron biologi merupakan sistem yang “*Fault Tolerant*” dalam 2 hal. Pertama, manusia dapat mengenali sinyal input yang agak berbeda dari yang pernah diterima sebelumnya. Sebagai contoh, manusia sering dapat mengenali seseorang yang wajahnya pernah dilihat dari foto, atau dapat mengenali seseorang yang wajahnya agak berbeda karena sudah lama tidak dijumpainya.

Kedua, otak manusia tetap mampu bekerja meskipun beberapa neuronnya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak, neuron lain kadang-kadang dapat dilatih untuk menggantikan fungsi sel yang rusak tersebut.

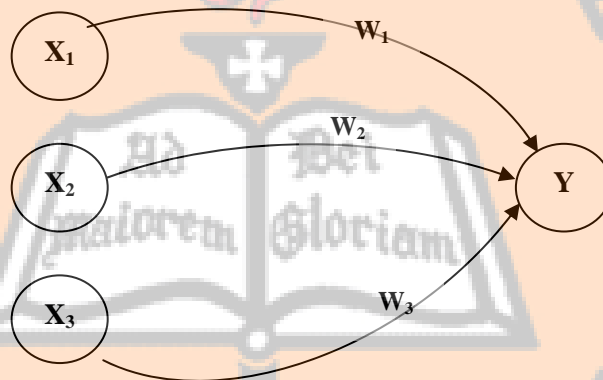
2.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan adalah sebuah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan Jaringan Syaraf Biologi.³ Jaringan Syaraf Tiruan dibentuk sebagai generalisasi Model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

³ *ibid.* hal 2-4

- Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- Sinyal dikirimkan di antara neuron-neuron melalui penghubung-penghubung.
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau melemahkan sinyal.
- Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

Sebagai contoh, perhatikan neuron Y pada gambar di bawah ini.



Gambar 2.4 Jaringan Syaraf Tiruan terdiri dari input (x_1 , x_2 , dan x_3), bobot masing-masing input (w_1 , w_2 , dan w_3), dan keluaran (y)

Neuron Y menerima input dari neuron x_1 , x_2 , dan x_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 , dan w_3 . Ketiga *Impuls* neuron yang ada dijumlahkan

$$net = x_1w_1 + x_2w_2 + x_3w_3$$

Besarnya impuls yang diterima oleh neuron y mengikuti fungsi aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan.

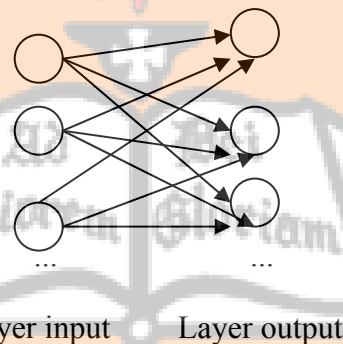
Jaringan Syaraf Tiruan ditentukan oleh 3 hal yaitu Arsitektur Jaringan, Metode Pelatihan, dan Fungsi Aktivasi.

2.3.1. Arsitektur Jaringan

Arsitektur jaringan di dalam Jaringan Syaraf Tiruan adalah pola hubungan antar neuron.⁴

a. Jaringan Layer Tunggal (*Single Layer Network*)

Pada jaringan ini sekumpulan input neuron dihubungkan langsung dengan sekumpulan outputnya. Unit-unit di dalam satu layer tidak terhubung satu sama lain.

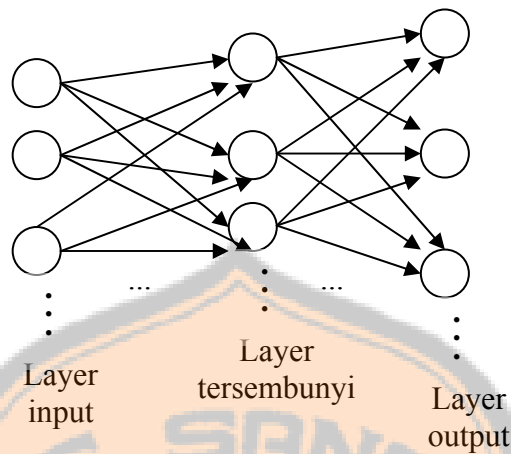


Gambar 2.5 Jaringan Layer Tunggal

b. Jaringan Layer Jamak (*Multi Layer Network*)

Jaringan layer jamak terdiri dari unit input, unit output, dan unit – unit lain diantara input dan output yang sering disebut sebagai layer tersembunyi. Jumlah layer tersembunyi tersebut bisa lebih dari satu layer. Masing-masing layer terdiri dari unit-unit neuron yang saling tidak berhubungan.

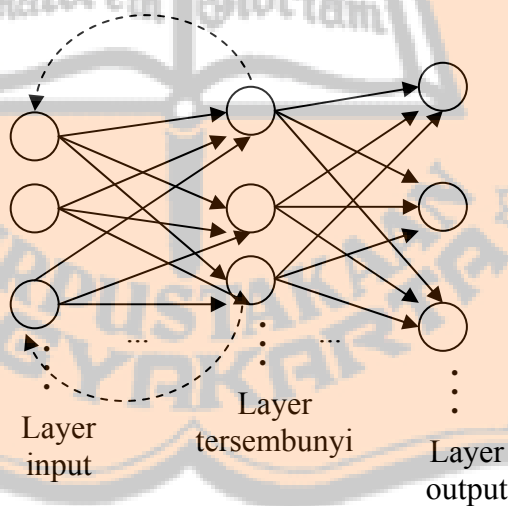
⁴ *ibid.* hal 24-25



Gambar 2.6 Jaringan Layer Jamak

c. Jaringan Recurrent

Model jaringan *Recurrent* mirip dengan Model jaringan layer tunggal maupun ganda hanya saja ada neuron output yang memberikan sinyal pada unit input. Hal itu disebut dengan *Feedback Loop*.



Gambar 2.7. Jaringan Recurrent Sederhana⁵

⁵ Fausett, Laurene.1994.*Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Hal 373

2.3.2. Metode Pelatihan

Pada proses pelatihan akan selalu dilakukan modifikasi bobot sampai ditemukan bobot yang paling optimal untuk mengenali pola.⁶ Metode pelatihan pada Jaringan Syaraf Tiruan terbagi menjadi dua, yaitu pelatihan *Supervised* dan pelatihan *Unsupervised*.

Pada pelatihan *Supervised* terdapat sejumlah pasangan data yaitu masukan dan target keluaran, yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan. Pasangan data tersebut berfungsi sebagai “pemandu” untuk melatih jaringan hingga diperoleh bentuk terbaik.

Sebaliknya, dalam pelatihan *Unsupervised* tidak ada “pemandu” yang akan mengarahkan proses pelatihan. Perubahan bobot di dalam proses pelatihannya dilakukan berdasarkan parameter tertentu dan jaringan dimodifikasi menurut ukuran parameter tersebut.

2.3.3. Fungsi aktivasi

Fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron.⁷ Argumentasi fungsi aktivasi adalah nilai *net* masukan. Jika *net* adalah $\sum x_i w_i$, maka fungsi aktifasinya adalah $f(\text{net}) = f(\sum x_i w_i)$.

Beberapa contoh fungsi aktivasi yang dipakai adalah Fungsi *Threshold* (batas ambang), Fungsi *Sigmoid*, dan Fungsi *Identitas*.

a. Fungsi *Threshold* (batas ambang)

$$f(x) = \begin{cases} 1 & \text{jika } x \geq a \\ 0 & \text{jika } x < a \end{cases} \quad a = \text{nilai Threshold}$$

⁶ Siang, JJ.2005.*Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*.hal 28-29

⁷ *ibid*.hal 26

Fungsi *Threshold* di atas adalah fungsi *Threshold* biner karena hasilnya terdiri dari nilai 1 atau 0. Fungsi *Threshold* yang lain adalah fungsi *Threshold* bipolar yang hasilnya nilai 1 atau -1.

b. Fungsi Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

Nilai yang dihasilkan fungsi tersebut terletak antara 0 dan 1.

c. Fungsi Identitas

$$f(x) = x$$

Fungsi identitas menghasilkan nilai output sama dengan nilai masukannya.

2.4. Pelatihan *Supervised* dan *Unsupervised*

Pada setiap pelatihan *Supervised*, suatu input diberikan ke jaringan. Jaringan akan memproses dan mengeluarkan keluaran. Selisih antara keluaran jaringan dengan target merupakan kesalahan yang terjadi. Jaringan akan memodifikasi bobot sesuai dengan kesalahan tersebut. Model Jaringan Syaraf Tiruan yang menggunakan pelatihan ini diantaranya adalah Jaringan Perceptron, Hebbian, ADALINE, dan Backpropagation.⁸

Pada pelatihan *Unsupervised* perubahan bobot di dalam proses pelatihannya dilakukan berdasarkan parameter tertentu dan jaringan dimodifikasi menurut ukuran parameter tersebut. Contoh Model yang masuk dalam kategori ini adalah Model *Kompetitif*, *Kohonen*, *LVQ (Learning Vector Quantization)*, dan *Neocognitron*.

⁸ *ibid.* hal 28-29

Sebagai contoh, dalam Model Jaringan *Kompetitif* yang terdiri dari 2 layer, yaitu layer input dan layer kompetisi. Layer input menerima data eksternal. Layer kompetisi berisi neuron-neuron yang saling berkompetisi agar memperoleh kesempatan untuk merespon sifat-sifat yang ada dalam data masukan. Neuron yang memenangkan kompetisi akan memperoleh sinyal yang berikutnya ia teruskan. Bobot neuron pemenang akan dimodifikasi sehingga lebih menyerupai data masukan.

2.5. Jaringan Kohonen

Jaringan Kohonen adalah Model jaringan yang menggunakan pelatihan *Unsupervised*.⁹ Pengenalan pola pada metode ini dilakukan dengan mengelompokkan pola yang mirip ke dalam satu kelompok. Pengelompokkan tersebut dilakukan dengan memanfaatkan suatu Matriks Bobot dengan jumlah baris sebanyak kelompok pola yang akan dibentuk dan jumlah kolom sebanyak jumlah kolom vektor input. Matriks Bobot tersebut akan selalu mengalami modifikasi pada setiap *Epoch*-nya. Satu *Epoch* berarti satu kali perhitungan semua vektor input.

2.5.1. Arsitektur Jaringan

Apabila masukan berupa vektor yang terdiri dari n komponen (*Tuple*) yang akan dikelompokkan ke dalam maksimum m buah kelompok (disebut Vektor Contoh). Keluaran jaringan adalah kelompok yang paling dekat / mirip dengan masukan yang diberikan. Ukuran yang dipakai dalam Model Jaringan Kohonen

⁹ *ibid*.hal 141-143

adalah jarak yang paling minimum. Rumus jarak yang dipakai pada percobaan ini

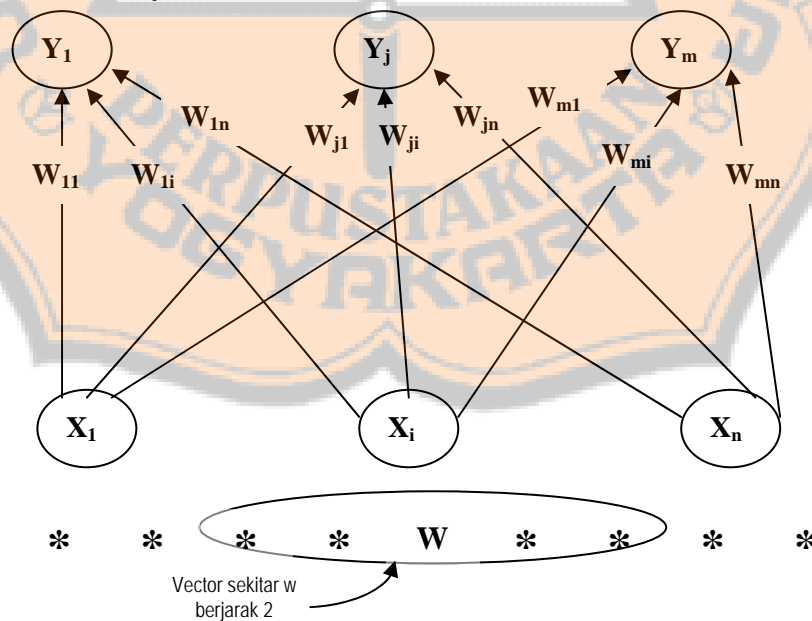
adalah :

$$D(\mathbf{j}) = \sum_i^n (w_{ji} - x_i)^2$$

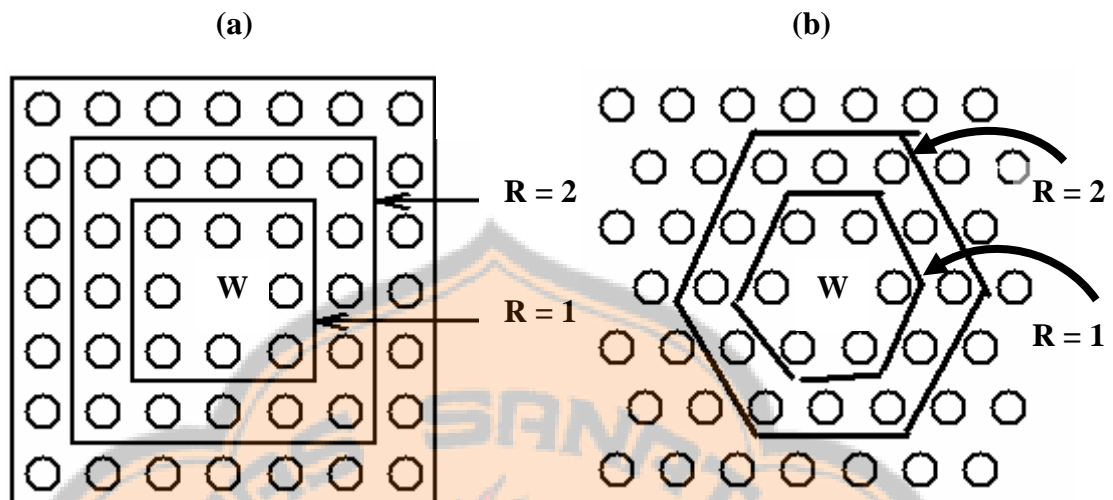
Bobot Vektor Contoh berfungsi sebagai penentu kedekatan Vektor Contoh tersebut dengan masukan yang diberikan. Selama proses pengaturan, Vektor Contoh yang pada saat itu paling dekat dengan masukan akan muncul sebagai pemenang. Vektor Pemenang (dan vektor-vektor di sekitarnya) akan dimodifikasi bobotnya.

Arsitektur Jaringan Kohonen mirip dengan arsitektur model jaringan yang lain, hanya saja Jaringan Kohonen tidak menggunakan perhitungan net maupun fungsi aktivasi.

Misalkan pada suatu iterasi tertentu, Vektor Contoh w menjadi pemenang. Pada iterasi berikutnya, vektor w dan vektor-vektor di sekitarnya akan dimodifikasi bobotnya.



Gambar 2.8 Representasi satu dimensi (linear) Model Jaringan Kohonen
 Gambar di atas menunjukkan kasus untuk vector w berupa vector satu dimensi (dengan jarak $R=2$).



Gambar 2.9 Representasi 2 dimensi Model Jaringan Kohonen dengan $R = 1$ dan $R = 2$

Gambar di atas menunjukkan vector sekitar w jika w direpresentasikan dalam 2 dimensi dengan $R = 1$ dan $R=2$. Jika menggunakan bentuk bujur sangkar dengan jarak $R = 1$, ada 8 vektor di sekitar w , tetapi jika menggunakan bentuk heksagonal hanya 6 vektor di sekitar vector w .

2.5.2. Algoritma Jaringan Kohonen

Algoritma pengelompokan pola jaringan Kohonen adalah sebagai berikut :

0. Inisialisasi

- Bobot w_{ij} (acak)
- Laju pemahaman awal dan faktor penurunannya
- Bentuk dan jari-jari (topologi) sekitarnya

1. Selama kondisi penghentian salah, lakukan langkah 2-7

2. Untuk setiap vector masukan x , lakukan 3-5

3. Hitung $D(j) = \sum_i (w_{ji} - x_i)^2$ untuk semua j

4. Tentukan indeks j sedemikian hingga $D(j)$ minimum
5. Untuk setiap unit j dan unit di sekitarnya dilakukan modifikasi bobot sebagai berikut :

$$w_{ji}^{baru} = w_{ji}^{lama} + \alpha(x_i - w_{ji}^{lama})$$

6. Modifikasi laju pemahaman

Laju pemahaman dimodifikasi dengan mengalikan nilai laju pemahaman dengan koefisien penurunannya.

7. Uji kondisi penghentian

Kondisi penghentian iterasi adalah selisih antara w_{ji} saat itu dengan w_{ji} pada iterasi sebelumnya. Apabila semua w_{ji} hanya berubah sedikit saja, berarti iterasi sudah mencapai konvergensi sehingga dapat dihentikan.

Pengelompokan vektor input dilakukan dengan menghitung jarak vektor dengan bobot optimal.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Sistem Pengenalan Pola ini digunakan untuk mengenali pola angka dari 0 sampai 9 yang berupa file dengan format grafik BMP-1 bit (*Biner*). Proses pengenalannya menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen untuk mengelompokkan angka-angka tersebut ke dalam kelas yang sesuai. Representasi bobot pada Model Jaringan Kohonen ini menggunakan Topologi Satu Dimensi (*Linear*), Topologi Bujursangkar, dan Topologi Heksagonal. Jumlah kelas angka yang akan dibentuk adalah 10 kelas karena angka yang akan dikenali adalah angka 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9. Masing-masing kelas tersebut berisi pola yang mirip satu sama lain.

Sistem Pengenalan Pola Angka ini terdiri dari 4 pemrosesan utama yaitu *Preprocessing*, proses Pelatihan, *Clustering*, dan proses Pengujian. *Preprocessing* adalah pemrosesan pola masukan menjadi pola yang siap untuk diproses pada pemrosesan selanjutnya. Pola yang siap yang dimaksud adalah pola dengan format grafik BMP-1 bit (*Biner*) dan dengan ukuran tertentu. Pada sistem ini disepakati bahwa ukuran pola tersebut adalah 30 x 30 pixel. Pemrosesan yang dilakukan meliputi mengubah ukuran pola, pemotongan dan mengubah matriks pola tersebut menjadi berbentuk vektor.

Proses pelatihan yang dilakukan adalah melatih matriks bobot pada Jaringan Kohonen yang dibentuk dari matriks random dengan jumlah baris sebanyak

jumlah kelas dan jumlah kolom sebanyak jumlah elemen vektor input. Masukan dari proses ini adalah vektor hasil keluaran dari proses *Preprocessing*, Jari-jari Vektor Pemenang, Laju Pemahaman (Alpha), Koefisien Penurunan Alpha, Topologi Jaringan, dan Maksimum Epoch. Masukan vektor pada proses ini terdiri dari vektor hasil *Preprocessing* pola angka 0, 1, 2 ...sampai 9, dengan jumlah masing-masing angka 5 buah pola. Vektor masukan ini disebut Vektor Contoh (*Template*) yang berfungsi untuk melatih Jaringan Kohonen. Pada proses ini matriks bobot dimodifikasi pada setiap perhitungan vektor masukan, yaitu pada baris vektor pemenangnya dan vektor tetangga sesuai dengan jari-jari masukan dan topologi yang dipilih. Proses pelatihan ini akan selalu diulang-ulang dengan memodifikasi Laju Pemahaman dengan menggunakan Koefisien Penurunannya pada setiap iterasi (*Epoch*) sampai diperoleh bobot optimal. Satu kali iterasi (*Epoch*) adalah satu kali pemrosesan semua Vektor Contoh. Bobot optimal adalah bobot yang memiliki selisih nilai tertentu dengan bobot sebelumnya, pada penelitian ini digunakan nilai 0.

Proses yang dilakukan setelah diperoleh bobot optimal adalah *Clustering*. *Clustering* ini adalah proses pengelompokkan Pola *Template* ke dalam semua kelas kemudian menamainya sesuai dengan angka-angka yang terkumpul di dalam masing-masing kelas. Kelas yang mempunyai identitas⁵ adalah kelas yang hanya menampung lima buah pola yang terdiri satu jenis angka saja. Sedangkan untuk kelas yang menampung lebih dari lima buah pola, jenis angka lebih dari satu, atau tidak memiliki anggota sama sekali tidak akan memiliki identitas. Pola

⁵ nama kelas sesuai angka yang terkumpul di dalamnya

uji yang masuk ke dalam kelas tanpa identitas akan menjadi pola yang tidak dikenali.

Proses pengenalan adalah pengelompokan pola uji yang dilakukan dengan menghitung jarak vektor pola uji dengan bobot optimal. Baris bobot yang menghasilkan perhitungan jarak terkecil merupakan kelas dari vektor masukan tersebut.

3.2. Perancangan

3.2.1. Perancangan Proses Pengenalan Pola

Proses yang ada pada sistem pengenalan pola ini meliputi Proses Memasukkan Data Pola Angka untuk pelatihan dan pengujian, menampilkan Data Pola Angka yang meliputi Pola Template, Pola Uji, hasil *Preprocessing* Pola Template dan hasil *Preprocessing* Pola Uji, menghapus data Pola Uji atau Pola Template, *Preprocessing*, Pelatihan, *Clustering*, dan Proses Pengenalan Pola Angka.

a. Proses Memasukkan Data Pola Angka

Pola angka yang dimasukkan adalah pola template dan pola uji. Pada proses ini pola angka 0, 1, 2 ...sampai 9 ke dalam suatu variabel. Variabel tersebut digunakan pada proses-proses selanjutnya dalam sistem pengenalan pola angka 0 sampai 9 menggunakan Metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

b. Proses Menampilkan Data Pola Angka

Proses ini berfungsi untuk menampilkan data Pola *Template*, Pola Uji, hasil *Preprocessing* Pola *Template*, dan hasil *Preprocessing* Pola Uji.

c. Proses Menghapus Data Pola Angka

Pada proses ini data Pola Uji atau Pola *Template* yang tersimpan di dalam variabel penyimpanan dapat dihapus. Proses ini disediakan pada Sistem Pengenalan Pola Angka ini sebagai fasilitas dalam penggantian Pola Uji dan Pola *Template*.

d. *Preprocessing*

Pada proses ini Pola Input yang berupa file gambar dengan format grafik BMP-1 bit (*biner*) menjadi gambar berukuran 30 x 30 pixel. Sebelum dilakukan perubahan ukuran, obyek angka pada gambar disesuaikan posisinya agar obyek memenuhi gambar. Penyesuaian tersebut dilakukan dengan memotong daerah yang kosong. Asumsi untuk daerah Pola Angka yang digunakan untuk masukan pada Jaringan Kohonen pada penelitian ini adalah 0.

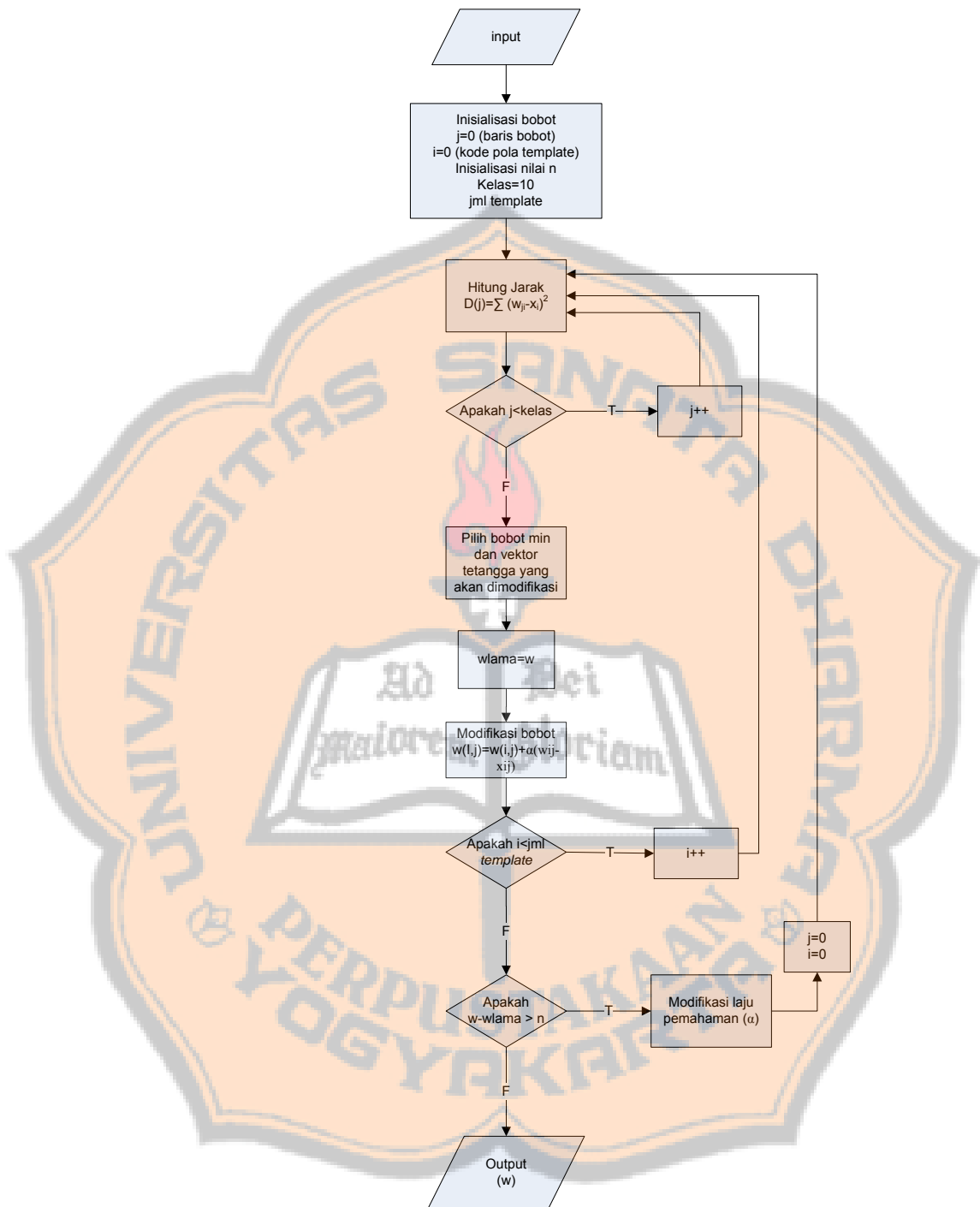
Gambar yang telah disesuaikan ukuran dan nilai pixelnya tersebut yang berupa matriks diubah menjadi bentuk vektor. Vektor tersebut memiliki anggota sebanyak 30 x 30 pixel yaitu 900 pixel, dengan penyusunan baris pertama matriks dilanjutkan baris ke-dua, ke-tiga, ke-empat, sampai baris ke-tiga puluh matriks.

e. Pelatihan

Proses Pelatihan bertujuan untuk melatih Jaringan Kohonen yaitu pada Matriks Bobotnya yang pada awalnya diinisialisasi sebagai matriks *Random* dengan jumlah baris sebanyak jumlah kelas dan jumlah kolom sebanyak jumlah elemen vektor input sampai menghasilkan Matriks Bobot yang paling optimal. Matriks Bobot optimal inilah yang digunakan untuk Klasifikasi Pola.

Diagram proses pelatihan Jaringan Kohonen ini dapat dilihat pada *Flowchart* pada halaman berikut.





Gambar 3.1 Flowchart proses pelatihan

Masukan (*Input*) dari proses Pelatihan ini adalah vektor hasil *Preprocessing*, Laju Pemahaman (α), Koefisien Penurunan

Alpha, Topologi Jaringan, Jari-jari Vektor Pemenang, dan Jumlah Maksimum Epoch. Jumlah vektor yang dimasukkan sebanyak 50 vektor yang terdiri dari Vektor Pola Angka 0 sampai 9 dengan masing-masing angka terdiri dari 5 buah pola.

Pada tahap inisialisasi dibentuk Matriks Bobot dengan jumlah baris merepresentasikan jumlah kelas yang akan dibentuk, jumlah kolomnya sama dengan jumlah komponen di dalam vektor masukan, dan memiliki komponen bilangan random. Selain Matriks Bobot, beberapa variabel yang diinisialisasi pada proses ini adalah variabel j yang digunakan untuk merekam kode kelas, variabel i yang digunakan untuk merekam kode Pola *Template*, nilai variabel n yang digunakan sebagai parameter pengujian keoptimalan Matriks Bobot, jumlah kelas ditetapkan sebanyak sepuluh kelas karena pola yang dikenali adalah pola 0 sampai 9, dan inisialisasi jumlah *Template* sesuai dengan masukan. Nilai variabel n yang digunakan dalam penelitian ini adalah 0.

Proses penghitungan jarak dilakukan untuk semua Pola *Template*, dengan rumus :

$$D(j) = \sum_i^{900} (w_{ji} - x_i)^2$$

Pada setiap penghitungan jarak sebuah Vektor Input dengan Matriks Bobot, dicari baris bobot yang memiliki jarak terpendek

dengan vektor input dan baris bobot tetangga sesuai dengan jari-jari dan topologi yang dimasukkan untuk dimodifikasi⁶, dengan rumus :

$$w_{ji}^{baru} = w_{ji}^{lama} + \alpha(x_i - w_{ji}^{lama})$$

Setelah semua proses di atas dilakukan pada semua Pola *Template* maka akan dilakukan pengujian apakah selisih bobot baru dengan bobot lama lebih dari nilai variable n . Jika syarat tersebut memenuhi maka semua proses Pelatihan dilakukan kembali dengan memproses lagi semua Pola *Template* dengan bobot baru hasil dari proses sebelumnya dan menggunakan Laju Pemahaman yang sudah dimodifikasi. Satu iterasi pemrosesan semua pola template disebut satu *Epoch*.

Jika selisih bobot terbaru dengan bobot dari *Epoch* sebelumnya tidak lebih besar dari nilai variabel n maka iterasi dihentikan dan bobot yang dihasilkan adalah bobot yang paling optimal pada pelatihan tersebut.

Proses Penentuan Vektor yang dimodifikasi

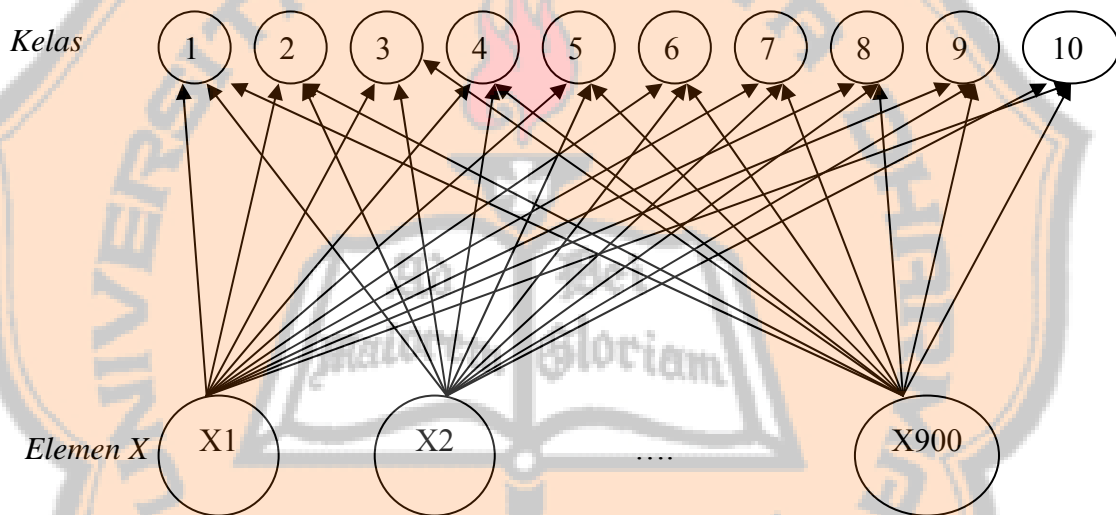
Pada setiap penghitungan jarak sebuah Vektor Input dengan Matriks Bobot, dicari baris bobot yang memiliki jarak terpendek dengan vektor input dan baris bobot tetangga sesuai dengan jari-jari dan topologi yang dimasukkan untuk dimodifikasi. Terdapat tiga topologi yang digunakan untuk menentukan vektor tetangga yang

⁶ Penentuan vektor bobot yang dimodifikasi akan dibahas lebih lanjut di Proses penentuan vektor tetangga yang dimodifikasi.

akan dimodifikasi, yaitu Topologi Linear, Bujursangkar, dan Heksagonal.

i. Topologi Linear

Sepuluh kelas yang digunakan untuk pengelompokan angka 0 sampai 10 ini direpresentasikan dalam bentuk satu dimensi atau vektor. Arsitektur jaringannya adalah sebagai berikut :



Gambar 3.2 Arsitektur Jaringan Kohonen Topologi Linear

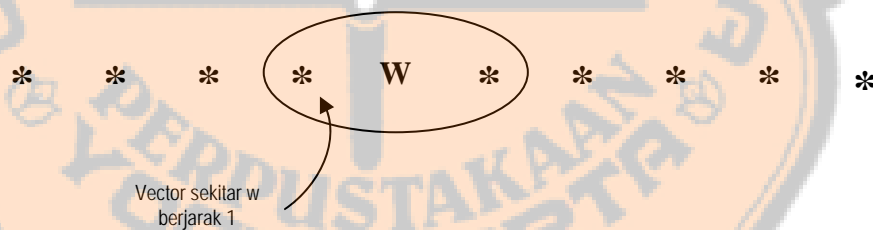
Masing-masing garis yang menghubungkan elemen X dengan kelas mempunyai nilai yang disimpan dalam sebuah matriks bobot dengan letak baris sesuai dengan kelas yang dituju dan pada kolom ke-n sesuai dengan elemen X_n . Jadi, Matriks Bobot tersebut adalah matriks 10×900 (10 baris dan 900 kolom). Setiap baris vektor dalam Matriks Bobot tersebut

mewakili satu kelas. Vektor bobot masing-masing kelas tersebut yang akan dimodifikasi pada setiap iterasi pada proses pelatihan.


$$\begin{pmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1900} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2900} \\ \dots & \dots & \dots & \dots & \dots \\ W_{101} & W_{102} & W_{103} & \dots & W_{10900} \end{pmatrix}$$

Gambar 3.3 Matriks Bobot

Apabila diketahui jari-jari yang dimasukkan pada Topologi Linear ini adalah 1. Vektor pemenang adalah vektor baris ke-5, maka vektor yang akan dimodifikasi bobotnya adalah vektor pemenang tersebut dan vektor tetangganya. Vektor tetangganya adalah 1 buah vektor sebelum vektor pemenang dan 1 buah vektor setelah vektor pemenang, yaitu vektor baris ke-4, dan vektor baris ke-6.



Gambar 3.4 Kelas yang dimodifikasi pada Topologi Linear dengan jari-jari 1

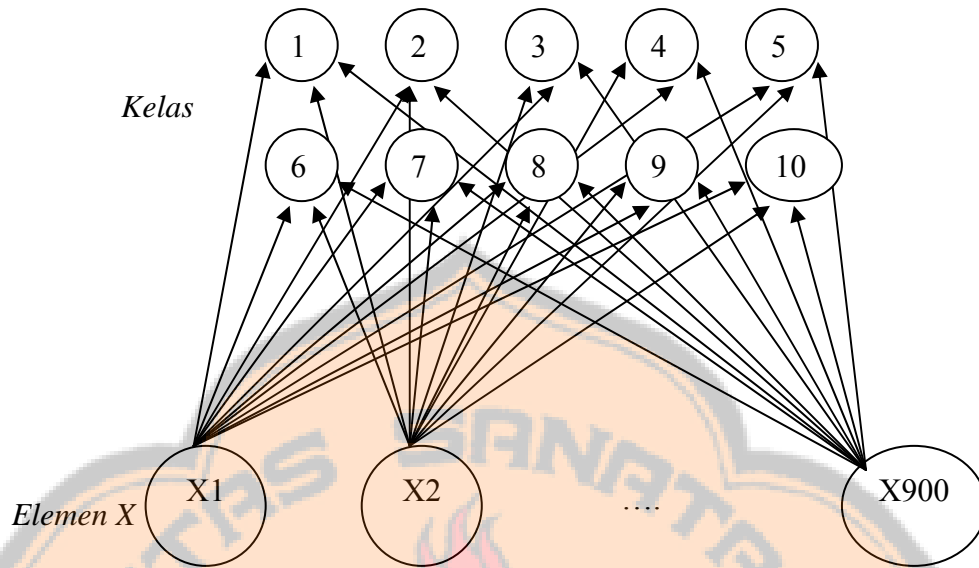


$W_{1\ 1}$	$W_{1\ 2}$	$W_{1\ 3}$...	$W_{1\ 900}$
$W_{2\ 1}$	$W_{2\ 2}$	$W_{2\ 3}$...	$W_{2\ 900}$
$W_{3\ 1}$	$W_{3\ 2}$	$W_{3\ 3}$...	$W_{3\ 900}$
$W_{4\ 1}$	$W_{4\ 2}$	$W_{4\ 3}$...	$W_{4\ 900}$
$W_{5\ 1}$	$W_{5\ 2}$	$W_{5\ 3}$...	$W_{5\ 900}$
$W_{6\ 1}$	$W_{6\ 2}$	$W_{6\ 3}$...	$W_{6\ 900}$
$W_{7\ 1}$	$W_{7\ 2}$	$W_{7\ 3}$...	$W_{7\ 900}$
$W_{8\ 1}$	$W_{8\ 2}$	$W_{8\ 3}$...	$W_{8\ 900}$
$W_{9\ 1}$	$W_{9\ 2}$	$W_{9\ 3}$...	$W_{9\ 900}$
$W_{10\ 1}$	$W_{10\ 2}$	$W_{10\ 3}$...	$W_{10\ 900}$

Gambar 3.5 Baris Vektor Bobot yang dimodifikasi

ii. Topologi Bujursangkar

Pada topologi ini kesepuluh kelas disusun dalam bentuk 2 dimensi atau matriks. Pada penelitian ini matriks kelas tersebut disusun menjadi matriks 2 baris dan 5 kolom. Setiap kelas terhubung dengan elemen vektor dari vektor input yang diberi nama X. Setiap keterhubungan tersebut memiliki nilai yang disimpan dalam Matrik Bobot dengan letak baris sesuai dengan kelas yang dituju dan pada kolom sesuai dengan indek yang dimiliki X.



Gambar 3.6 Arsitektur Jaringan Kohonen Topologi Bujursangkar

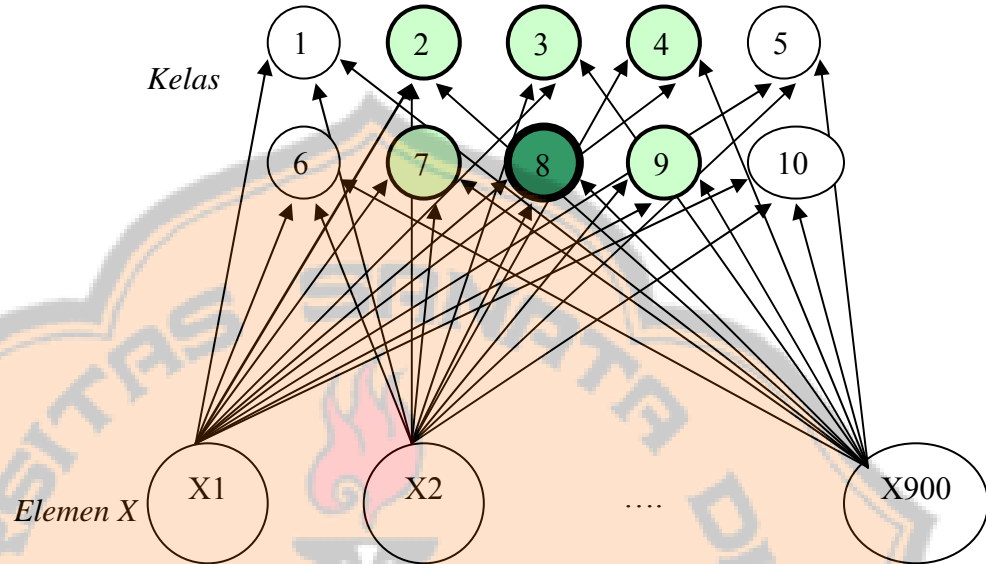
Matrik Bobot ini terdiri dari 10 baris dan 900 kolom dimana setiap barisnya mewakili satu kelas.

$$\begin{pmatrix}
 W_{11} & W_{12} & W_{13} & \dots & W_{1900} \\
 W_{21} & W_{22} & W_{23} & \dots & W_{2900} \\
 \dots & \dots & \dots & \dots & \dots \\
 W_{101} & W_{102} & W_{103} & \dots & W_{10900}
 \end{pmatrix}$$

Gambar 3.7 Matriks Bobot

Apabila diketahui jari-jari yang dimasukkan pada Topologi Bujursangkar ini adalah 1. Vektor Pemenang adalah Vektor baris ke-8, maka vektor yang akan dimodifikasi bobotnya adalah vektor pemenang tersebut dan vektor tetangganya. Vektor tetangga pada keadaan ini adalah vektor yang dimiliki oleh kelas yang merupakan kelas tetangga dengan kelas vektor pemenang dengan aturan 8-tetangga pada matriks kelas. Sehingga Vektor Tetangga

yang dimodifikasi adalah Vektor baris ke-7, Vektor baris ke-2, Vektor baris ke-3, Vektor baris ke-4, dan Vektor baris ke-9.



Gambar 3.8 Kelas yang dimodifikasi pada Topologi Bujursangkar dengan jari-jari 1

W_{11}	W_{12}	W_{13}	...	W_{1900}
W_{21}	W_{22}	W_{23}	...	W_{2900}
W_{31}	W_{32}	W_{33}	...	W_{3900}
W_{41}	W_{42}	W_{43}	...	W_{4900}
W_{51}	W_{52}	W_{53}	...	W_{5900}
W_{61}	W_{62}	W_{63}	...	W_{6900}
W_{71}	W_{72}	W_{73}	...	W_{7900}
W_{81}	W_{82}	W_{83}	...	W_{8900}
W_{91}	W_{92}	W_{93}	...	W_{9900}
W_{101}	W_{102}	W_{103}	...	W_{10900}

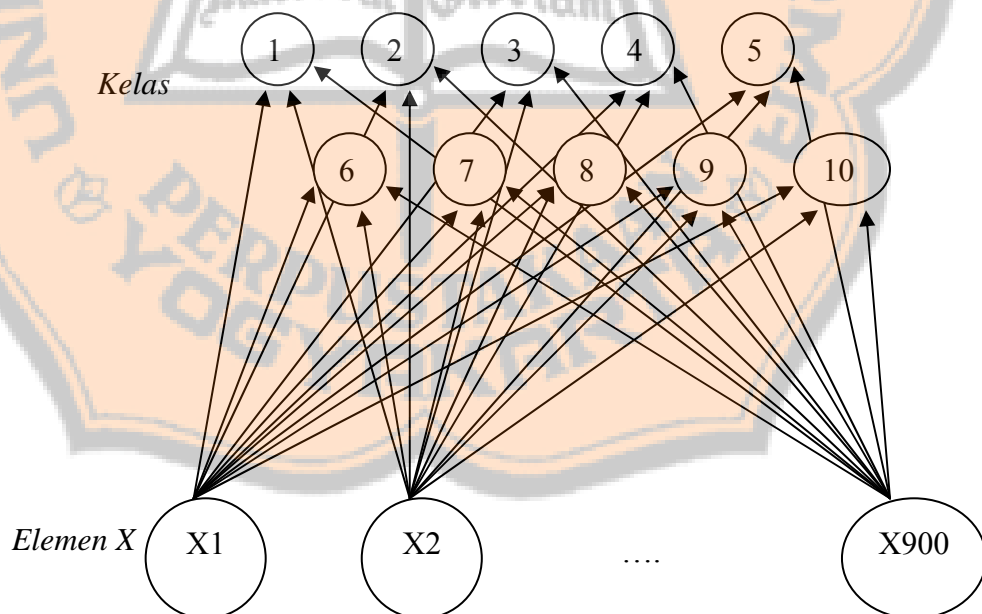
Gambar 3.9 Baris vektor bobot yang dimodifikasi

iii. Topologi Heksagonal

Penyusunan kelas pada Topologi Heksagonal ini juga dalam bentuk 2 dimensi atau matriks, yaitu matriks 2 baris dan 5 kolom. Hanya saja dalam penyusunannya kita asumsikan sebagai berikut :

1	2	3	4	5
6	7	8	9	10

Seperti topologi yang lain, setiap kelas pada matriks kelas terhubung dengan vektor input (elemen X). Setiap keterhubungan tersebut memiliki nilai yang disimpan dalam matriks bobot dengan letak baris sesuai dengan kelas yang dituju dan pada kolom sesuai dengan indek yang dimiliki X.



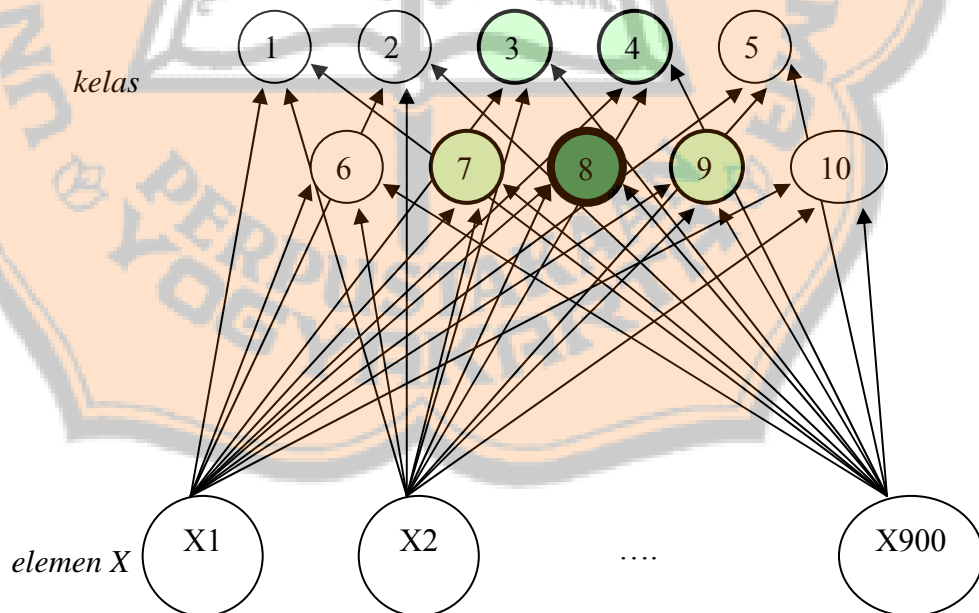
Gambar 3.10 Arsitektur Jaringan Kohonen Topologi Heksagonal

Matriks Bobot tersebut terdiri dari 10 baris dan 900 kolom dimana setiap barisnya mewakili satu kelas.

$$\begin{pmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1900} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2900} \\ \dots & \dots & \dots & \dots & \dots \\ W_{101} & W_{102} & W_{103} & \dots & W_{10900} \end{pmatrix}$$

Gambar 3.11 Matriks Bobot

Apabila diketahui jari-jari yang dimasukkan pada topologi bujur sangkar ini adalah 1. Vektor pemenang adalah vektor baris ke-8, maka vektor yang akan dimodifikasi bobotnya adalah vektor pemenang tersebut dan vektor tetangganya. Vektor tetangga dari kelas 8 tersebut adalah vektor di sekitar kelas 8, seperti terlihat pada gambar dibawah ini.



Gambar 3.12 Kelas yang dimodifikasi pada Topologi Heksagonal dengan jari-jari 1

Jadi, vektor tetangga yang dimodifikasi dengan jari-jari 1 adalah vektor baris ke-7, vektor baris ke-3, vektor baris ke-4, dan vektor baris ke-9.

$W_{1 1}$	$W_{1 2}$	$W_{1 3}$...	$W_{1 900}$
$W_{2 1}$	$W_{2 2}$	$W_{2 3}$...	$W_{2 900}$
$W_{3 1}$	$W_{3 2}$	$W_{3 3}$...	$W_{3 900}$
$W_{4 1}$	$W_{4 2}$	$W_{4 3}$...	$W_{4 900}$
$W_{5 1}$	$W_{5 2}$	$W_{5 3}$...	$W_{5 900}$
$W_{6 1}$	$W_{6 2}$	$W_{6 3}$...	$W_{6 900}$
$W_{7 1}$	$W_{7 2}$	$W_{7 3}$...	$W_{7 900}$
$W_{8 1}$	$W_{8 2}$	$W_{8 3}$...	$W_{8 900}$
$W_{9 1}$	$W_{9 2}$	$W_{9 3}$...	$W_{9 900}$
$W_{10 1}$	$W_{10 2}$	$W_{10 3}$...	$W_{10 900}$

Gambar 3.13 Baris Vektor Bobot yang dimodifikasi

f. *Clustering*

Setelah dihasilkan bobot optimal dari proses pelatihan, maka untuk mengetahui identitas kelas dilakukan proses *Clustering*. Proses *Clustering* terdiri dari 2 tahap, yaitu klasifikasi pola *Template* dan penamaan⁷ kelas. Pengklasifikasian pola *Template* dilakukan dengan menghitung jarak pola *Template* dengan bobot optimal yang sudah dihasilkan. Baris bobot yang menghasilkan nilai minimum adalah baris yang menyatakan kelas dari pola *Template* tersebut. Jika terdapat lebih dari satu nilai minimum maka disepakati bahwa pola

⁷ pemberian identitas kelas

Template tersebut termasuk di dalam kelas dengan indeks yang terkecil dari nilai minimum tersebut.⁸

Pengelompokkan pola *Template* ini digunakan untuk menamai masing-masing kelas yang ditempati oleh pola-pola *Template*. Kelas yang memiliki anggota semua pola angka yang sama (lima pola angka) memiliki identitas angka tersebut. Sehingga pada saat pengujian, apabila ada pola uji yang masuk dalam kelas tersebut maka pola uji tersebut dikenali sebagai angka identitas dari kelas tersebut. Sedangkan untuk kelas yang memiliki lebih dari satu jenis pola *Template* atau sama sekali tidak memiliki anggota, tidak akan memiliki angka identitas. Apabila pada proses pengujian terdapat pola uji yang masuk ke dalam kelas ini maka pola tersebut termasuk pola yang tidak dikenali.

g. Pengujian

Proses pengujian dilakukan dengan menggunakan pola-pola angka selain pola angka yang digunakan untuk proses pelatihan. Pola yang diuji adalah pola angka dari 0 sampai 9 yang berupa file dengan format grafik BMP-1 bit (*biner*) yang telah dikenai *Preprocessing* yang sama dengan pola *Template* sehingga menjadi berbentuk vektor dengan anggota sebanyak 30 x 30 yaitu 900 pixel. Vektor masukan tersebut kemudian dihitung jaraknya dengan bobot optimal yang dihasilkan pada proses pelatihan. Dari proses perhitungan jarak

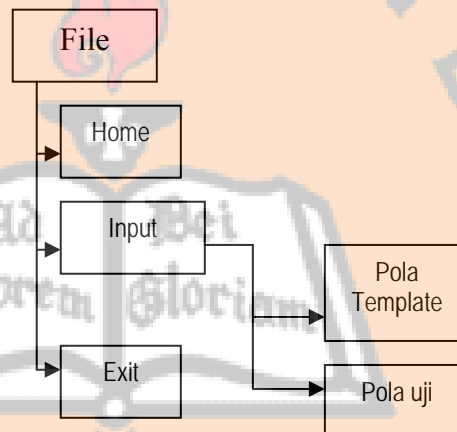
⁸ E.Dayhoff, Judith. (1990). *Neural Network Architectures An Introduction*. Hal 166.

tersebut maka dapat diketahui kelas dari vektor masukan tersebut. Vektor masukan akan dikenali sebagai angka identitas dari kelas tersebut. Angka identitas kelas adalah angka yang diperoleh dari proses pengelompokan pola *Template*.

3.2.2. Perancangan Menu

Program Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen ini memiliki 5 menu utama, yaitu:

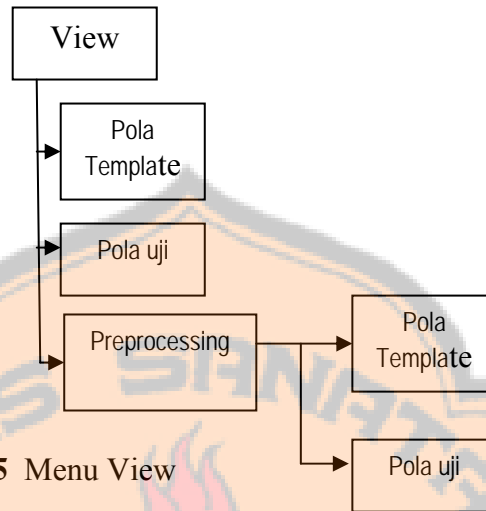
a. Menu File



Gambar 3.14 Menu File beserta submenu di dalamnya

Menu file ini memiliki submenu Home, Input, dan Exit. Home berfungsi memanggil form Home, Input mempunyai submenu Pola Template dan Pola Uji yang berfungsi memanggil form untuk memasukkan pola dan Exit untuk keluar dari program. Ketika menu Exit dipilih maka akan keluar pesan penegasan apakah user benar ingin keluar dari program.

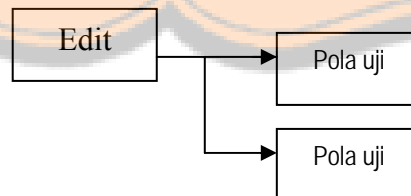
b. Menu View



Gambar 3.15 Menu View

Menu ini memiliki submenu Pola Template, Pola Uji, dan Preprocessing yang memiliki submenu Pola Template dan Pola Uji. Submenu Pola Template berfungsi untuk menampilkan pola *Template*, submenu Pola Uji berfungsi untuk menampilkan pola uji, submenu Pola Template pada submenu Preprocessing berfungsi untuk menampilkan hasil *Preprocessing* pola *Template*, dan submenu Pola Uji pada submenu Preprocessing berfungsi untuk menampilkan hasil *Preprocessing* pola uji.

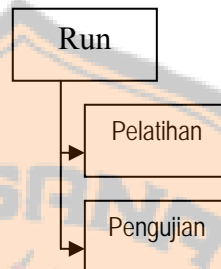
c. Menu Edit



Gambar 3.16 Menu Edit beserta submenu di dalamnya

Menu ini berisi submenu Pola Uji dan Pola *Template*. Fungsi dari submenu adalah sebagai pemanggil form yang digunakan untuk menghapus seluruh pola uji atau pola *Template*.

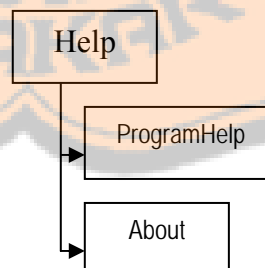
d. Menu Run



Gambar 3.17 Menu Run beserta submenu di dalamnya

Menu ini terdiri dari submenu Pelatihan dan Pengujian. Submenu Pelatihan berfungsi sebagai pemanggil form Pelatihan yang digunakan untuk melakukan pelatihan jaringan. Submenu Pengujian berfungsi untuk memproses pola uji ke jaringan yang telah dilatih pada proses pelatihan dan menampilkan form Hasil Pengujian yang berupa tampilan pola uji serta pernyataan dikenali sebagai angka apa. Jika pola uji belum ada maka ketika memilih submenu Pengujian akan muncul pesan yang menyatakan pola uji belum ada dan perintah untuk mengisinya.

e. Menu Help

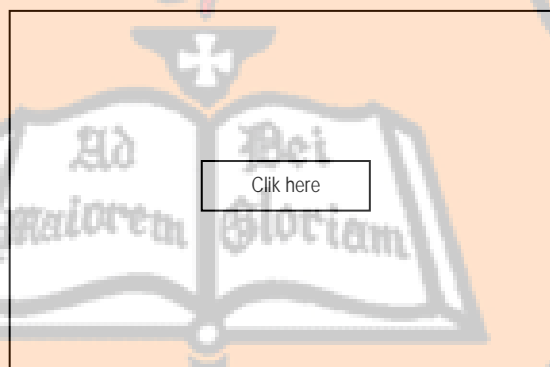


Gambar 3.18 Menu Help beserta submenu di dalamnya

Menu ini terdiri dari submenu ProgramHelp dan About. Submenu ProgramHelp berfungsi memanggil form ProgramHelp yang berisi informasi tentang cara menggunakan program Pengenalan Pola Angka 0 sampai 9 dengan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen ini. Submenu About berfungsi sebagai pemanggil form yang berisi tentang informasi tentang Software meliputi versi software, tahun pembuatan, dan siapa yang membuat.

3.2.3. Perancangan *Interface* Sistem Pengenalan Pola


a. Form *Enter*



Gambar 3.19 Form Enter

Form ini merupakan 'pintu gerbang' dari program pengenalan pola angka ini. Form ini berfungsi untuk mengosongkan semua variabel global yang digunakan pada program. Tombol 'Klik here' akan mengantarkan user ke halaman Home yang merupakan halaman utama program Pengenalan Pola Angka Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

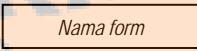
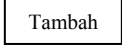
b. Form Home

Menu Utama
Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen <div style="text-align: center;">  </div> Oleh : Clara Wastiunamsih (033124002) Dosen Pembimbing : Y.Joko Nugroho, S.Si. Program Studi Ilmu Komputer Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sanata Dharma Yogyakarta 2007

Gambar 3.20 Form Home

Form ini merupakan halaman utama dari program Pengenalan Pola Angka Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen. Halaman utama ini menampilkan judul program, logo Universitas Sanata Dharma, nama pembuat, nama dosen pembimbing, Program Studi, Jurusan, Fakultas, dan Universitas dari pembuat program. Pada form ini terdapat menu utama dengan rincian seperti penjelasan pada perancangan menu.

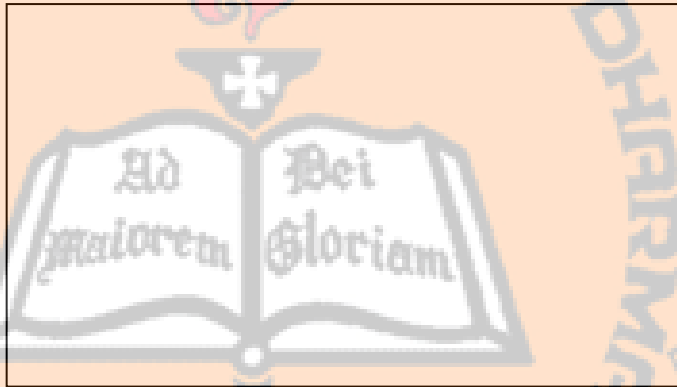
c. Form Input Pola

Menu Utama
Pengenalan Pola Angka 0 sampai 9 <div style="text-align: center;">  </div> <div style="text-align: center; margin-top: 20px;">  </div>

Gambar 3.21 Form Input Pola

Form ini berfungsi untuk menginputkan pola uji. Pola uji diinputkan satu persatu dengan menekan tombol tambah dan akan langsung ditampilkan.

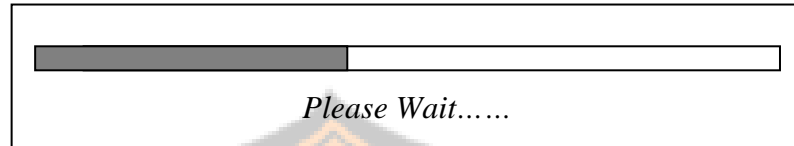
d. Form Tampil Pola

Menu Utama
Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <i>Nama form</i> </div> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 10px auto; text-align: center;">  </div>

Gambar 3.22 Form Tampil Pola

Format form ini digunakan untuk menampilkan pola sesuai dengan menu yang dipilih, yaitu Pola Template, Pola Uji, hasil *Preprocessing* pola *Template*, atau hasil *Preprocessing* pola uji. Pada saat proses menampilkan pola *Template* dan hasil *Preprocessing* pola *Template*, ditampilkan waitbar dengan tulisan "Please Wait" karena proses penampilannya membutuhkan waktu yang agak lama. Hal tersebut

berfungsi untuk memberi informasi kepada user tentang lama proses penampilan sehingga user bersedia menunggu.



Gambar 3.23 Waitbar

e. Form Edit Pola

Menu Utama
Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen
<input type="text" value="Nama form"/>
 Gambar Pola
<input type="button" value="Hapus Semua Pola"/>

Gambar 3.24 Form Edit Pola

Form Edit Pola tersebut akan menampilkan semua Pola Template atau Pola Uji. Pada form ini terdapat tombol Hapus Semua Pola, yang berfungsi untuk menghapus semua Pola. Dengan menekan tombol ini maka semua Pola akan dihapus kemudian program akan menampilkan form Pola yang kosong.

f. Form Pelatihan

Menu Utama					
Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen					
<i>Nama form</i>					
Alpha	<input type="text"/>	<table border="1"> <tr> <td>Proses</td> </tr> <tr> <td><i>Clustering</i></td> </tr> <tr> <td>Epoch</td> </tr> </table>	Proses	<i>Clustering</i>	Epoch
Proses					
<i>Clustering</i>					
Epoch					
Koefisien penurunan	<input type="text"/>				
Jari-jari	<input type="text"/> <input checked="" type="checkbox"/>				
Topologi	<input type="text"/> <input checked="" type="checkbox"/>				
Maximum epoch	<input type="text"/> <input checked="" type="checkbox"/>				
<table border="1"> <tr> <td style="width: 50%; text-align: center;"><i>(gambar bobot sebelum pelatihan)</i></td> <td style="width: 50%; text-align: center;"><i>(gambar bobot setelah pelatihan)</i></td> </tr> </table>		<i>(gambar bobot sebelum pelatihan)</i>	<i>(gambar bobot setelah pelatihan)</i>		
<i>(gambar bobot sebelum pelatihan)</i>	<i>(gambar bobot setelah pelatihan)</i>				

Gambar 3.25 Form Pelatihan

Pada form ini dilakukan pelatihan jaringan dengan memasukkan nilai Alpha, Koefisien Penurunan Alpha, Jari-jari, Topologi, dan Maximum Epoch yang dimasukkan oleh user. Nilai Alpha dan Koefisien penurunannya harus berupa nilai antara 0 dan 1. Jika user memasukkan nilai diluar range tersebut maka akan tampil *Warning* yang menampilkan informasi nilai yang seharusnya dimasukkan. Untuk memasukkan jari-jari tinggal memilih salah satu nilai pada popmenu nilai jari-jari. Jika user tidak memilih popmenu ini maka nilai default jari-jari adalah 0. Begitu pula untuk topologi dan Maximum Epoch, jika user tidak memilih

popmenu tersebut maka defaultnya adalah sesuai dengan yang ditampilkan pada popmenu tersebut.

Input-input tersebut akan digunakan untuk proses pelatihan yaitu dengan menekan tombol proses. Pada saat menekan tombol proses maka dilakukan proses *Preprocessing* pola *Template* yang telah dimasukkan di dalam program, pelatihan dengan hasil *Preprocessing* pola *Template* tersebut, serta masukan yang dimasukkan oleh user melalui form pelatihan tersebut. Pada saat proses berjalan akan ditampilkan waitbar, dan nilai *Epoch* yang sedang ditempuh. Ketika proses pelatihan berakhir, maka akan ditampilkan gambar bobot sebelum pelatihan dan setelah pelatihan. Proses akan berhenti ketika selisih bobot terbaru dengan bobot sebelumnya bernilai 0 atau ketika proses telah mencapai nilai maksimum epoch yang telah dimasukkan sebelumnya.

Setelah proses ini selesai maka user harus menekan tombol *Clustering*, untuk menamai kelas-kelas tersebut berdasarkan bobot optimal yang telah diperoleh dari proses pelatihan yang baru saja dilakukan. Dengan demikian masing-masing kelas yang memenuhi syarat telah memiliki identitas angka.⁹ Pada saat menekan tombol *Clustering* ini maka program akan menampilkan form hasil pelatihan.

⁹ Lihat pada perancangan proses *clustering*.

g. Form Hasil Pelatihan

Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen

Nama form

...

Back

Gambar 3.26 Form Hasil Pelatihan

Form ini menampilkan kelas beserta anggotanya. Kelas-kelas tersebut akan dinamai dengan huruf A, B, C, ...sampai J. Anggotanya berupa angka 0, 1, 2, 3, ..sampai 9 dengan masing-masing angka berjumlah 5 buah. Proses penampilan anggota angka ini dilakukan dengan menampilkan indek urutan dari pola template tersebut. Berdasarkan hasil *Clustering* tersebut user dapat memberi identitas pada setiap kelas dengan popupmenu disamping tampilan hasil *Clustering*.

Tombol *Back* di bagian pojok kanan pada form ini akan membawa user ke form Home. Setelah itu user bisa melakukan proses pengujian dengan menggunakan pola uji.

h. Form Hasil Pengujian

Form ini akan ditampilkan oleh program jika user sudah memasukkan pola uji, melakukan proses pelatihan, dan melakukan proses *Clustering* serta menamai setiap kelasnya. Setelah syarat-syarat tersebut terpenuhi maka pada form ini akan ditampilkan semua pola uji, kemudian di bawahnya akan ditampilkan informasi tentang pola uji tersebut dikenali sebagai angka tertentu atau tidak dikenali. Keterangan pengenalan pola tersebuturut dari kiri ke kanan sebagai pola ke-1, ke-2, ke-3, ...dan seterusnya sesuai dengan jumlah pola uji.

Menu Utama	
Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen	
<input type="text" value="Nama form"/>	
<input type="text"/>	<input type="text"/>
Dikenali sebagai angka....	
Dikenali sebagai angka....	

Gambar 3.27 Form Hasil Pengujian

i. Form ProgramHelp

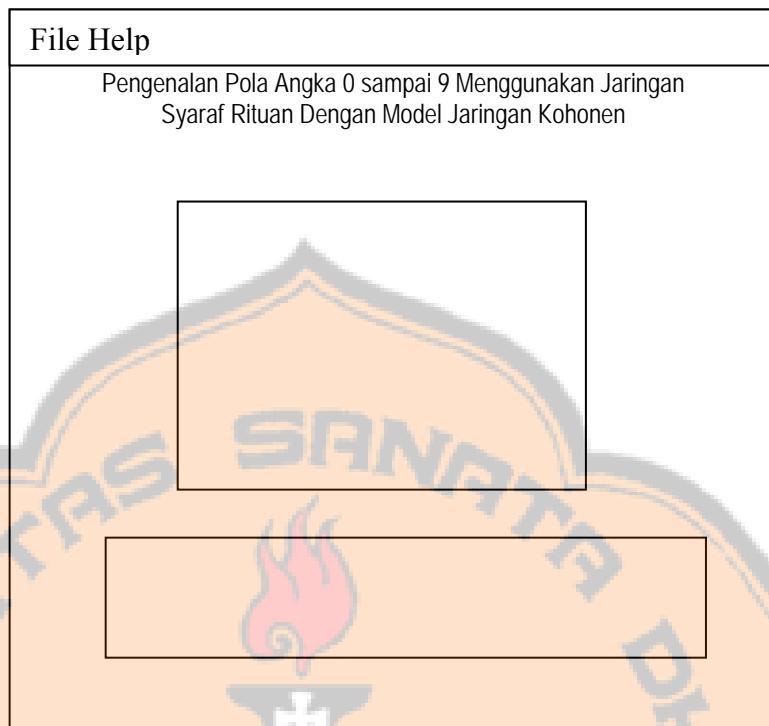
File Help
Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Rituan Dengan Model Jaringan Kohonen
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <i>Nama form</i> </div> <div style="border: 1px solid black; height: 100px; width: 100%; margin-top: 10px;"></div>

Gambar 3.28 Form ProgramHelp

Form ProgramHelp ini berisi tentang informasi cara pemakaian program dan tentang Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen. Menu yang dimiliki halaman ini berbeda dengan menu utama, halaman ini hanya memiliki menu File yang terdiri dari submenu Exit, dan menu Help yang terdiri dari submenu ProgramHelp dan About.

j. Form About

Form ini berisi tentang nama program dan pembuat program. Menu yang dimiliki halaman ini sama dengan menu pada form ProgramHelp, yaitu menu File yang terdiri dari submenu Exit, dan menu help yang terdiri dari submenu ProgramHelp dan About.



Gambar 3.29 Form About

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1. Implementasi Proses Pengenalan Pola

4.1.1. Proses Memasukkan Data Pola Angka

Pada proses memasukkan Pola Uji pola angka 0, 1, 2 ...sampai 9 disimpan dalam suatu variabel yang disebut pola uji. Pola uji tersebut digunakan untuk menguji tingkat keberhasilan pola angka 0 sampai 9 menggunakan metode Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

```

...
[nama_file1,nama_path1] = uigetfile('*.bmp','Input Pola Uji');

global jml;
if isempty(jml)
    jml=0;
end
if ~isequal(nama_file1, 0)
    jml=jml+1;
    data=imread(fullfile(nama_path1,nama_file1));
    handles.data=imresize(data,[30 30]);
    guidata(hObject,handles);
    handles.current_data=handles.data;
    global uji;
    uji(1:30,1:30,jml)=handles.data;
    subplot(6,5,jml+5),imshow(uji(1:30,1:30,jml))
else
    return;
end
...

```

File yang bisa dimasukkan dalam program ini hanyalah file *.bmp 1 bit (*Biner*). Setelah diambil dari direktori tertentu maka pola yang dipilih tersebut akan ditampilkan. Pada potongan program di atas proses penampilan pola menggunakan subplot. Selain ditampilkan pola tersebut disimpan dalam sebuah

variabel yang berupa matrik 3 dimensi bernama uji. Variabel uji ini bertipe global sehingga dapat diakses di manapun dalam bagian program. Untuk dapat merangkaikan pola-pola tersebut menjadi satu dalam variabel uji maka ukuran pola tersebut harus diubah menjadi ukuran tertentu, dalam hal ini 30 X 30 pixel. Variabel jml menyatakan jumlah pola uji yang telah dimasukkan yang juga menyatakan panjang nilai z dalam matriks 3 dimensi uji, variabel ini juga bertipe global. Proses yang sama juga terjadi saat memasukkan pola *Template*.

4.1.2. Proses Menampilkan Data Pola Angka

Proses ini berfungsi untuk menampilkan data Pola *Template*, Pola Uji, hasil *Preprocessing* Pola *Template*, dan hasil *Preprocessing* Pola Uji.

```

...
global uji;
[br kl lb]=size(uji);
axes(handles.axes1);
for ris=1:lb
    subplot(6,5,ris+10),imshow(uji(:,:,ris))
end
...

```

Penggalan program diatas merupakan code untuk menampilkan pola uji yang disimpan dalam variabel uji. Untuk menampilkan pola-pola yang lain tinggal mengganti variable uji di atas dengan variabel penyimpan pola yang akan ditampilkan.

4.1.3. Proses Menghapus Data Pola Angka

Pada proses ini data pola yang tersimpan di dalam variabel penyimpan dapat dihapus. Proses tersebut dilakukan dengan mengosongkan variabel penyimpan pola dan variabel jml yang menyimpan jumlah pola.


```

...
global uji;
uji=[];
global jml;
jml=0;
...

```

4.1.4. *Preprocessing*

Pada proses ini pola input yang berupa file gambar dengan format grafik BMP-1 bit (*Biner*) menjadi gambar berukuran 30 x 30 pixel. Sebelum dilakukan pengubahan ukuran, obyek angka pada gambar disesuaikan posisinya agar obyek memenuhi gambar. Penyesuaian tersebut dilakukan dengan memotong daerah yang kosong. Asumsi untuk daerah pola angka yang digunakan untuk masukan pada jaringan Kohonen pada penelitian ini adalah 0.

Gambar yang telah disesuaikan ukuran dan nilai pixelnya tersebut yang berupa matriks diubah menjadi bentuk vektor. Vektor tersebut memiliki anggota sebanyak 30 x 30 pixel yaitu 900 pixel, dengan penyusunan baris pertama matriks dilanjutkan baris ke-dua, ke-tiga, ke-empat, sampai baris ke-tiga puluh matriks.

```

...
function [pola,pola_tampil]=preprocessing2(latih)

[x y z]=size(latih);
for k=1:z
    pola1=latih(:,:,k);
    gambar = potong(pola1);

    pola2=jadi_vektor(gambar);
    pola(k,1:900)=pola2(1:900);
    pola_tampil(:,:,k)=gambar;
end
pola;
pola_tampil;
...

```

```
function pola=potong(pola1)
```

```
[m,n]=size(pola1);
v=zeros(1,n)+1;
tamp=0;
p=[];
```

```
%potong dari atas
```

```
for i=1:m
    p=pola1(i,:);
    cek=sama(v,p);
    if cek==1
        tamp=tamp+1;
    else
        break;
    end
end
pola1(1:tamp,:)=[];
```

```
%potong dari bawah
```

```
[m,n]=size(pola1);
tamp=0;
p=[];
for i=m:-1:1
    p=pola1(i,:);
    cek=sama(v,p);
    if cek==1
        tamp=tamp+1;
    else
        break;
    end
end
indek=m-tamp;
pola1(indek:m,:)=[];
```

```
%potong dari kiri
```

```
[m,n]=size(pola1);
v=zeros(m,1)+1;
tamp=0;
p=[];
for i=1:n
    p=pola1(:,i);
    cek=sama(v,p);
    if cek==1
        tamp=tamp+1;
    else
        break;
    end
end
pola1(:,1:tamp)=[];
```

```

%potong dari kanan
[m,n]=size(polal);
v=zeros(m,1)+1;

tamp=0;
p=[];
for i=n:-1:1
    p=polal(:,i);
    cek=sama(v,p);
    if cek==1
        tamp=tamp+1;
    else
        break;
    end
end
indek=n-tamp;
polal(:,indek:n)=[];

%ubah jadi ukuran 30 x 30 pixel
pola=imresize(polal,[30 30]);
...

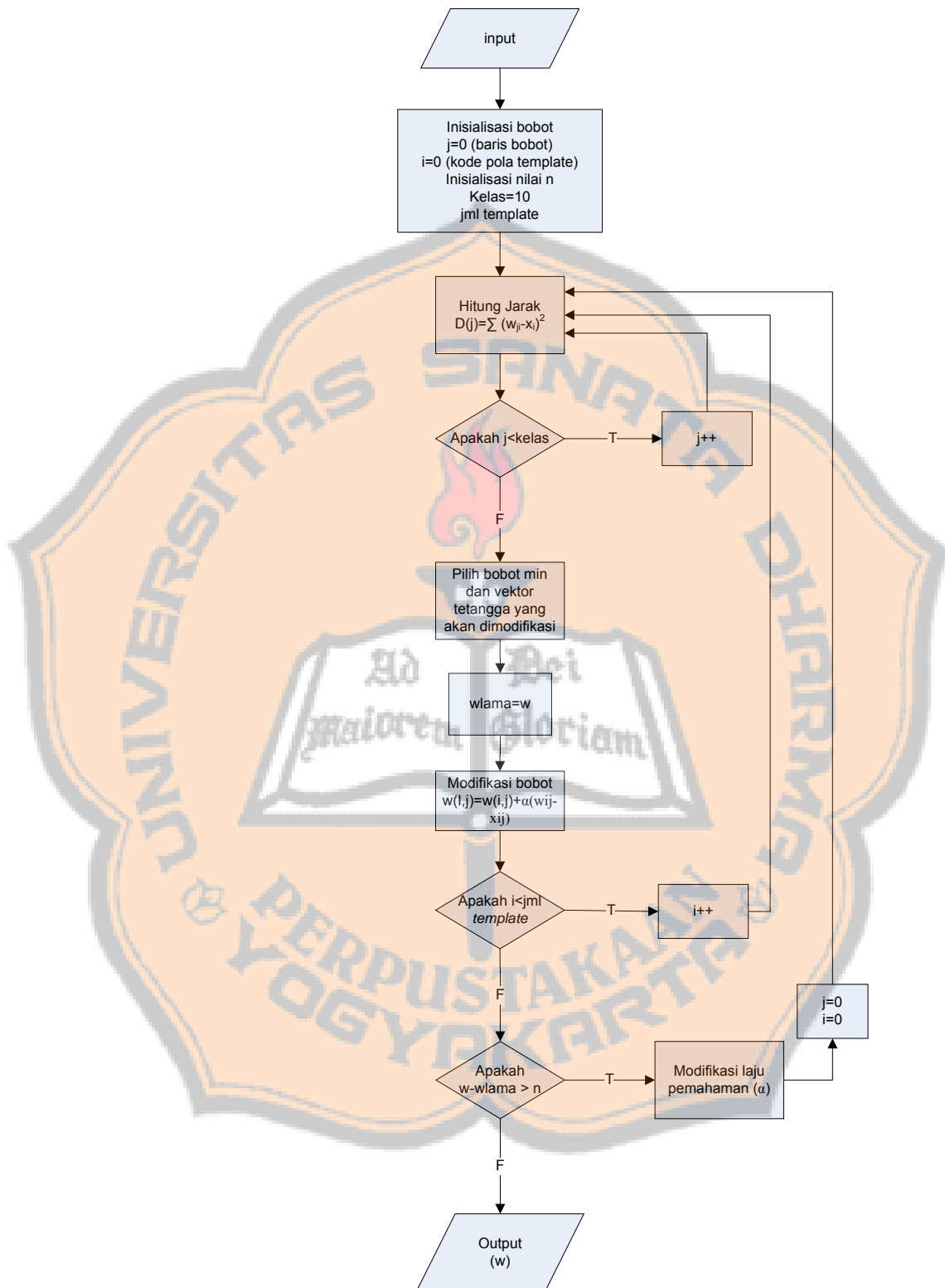
function v=jadi_vektor(a)
[m,n]=size(a);
v=[];
r=1;
for i=1:m
    for j=1:n
        v(1,r)=a(i,j);
        r=r+1;
    end
end
v;

```

4.1.5. Pelatihan

Proses pelatihan bertujuan untuk melatih jaringan Kohonen yaitu pada matriks bobotnya yang pada awalnya diinisialisasi sebagai matriks *Random* dengan jumlah baris sebanyak jumlah kelas dan jumlah kolom sebanyak jumlah elemen vektor input sampai menghasilkan matriks bobot yang paling optimal. Matriks bobot optimal inilah yang digunakan untuk klasifikasi pola.

Diagram proses pelatihan jaringan Kohonen ini dapat dilihat pada *Flowchart* berikut ini.



Gambar 4.1 Flowchart proses pelatihan

Masukan (*Input*) dari proses pelatihan ini adalah vektor hasil *Preprocessing*, Laju Pemahaman (*alpha*), koefisien penurunan *alpha*, topologi jaringan, jari-jari vektor pemenang, dan jumlah maksimum epoch. Jumlah vektor yang dimasukkan sebanyak 50 vektor yang terdiri dari vektor pola angka 0 sampai 9 dengan masing-masing angka terdiri dari 5 buah pola.

Pada tahap inisialisasi dibentuk matriks bobot dengan jumlah baris merepresentasikan jumlah kelas yang akan dibentuk, jumlah kolomnya sama dengan jumlah komponen di dalam vektor masukan, dan memiliki komponen bilangan *Random*. Selain matriks bobot, beberapa variabel yang diinisialisasi pada proses ini adalah variabel *j* yang digunakan untuk merekam kode kelas, variabel *i* yang digunakan untuk merekam kode pola *Template*, nilai variabel *n* yang digunakan sebagai parameter pengujian keoptimalan matriks bobot, jumlah kelas ditetapkan sebanyak sepuluh kelas karena pola yang dikenali adalah pola 0 sampai 9, dan inisialisasi jumlah *Template* sesuai dengan masukan. Nilai variabel *n* yang digunakan dalam penelitian ini adalah 0.

Proses penghitungan jarak dilakukan untuk semua pola *Template*, dengan

rumus :

$$D(j) = \sum_{i=1}^{900} (w_{ji} - x_i)^2$$

...

```
function d=jarak(w,b)
[m,n]=size(b);
for i=1:m
    d(i,1)=0;
    for j=1:n
        d(i,1)=d(i,1)+((b(i,j)-w(1,j))^2);
    end
end
d;
```

Pada setiap penghitungan jarak sebuah vektor input dengan matriks bobot, dicari baris bobot yang memiliki jarak terpendek dengan vektor input dan baris bobot tetangga sesuai dengan jari-jari dan topologi yang dimasukkan untuk dimodifikasi¹, dengan rumus :

$$w_{ji}^{baru} = w_{ji}^{lama} + \alpha(x_i - w_{ji}^{lama})$$

Setelah semua proses di atas dilakukan pada semua pola *Template* maka akan dilakukan pengujian apakah selisih bobot baru dengan bobot lama lebih dari nilai variable n. Jika syarat tersebut memenuhi maka semua proses pelatihan dilakukan kembali dengan memproses lagi semua pola *Template* dengan bobot baru hasil dari proses sebelumnya dan menggunakan Laju Pemahaman yang sudah dimodifikasi. Satu iterasi pemrosesan semua pola *Template* disebut satu *Epoch*.

Jika selisih bobot terbaru dengan bobot dari *Epoch* sebelumnya tidak lebih besar dari nilai variabel n maka iterasi dihentikan dan bobot yang dihasilkan adalah bobot yang paling optimal pada pelatihan tersebut.

```
...
global popmaks;
if popmaks == 0
    maks_epoch=100;
else
    maks_epoch=handles.maks_epoch;
end

wait=waitbar(0,'Please wait ...');
%untuk tampil epoch
global editalpha;
global editkof;
global popjari;
global poptopologi;
```

¹ Penentuan vektor bobot yang dimodifikasi akan dibahas lebih lanjut di Proses penentuan vektor tetangga yang dimodifikasi.


```

waitbar(12/mak)

[xt1,yt1,zt1]=grafik_w(b);

waitbar(20/mak)
ul=20;
set(handles.text6,'String','Jumlah Epoch');
global batas;
batas=0;
while cek==1

    set(handles.tunggu_text,'String','Program sekarang
melakukan Proses Pelatihan Jaringan. Mohon Tunggu...');
set(handles.epoch_text,'String',epoch);

%=====satu epoch_text =====
[klasifikasi,tampil_klas]=pengenalan(b,x);

for e=1:jmlh
    k=[];
    in=x(e,:);
    d=jarak(in,b);
    k=min(d);
    [brs,klm]=size(d);
    for i=1:brs
        if k==d(i,1)
            kelas=i;
            break;
        end
    end
    kelas;

    if topologi == 1
        b=modif_bobot2(b,alpha,x(e,:),kelas,jari);
    elseif topologi == 2
        b=modif_bobot3(b,alpha,x(e,:),kelas,jari);
    else
        b=modif_bobot4(b,alpha,x(e,:),kelas,jari);
    end
end

%=====pengecekan=====
selisih=abs(b-b0);
cek1=max(max(selisih));
cek=cek1>to;

alpha=modif_alpha(alpha,turun);

ul=ul+1;
waitbar(ul/mak)

b0=b;
epoch=epoch+1;
global batas;

```



```

        if batas == maks_epoch
            break;
        end
        batas=batas+1;
    end
    %untuk tampil bobot
    set(handles.text13,'Visible','on');
    set(handles.text14,'Visible','on');
    set(handles.hasil_axes,'Visible','on');
    set(handles.hasil2_axes,'Visible','on');

        set(handles.tunggu_text,'String','Proses Pelatihan
        Jaringan selesai. Sedang menyiapkan gambar
        bobot...');
        axes(handles.hasil_axes);
        set(handles.tunggu_text,'Visible','off');
        plot3(xt1,yt1,zt1,'-bo','LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','r',...
            'MarkerSize',10);

        [xt,yt,zt]=grafik_w(b);
        axes(handles.hasil2_axes);
        plot3(xt,yt,zt,'-bo','LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','r',...
            'MarkerSize',10);

    ul=mak-2;
    waitbar(ul/mak)
    waitbar(mak/mak)
    close(wait)

end
end
end
...

```

Proses Penentuan Vektor yang dimodifikasi

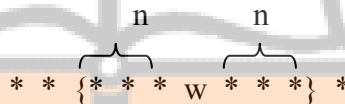
Pada setiap penghitungan jarak sebuah vektor input dengan matriks bobot, dicari baris bobot yang memiliki jarak terpendek dengan vektor input dan baris bobot tetangga sesuai dengan jari-jari dan topologi yang dimasukkan untuk dimodifikasi. Terdapat tiga topologi yang digunakan untuk menentukan vektor tetangga yang akan dimodifikasi, yaitu topologi linear, bujursangkar, dan heksagonal.

a. Topologi Linear

```
function g=modif_bobot2(b,a,x,kelas,jari)

%modif bobot untuk topologi linear
[m,n]=size(b);
r=jari;
j=1+(2*r);
g=b;
kelas=kelas-r;
for r=1:j
if kelas > 0 & kelas < m+1
for i=1:n
g(kelas,i)=b(kelas,i)+(a*(x(1,i)-b(kelas,i)));
end
end
end
kelas=kelas+1;
end
g;
```

Pada penggalan program di atas diasumsikan bahwa penyusunan kelas berada pada bentuk linear / vektor. Apabila diketahui jari-jari bernilai n dan vektor pemenangnya adalah w maka vektor tetangga yang juga dimodifikasi adalah vektor mulai dari $w-n$ sampai $w+n$.



Gambar 4.2 Topologi linear

b. Topologi Bujursangkar

```
function g=modif_bobot3(b,a,x,kelas,jari)
%modif bobot untuk topologi bujursangkar
[m,n]=size(b);
topo=[1 2 3 4 5;6 7 8 9 10];
[baris,kolom]=find(topo == kelas);
g=b;
if jari > 0
% fprintf('\nMASUK KELAS JARI > 0')
tamp1=kolom-jari;
tamp2=kolom+jari;
if tamp1 < 1
tamp1=1;
end
if tamp2 > 5
tamp2=5;
end
```

```

for br=1:2
    for kl=tamp1:tamp2
        kel=topo(br,kl);
        for i=1:n
            g(kel,i)=b(kel,i)+(a*(x(1,i)-b(kel,i)));
        end
    end
end

else
%   fprintf('\nMASUK jari = 0')
for i=1:n
    g(kelas,i)=b(kelas,i)+(a*(x(1,i)-b(kelas,i)));
end
end
g;

```

Pada topologi ini apabila jari-jari yang dimasukkan adalah 0 maka yang dimodifikasi hanyalah vektor pemenang. Sedangkan untuk jari-jari 1 lebih maka yang dimodifikasi adalah vektor pemenang dan vektor tetangganya. Proses menentukan vektor tetangga adalah dengan menggunakan sebuah matriks pembantu berukuran 2 X 5 dengan susunan seperti pada gambar dibawah ini.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

Gambar 4.3 Matriks pembantu pada topologi bujursangkar

Langkah awal yang dilakukan adalah mencari posisi kelas pemenang pada matriks pembantu. Apabila diketahui bahwa jari-jari bernilai n , dan posisi vektor pemenang adalah (x,y) . Maka vektor tetangganya adalah vektor pada posisi $(x-n,y-n)$ sampai $(x+n,y+n)$.

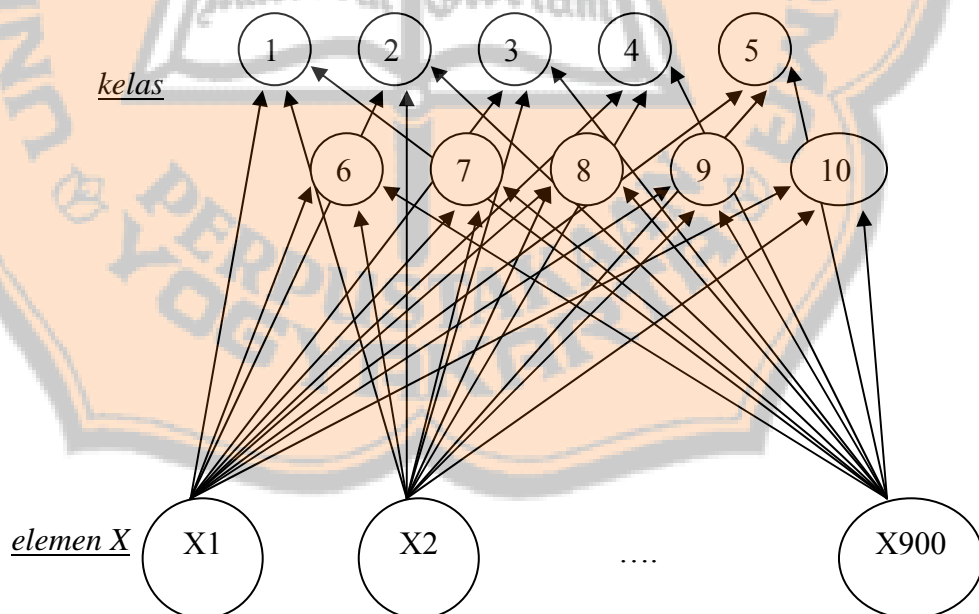
c. Topologi Heksagonal

Penyusunan kelas pada topologi heksagonal ini juga dalam bentuk 2 dimensi atau matriks, yaitu matriks 2 baris dan 5 kolom. Hanya saja dalam penyusunannya kita asumsikan sebagai berikut :

1	2	3	4	5
6	7	8	9	10

Gambar 4.4 Susunan kelompok pada topologi Heksagonal

Seperti topologi yang lain, setiap kelas pada matriks kelas terhubung dengan vektor input (elemen X). Setiap keterhubungan tersebut memiliki nilai yang disimpan dalam matriks bobot dengan letak baris sesuai dengan kelas yang dituju dan pada kolom sesuai dengan indek yang dimiliki X.



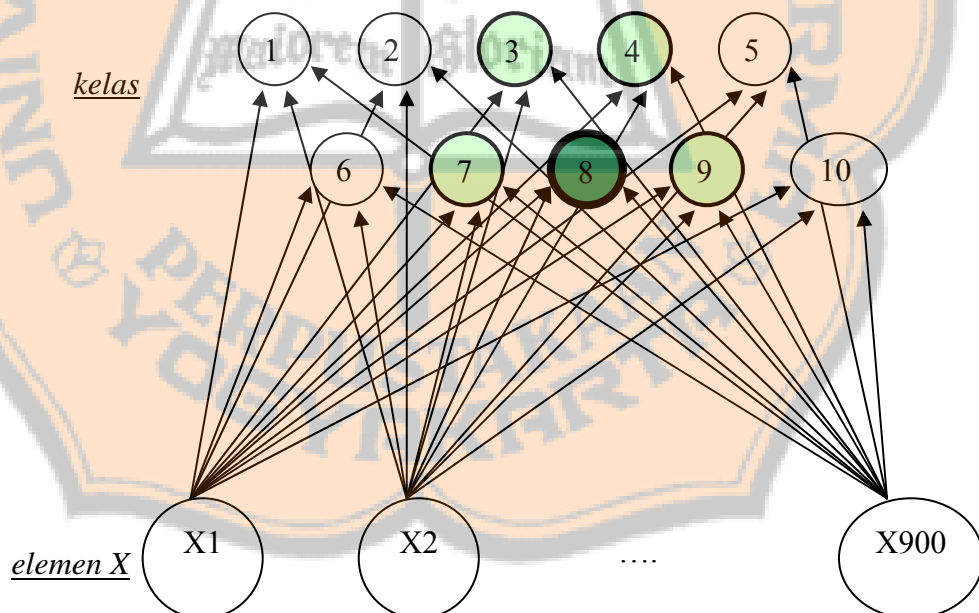
Gambar 4.5 Arsitektur jaringan kohonen topologi heksagonal

Matriks bobot tersebut terdiri dari 10 baris dan 900 kolom dimana setiap barisnya mewakili satu kelas.

$$\begin{pmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1900} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2900} \\ \dots & \dots & \dots & \dots & \dots \\ W_{101} & W_{102} & W_{103} & \dots & W_{10900} \end{pmatrix}$$


Gambar 4.6 Matriks bobot

Apabila diketahui jari-jari yang dimasukkan pada topologi bujur sangkar ini adalah 1. Vektor pemenang adalah vektor baris ke-8, maka vektor yang akan dimodifikasi bobotnya adalah vektor pemenang tersebut dan vektor tetangganya. Vektor tetangga dari kelas 8 tersebut adalah vektor di sekitar kelas 8, seperti terlihat pada gambar dibawah ini.



Gambar 4.7 Kelas yang dimodifikasi pada topologi heksagonal dengan jari-jari 1

Jadi, vektor tetangga yang dimodifikasi dengan jari-jari 1 adalah vektor baris ke-7, vektor baris ke-3, vektor baris ke-4, dan vektor baris ke-9.



```

W11 W12 W13 ... W1900
W21 W22 W23 ... W2900
W31 W32 W33 ... W3900
W41 W42 W43 ... W4900
W51 W52 W53 ... W5900
W61 W62 W63 ... W6900
W71 W72 W73 ... W7900
W81 W82 W83 ... W8900
W91 W92 W93 ... W9900
W101 W102 W103 ... W10900

```

Gambar 4.8 Baris vektor bobot yang dimodifikasi

```
function g=modif_bobot4(b,a,x,kelas,jari)
```

```
%modif bobot untuk topologi heksagonal
```

```
[m,n]=size(b);
```

```
topo=[1 2 3 4 5;6 7 8 9 10];
```

```
[baris,kolom]=find(topo == kelas);
```

```
g=b;
```

```
if jari > 0
```

```
% fprintf('\nMASUK KELAS JARI > 0')
```

```
tamp1=kolom-jari;
```

```
tamp2=kolom+jari;
```

```
if tamp1 < 1
```

```
tamp1=1;
```

```
end
```

```
if tamp2 > 5
```

```
tamp2=5;
```

```
end
```

```
if baris == 1
```

```
if (kolom-jari)>0
```

```
for br=1:2
```

```
if br == 2
```

```
tamp1=tamp1+1;
```

```
end
```

```
for kl=tamp1:tamp2
```

```
kel=topo(br,kl);
```

```

        for i=1:n
            g(kel,i)=b(kel,i)+(a*(x(1,i)-(kel,i)));
        end
    end
end

else
    for br=1:2
        for kl=tamp1:tamp2
            kel=topo(br,kl);
            for i=1:n
                g(kel,i)=b(kel,i)+(a*(x(1,i)-b(kel,i)));
            end
        end
    end
end

else
    if (kolom+jari) < 6
        for br=1:2
            if br == 1
                tamp2=tamp2-1;
            end
            for kl=tamp1:tamp2
                kel=topo(br,kl);
                for i=1:n
                    g(kel,i)=b(kel,i)+(a*(x(1,i)-b(kel,i)));
                end
            end
        end
    else
        for br=1:2
            for kl=tamp1:tamp2
                kel=topo(br,kl);
                for i=1:n
                    g(kel,i)=b(kel,i)+(a*(x(1,i)-b(kel,i)));
                end
            end
        end
    end
end

end

else
%   fprintf('\nMASUK jari = 0')
    for i=1:n
        g(kelas,i)=b(kelas,i)+(a*(x(1,i)-b(kelas,i)));
    end

end
g;

```

4.1.6. *Clustering*

Setelah dihasilkan bobot optimal dari proses pelatihan, maka untuk mengetahui identitas kelas dilakukan proses *Clustering*. Proses *Clustering* terdiri dari 2 tahap, yaitu klasifikasi pola *Template* dan penamaan² kelas. Pengklasifikasian pola *Template* dilakukan dengan menghitung jarak pola *Template* dengan bobot optimal yang sudah dihasilkan. Baris bobot yang menghasilkan nilai minimum adalah baris yang menyatakan kelas dari pola *Template* tersebut. Jika terdapat lebih dari satu nilai minimum maka disepakati bahwa pola *Template* tersebut termasuk di dalam kelas dengan indeks yang terkecil dari nilai minimum tersebut.

Pengelompokkan pola *Template* ini digunakan untuk menamai masing-masing kelas yang ditempati oleh pola-pola *Template*. Kelas yang memiliki anggota semua pola angka yang sama (lima pola angka) memiliki identitas angka tersebut. Sehingga pada saat pengujian, apabila ada pola uji yang masuk dalam kelas tersebut maka pola uji tersebut dikenali sebagai angka identitas dari kelas tersebut. Sedangkan untuk kelas yang memiliki lebih dari satu jenis pola *Template* atau sama sekali tidak memiliki anggota, tidak akan memiliki angka identitas. Apabila pada proses pengujian terdapat pola uji yang masuk ke dalam kelas ini maka pola tersebut termasuk pola yang tidak dikenali.

Penamaan kelas tersebut dilakukan dengan memasukkan identitas kelas dengan fasilitas poppupmenu yang telah disediakan disamping tampilan hasil *Clustering*.

² pemberian identitas kelas


```

global b;
global x;
global klas;
klas=uji_pola(b,x);
global hasil;
[hasil,b]=kelompok(klas,b);
set(handles.kel_text, 'String',hasil);
...

```

```

function klasifikasi=uji_pola(b,x)

```

```

[jml,kol]=size(x);
[brs,klm]=size(b);
for bar=1:brs
    klasifikasi(bar,1)=bar;
end
n=2;

    for e=1:jml
        k=[];
        input=x(e,:);
        d=jarak(input,b);
        k=min(d);
        [brs,klm]=size(d);
        for i=1:brs
            if k==d(i,1)
                kelas=i;
                break;
            end
        end
        klasifikasi(kelas,n)=e;
        n=n+1;
    end
klasifikasi;
...

```

```

function [hasil,bobot]=kelompok(klasifikasi,bobot)

```

```

[m n]=size(klasifikasi);
bt1=0;
bt2=6;

for i=1:m
    hasil(i,1:9)='kelompok ';
    hasil(i,10:11)=[64+i, ' '];
    hasil(i,12:14)=' = ';
    posisi=15;

    for j=2:n
        var=klasifikasi(i,j);
        if var~=0
            for cari=1:m
                if var>bt1 & bt2>var
                    var=bt1/5;
                    bt1=0;
                end
            end
        end
    end
end

```

```

        bt2=6;
        break;
    end
    bt1=bt1+5;
    bt2=bt2+5;
end

```

```

        kata=int2str(var);
        [b k]=size(kata);
        posisi_akhir=posisi+k;
        hasil(i,posisi:posisi_akhir)=[kata,' '];
        posisi=posisi_akhir+1;
    end
end
end
hasil;
...

```

4.1.7. Pengujian

Proses pengujian dilakukan dengan menggunakan pola-pola angka selain pola angka yang digunakan untuk proses pelatihan. Pola yang diuji adalah pola angka dari 0 sampai 9 yang berupa file dengan format grafik BMP-1 bit (*Biner*) yang telah dikenai *Preprocessing* yang sama dengan pola *Template* sehingga menjadi berbentuk vektor dengan anggota sebanyak 30 x 30 pixel yaitu 900 pixel. Vektor masukan tersebut kemudian dihitung jaraknya dengan bobot optimal yang dihasilkan pada proses pelatihan. Dari proses perhitungan jarak tersebut maka dapat diketahui kelas dari vektor masukan tersebut. Vektor masukan akan dikenali sebagai angka identitas dari kelas tersebut. Angka identitas kelas adalah angka yang diperoleh dari proses pengelompokkan pola *Template*.

```

...

global uji;
[x1 y1 z1]=size(uji);
global uji;
h=preprocessing2(uji);
for i=1:z1
    global b;

```

```

[temp,tampil]=pengenalan(b,h);
end

axes(handles.axes1);
for i=1:z1
    global uji;
    subplot(6,5,i+10),imshow(uji(:,:,i))
end

global hasil;
kenal=kenali2(hasil,temp);
[baris kolom]=size(kenal);

for j=1:baris
    jml=11+kolom;
    if j < 10

        cling(j,1:jml)=[ 'Inputke-',int2str(j),'
',kenal(j,1:kolom)];

    else

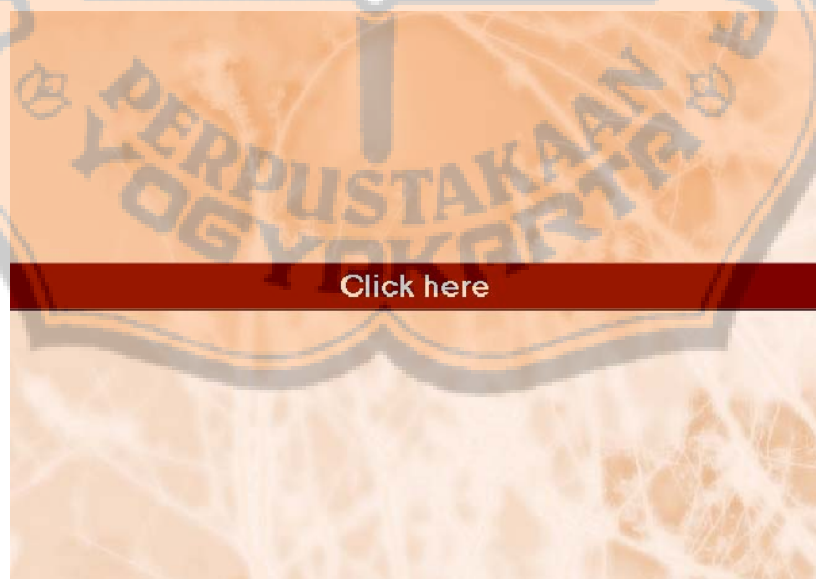
        cling(j,1:jml)=[ 'Input ke-',int2str(j),kenal(j,1:kolom)];

    end
end
set(handles.tampil_text,'String',cling)

```

4.2. Implementasi Interface Sistem Pengenalan Pola

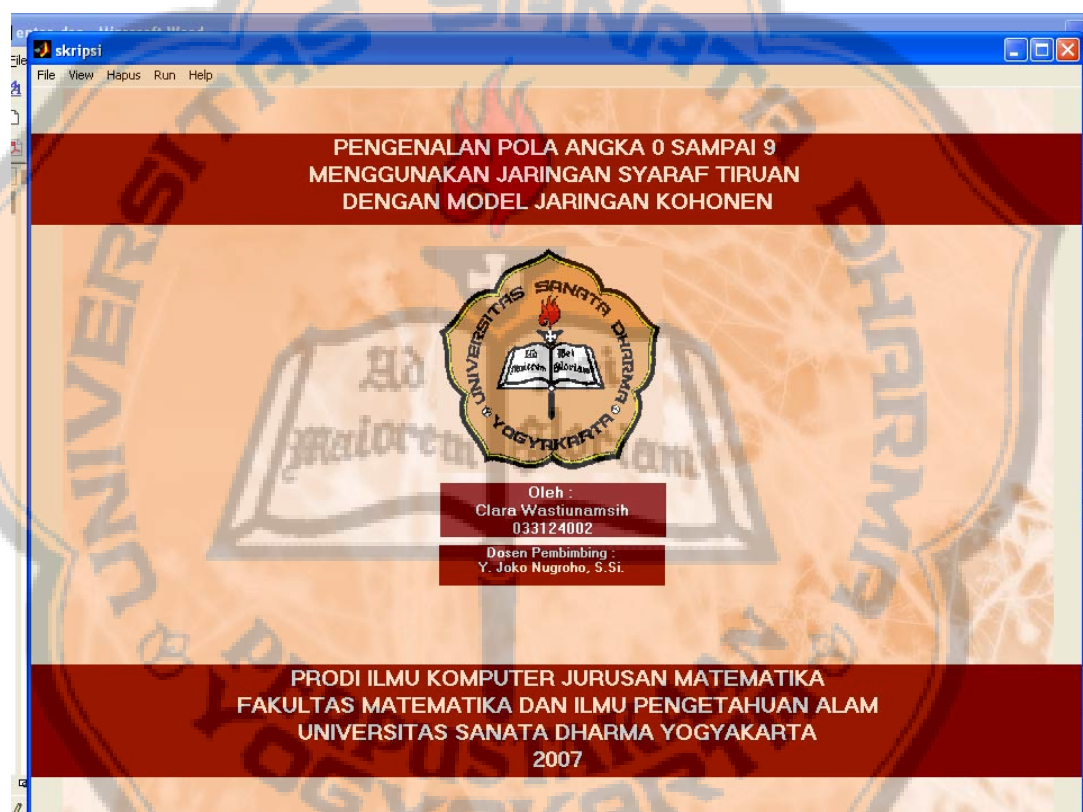
4.2.1. Form enter



Gambar 4.9 Form enter

Form ini merupakan 'pintu gerbang' dari program pengenalan pola angka ini. Form ini berfungsi untuk mengosongkan semua variabel global yang digunakan pada program. Tombol 'Clik here' akan mengantarkan user ke halaman Home yang merupakan halaman utama program Pengenalan Pola Angka Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen.

4.2.2. Form Home



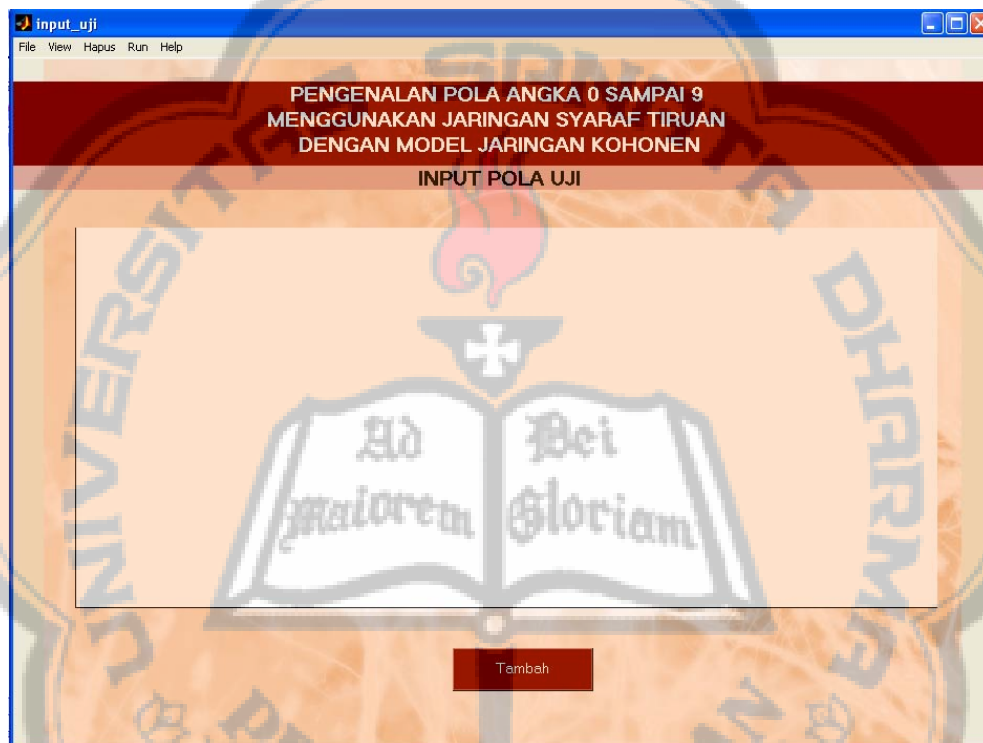
Gambar 4.10 Form home

Form ini merupakan halaman utama dari program Pengenalan Pola Angka Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen. Halaman utama ini menampilkan judul program, logo Universitas Sanata Dharma, nama pembuat, nama dosen pembimbing, Program Studi, Jurusan, Fakultas, dan

Universitas dari pembuat program. Pada form ini terdapat menu utama dengan rincian seperti penjelasan pada perancangan menu.

4.2.3. Form Input Pola

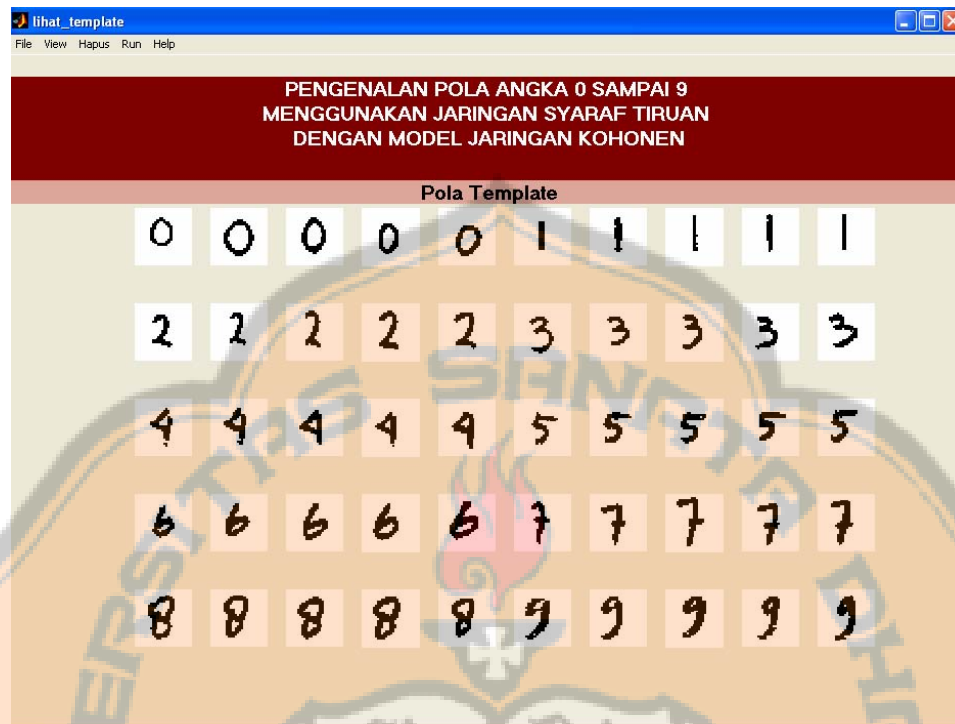
Form ini berfungsi untuk menginputkan pola uji. Pola uji diinputkan satu persatu dengan menekan tombol tambah dan akan langsung ditampilkan.



Gambar 4.11 Form Input Pola

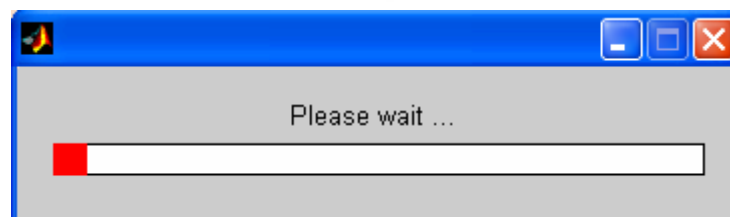
Tampilan untuk input Pola *Template* memiliki desain dan cara yang sama dengan form Input Pola Uji di atas.

4.2.4. Form Tampil Pola



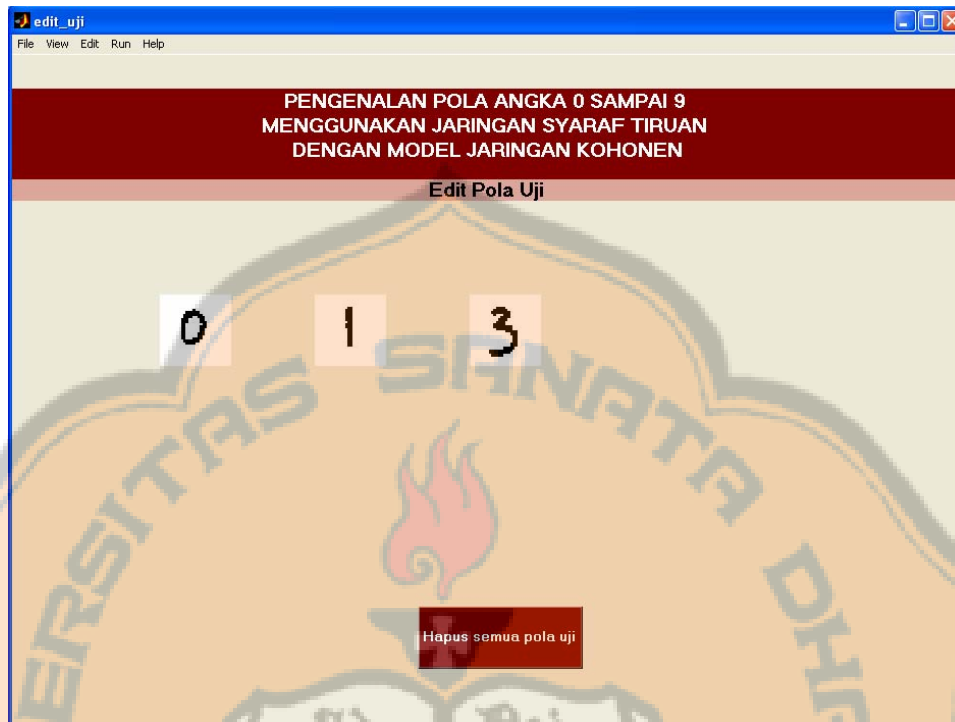
Gambar 4.12 Form Tampil Pola

Format form ini digunakan untuk menampilkan pola sesuai dengan menu yang dipilih, yaitu *Pola Template*, *Pola Uji*, hasil *Preprocessing* pola *Template*, atau hasil *Preprocessing* pola uji. Pada saat proses menampilkan pola *Template* dan hasil *Preprocessing* pola *Template*, ditampilkan waitbar dengan tulisan 'Please Wait' karena proses penampilannya membutuhkan waktu yang agak lama. Hal tersebut berfungsi untuk memberi informasi kepada user tentang lama proses penampilan sehingga user bersedia menunggu.



Gambar 4.13 Waitbar

4.2.5. Form Edit Pola



Gambar 4.14 Form Edit Pola

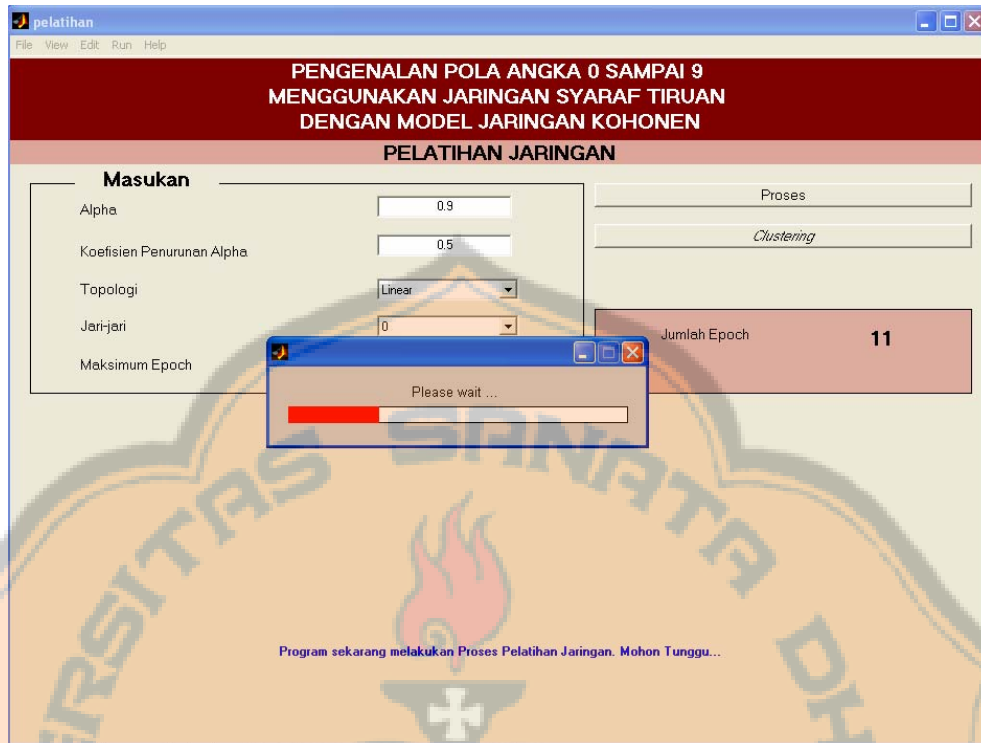
Form Edit Pola tersebut akan menampilkan semua pola. Pada form ini terdapat button Hapus Semua Pola, yang berfungsi untuk menghapus semua pola. Dengan menekan button ini maka semua pola akan dihapus kemudian program akan menampilkan form pola yang kosong.

4.2.6. Form Pelatihan

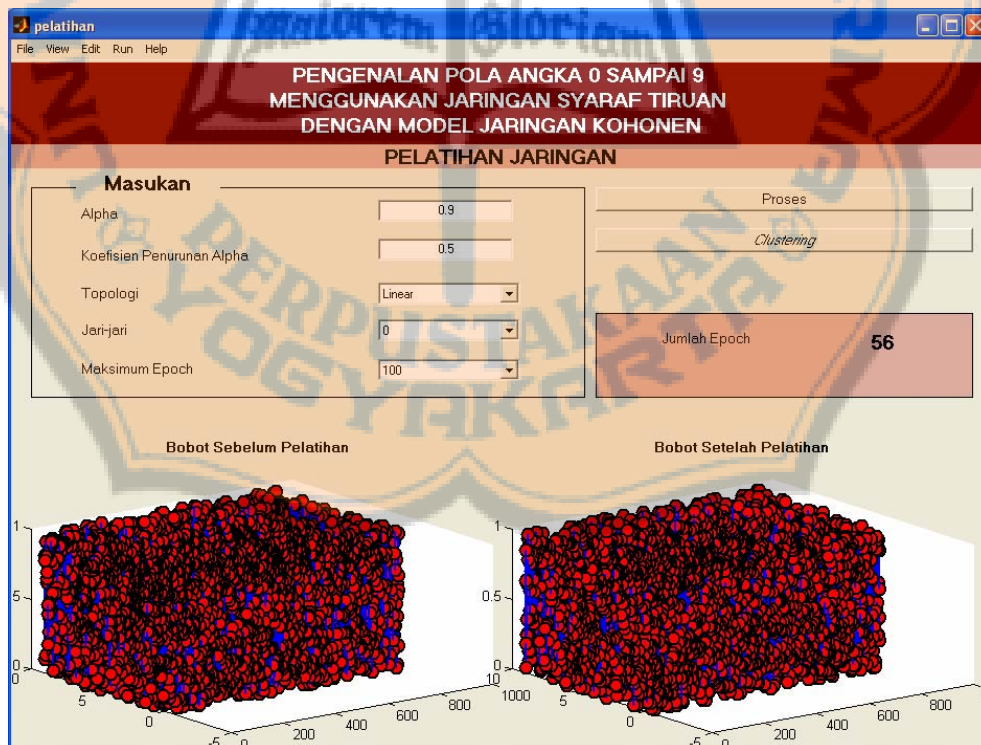
PELATIHAN JARINGAN	
Masukan	<input type="button" value="Proses"/>
Alpha	<input type="text"/>
Koefisien Penurunan Alpha	<input type="text"/>
Topologi	<input type="text" value="Linear"/>
Jari-jari	<input type="text" value="0"/>
Maksimum Epoch	<input type="text" value="100"/>
	<input type="button" value="Clustering"/>

Gambar 4.15 Form Pelatihan

Gambar 4.15 menampilkan form pelatihan yang masih kosong. Sedangkan gambar 4.16 menampilkan form pelatihan pada saat proses pelatihan berlangsung. Setelah proses pelatihan selesai, program akan menampilkan gambar bobot sebelum pelatihan dan sesudah pelatihan, hal ini dapat dilihat pada gambar 4.17.



Gambar 4.16 Form Pelatihan



Gambar 4.17 Form Pelatihan

Pada form ini dilakukan pelatihan jaringan dengan memasukkan nilai alpha, koefisien penurunan alpha, jari-jari, topologi, dan maximum epoch yang dimasukkan oleh user. Nilai alpha dan koefisien penurunannya harus berupa nilai antara 0 dan 1. Jika user memasukkan nilai diluar range tersebut maka akan tampil warning yang menampilkan informasi nilai yang seharusnya dimasukkan. Untuk memasukkan jari-jari tinggal memilih salah satu nilai pada popmenu nilai jari-jari. Jika user tidak memilih popmenu ini maka nilai default jari-jari adalah 0. Begitu pula untuk topologi dan maximum *Epoch*, jika user tidak memilih popmenu tersebut maka defaultnya adalah sesuai dengan yang ditampilkan pada popmenu tersebut.

Input-input tersebut akan digunakan untuk proses pelatihan yaitu dengan menekan tombol proses. Pada saat menekan tombol proses maka dilakukan proses *Preprocessing* pola *Template* yang telah dimasukkan di dalam program, pelatihan dengan hasil *Preprocessing* pola *Template* tersebut, serta masukan yang dimasukkan oleh user melalui form pelatihan tersebut. Pada saat proses berjalan akan ditampilkan waitbar, dan nilai *Epoch* yang sedang ditempuh. Ketika proses pelatihan berakhir, maka akan ditampilkan gambar bobot sebelum pelatihan dan setelah pelatihan. Sumbu x pada gambar ini menyatakan jumlah neuron input, sumbu y menyatakan jumlah kelas yang dibentuk, dan sumbu z menyatakan nilai bobot yaitu antara 0 dan 1. Proses akan berhenti ketika selisih bobot terbaru dengan bobot sebelumnya bernilai 0 atau ketika proses telah mencapai nilai maksimum *Epoch* yang telah dimasukkan sebelumnya.

Setelah proses ini selesai maka user harus menekan tombol *Clustering*, untuk menamai kelas-kelas tersebut berdasarkan bobot optimal yang telah diperoleh dari proses pelatihan yang baru saja dilakukan. Dengan demikian masing-masing kelas yang memenuhi syarat telah memiliki identitas angka.³ Pada saat menekan tombol *Clustering* ini maka program akan menampilkan form hasil pelatihan.

4.2.7. Form Hasil Pelatihan

The screenshot shows a web application window with the following content:

Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Tiruan Dengan Model Jaringan Kohonen

HASIL PELATIHAN JARINGAN

kelompok A =	41 42 43 44 45	id Kelas A	null
kelompok B =	26 27 28 29 30	id Kelas B	8
kelompok C =	46 47 48 49 50	id Kelas C	3
kelompok D =	31 32 33 34 35	id Kelas D	9
kelompok E =	11 12 13 14 15 37 38 39 40	id Kelas E	6
kelompok G =	21 22 23 24 25	id Kelas F	null
kelompok H =	6 7 8 9 10	id Kelas G	4
kelompok I =	16 17 18 19 20 36	id Kelas H	1
kelompok J =	1 2 3 4 5	id Kelas I	null
		id Kelas J	0

Back

Gambar 4.18 Form Hasil Pelatihan

Form ini menampilkan kelas beserta anggotanya. Kelas-kelas tersebut akan dinamai dengan huruf A, B, C, ...sampai J. Anggotanya berupa indeks dari pola *Template* yang digunakan pada proses pelatihan. Penamaan kelas dilakukan

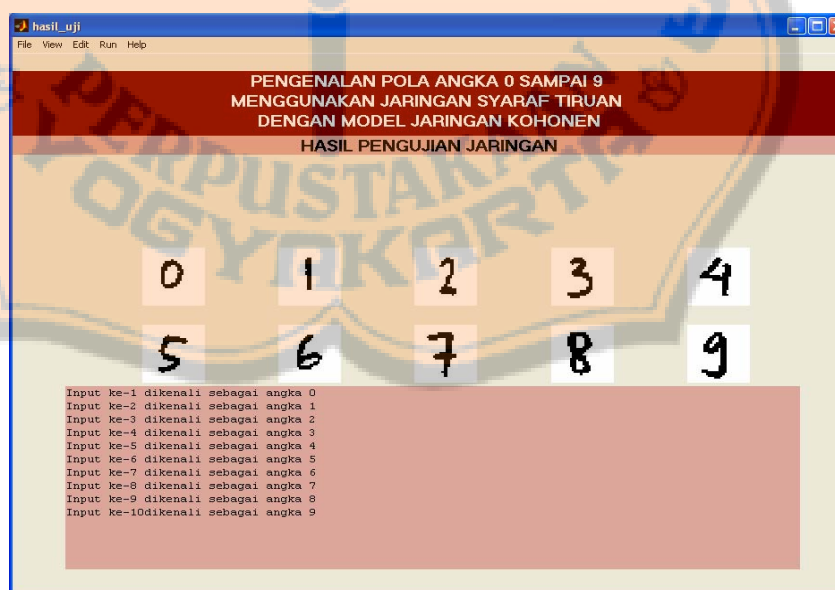
³ Lihat pada perancangan proses *clustering*.

dengan memasukkan identitas kelas melalui popupmenu yang disediakan disamping tampilan hasil *Clustering*. Pemberian identitas kelas berdasarkan hasil *Clustering* tersebut.

Tombol *Back* di bagian pojok kanan pada form ini akan membawa user ke form Home. Setelah itu user bisa melakukan proses pengujian dengan menggunakan pola uji.

4.2.8. Form Hasil Pengujian

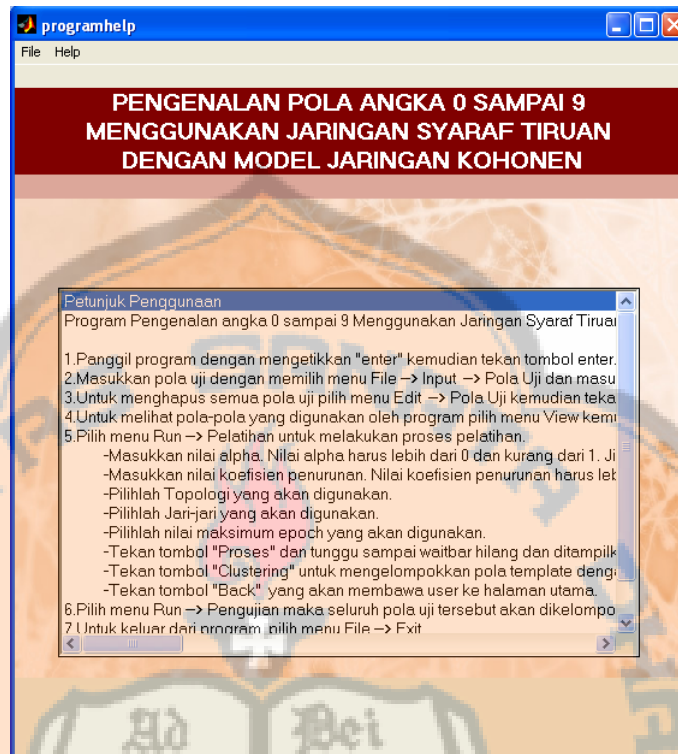
Form ini akan ditampilkan oleh program jika user sudah memasukkan pola uji, melakukan proses pelatihan, dan melakukan proses *Clustering*. Setelah syarat-syarat tersebut terpenuhi maka pada form ini akan ditampilkan semua pola uji, kemudian di bawahnya akan ditampilkan informasi tentang pola uji tersebut dikenali sebagai angkatertentu atau tidak dikenali. Keterangan pengenalan pola tersebut urut dari kiri ke kanan sebagai pola ke-1, ke-2, ke-3, ...dan seterusnya sesuai dengan jumlah pola uji.



Input	Dikenali sebagai
Input ke-1	angka 0
Input ke-2	angka 1
Input ke-3	angka 2
Input ke-4	angka 3
Input ke-5	angka 4
Input ke-6	angka 5
Input ke-7	angka 6
Input ke-8	angka 7
Input ke-9	angka 8
Input ke-10	angka 9

Gambar 4.19 Form Hasil Pengujian

4.2.9. Form ProgramHelp

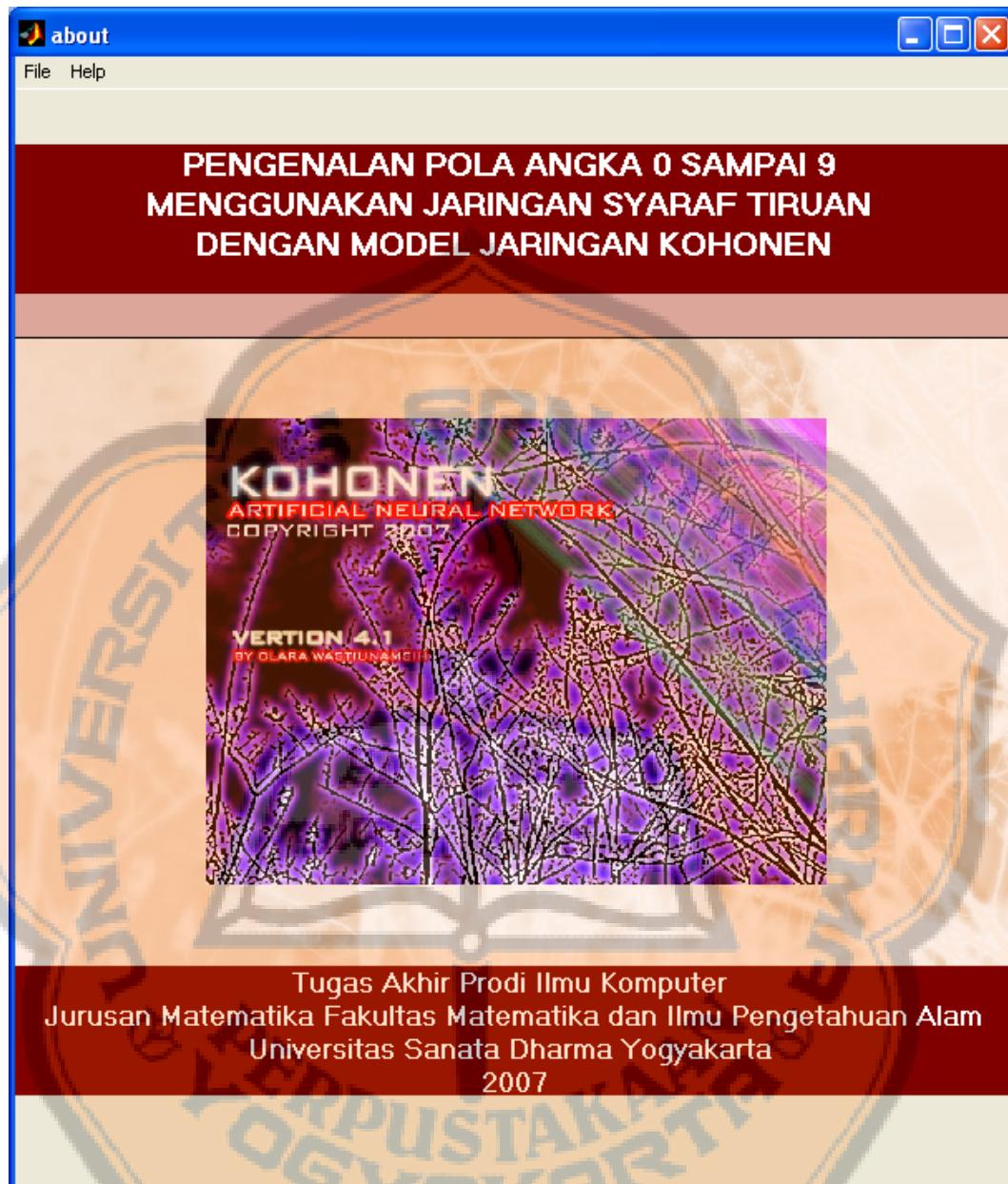


Gambar 4.20 Form ProgramHelp

Form ProgramHelp ini berisi tentang informasi cara pemakaian program dan tentang Pengenalan Pola Angka 0 sampai 9 Menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen. Menu yang dimiliki halaman ini berbeda dengan menu utama, halaman ini hanya memiliki menu file yang terdiri dari submenu exit, dan menu help yang terdiri dari submenu programHelp dan About.

4.2.10. Form About

Form ini berisi tentang nama program dan pembuat program. Menu yang dimiliki halaman ini sama dengan menu pada form programHelp, yaitu menu file yang terdiri dari submenu exit, dan menu help yang terdiri dari submenu programHelp dan About.



Gambar 4.21 Form About

4.3. Hasil Percobaan

Pengujian yang dilakukan pada sistem pengenalan pola angka ini menggunakan 20 buah pola uji yang terdiri dari dua kelompok angka 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9. Kelompok pertama merupakan angka hasil tulisan tangan orang yang sama dengan pola *Template* yang digunakan pada program ini, sedangkan kelompok yang lain adalah angka hasil tulisan tangan orang lain. Pola-pola tersebut berupa gambar bertipe .bmp 1 bit (*Biner*). Pola uji tersebut bukan pola yang digunakan untuk pelatihan. Masing-masing kelompok angka tersebut akan diuji pada hasil pelatihan menggunakan topologi linear, bujursangkar, dan heksagonal dengan jari-jari 0 dan jari-jari 1. Pada percobaan menggunakan jari-jari 0 hasil yang diperoleh pada semua topologi sama karena pada jari-jari 0 vektor bobot yang dimodifikasi hanyalah vektor pemenang saja. Pada setiap percobaan digunakan kombinasi nilai alpha dan koefisien penurunan alpha lebih dari 0 dan kurang dari 1. Pada percobaan ini dicobakan nilai alpha 0.1, 0.2, 0.3, ...,0.9 dan nilai koefisien penurunan alpha 0.1, 0.2, 0.3, ...,0.9. Masing-masing nilai alpha dan koefisien penurunan tersebut dikombinasikan untuk memperoleh hasil pengujian terbaik. Hasil akhir yang akan diketahui pada percobaan ini adalah nilai alpha, koefisien penurunan alpha, jari-jari dan topologi apa yang menghasilkan hasil pengenalan angka terbaik.

Dengan spek komputer Pentium 4 CPU 1.70GHz RAM 256 MB, proses Pelatihan membutuhkan waktu 01:12:32 untuk 50 buah pola template.

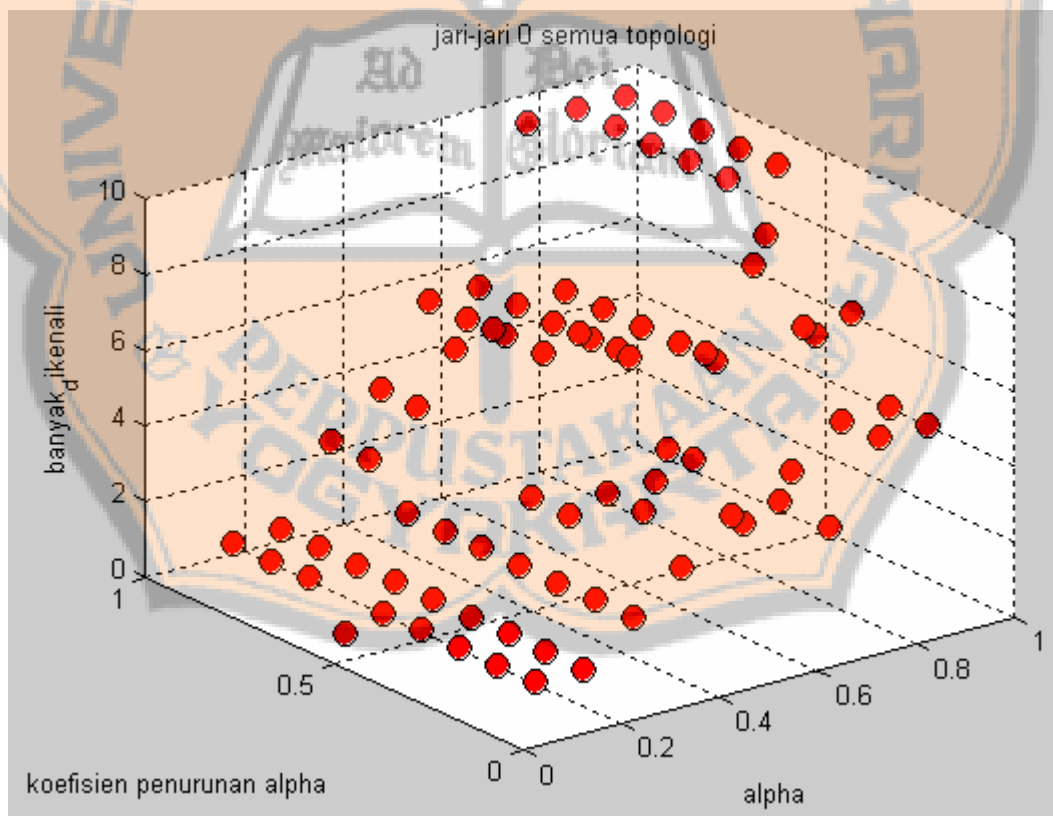
Tabel 1. Hasil percobaan dengan jari-jari 0 semua topologi

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	0	10	1	10%	17
2.	0.2	0.1	0	10	1	10%	18
3.	0.3	0.1	0	10	2	20%	18
4.	0.4	0.1	0	10	3	30%	19
5.	0.5	0.1	0	10	4	40%	18
6.	0.6	0.1	0	10	4	40%	18
7.	0.7	0.1	0	10	3	30%	18
8.	0.8	0.1	0	10	5	50%	18
9.	0.9	0.1	0	10	5	50%	18
10.	0.1	0.2	0	10	1	10%	24
11.	0.2	0.2	0	10	1	10%	24
12.	0.3	0.2	0	10	2	20%	25
13.	0.4	0.2	0	10	4	40%	25
14.	0.5	0.2	0	10	5	50%	25
15.	0.6	0.2	0	10	3	30%	25
16.	0.7	0.2	0	10	4	40%	25
17.	0.8	0.2	0	10	5	50%	25
18.	0.9	0.2	0	10	5	50%	25
19.	0.1	0.3	0	10	1	10%	31
20.	0.2	0.3	0	10	1	10%	33
21.	0.3	0.3	0	10	2	20%	33
22.	0.4	0.3	0	10	4	40%	33
23.	0.5	0.3	0	10	4	40%	33
24.	0.6	0.3	0	10	7	70%	33
25.	0.7	0.3	0	10	9	90%	33
26.	0.8	0.3	0	10	7	70%	33
27.	0.9	0.3	0	10	7	70%	33
28.	0.1	0.4	0	10	1	10%	42
29.	0.2	0.4	0	10	1	10%	43
30.	0.3	0.4	0	10	2	20%	43
31.	0.4	0.4	0	10	3	30%	43
32.	0.5	0.4	0	10	7	70%	43
33.	0.6	0.4	0	10	4	40%	43
34.	0.7	0.4	0	10	6	60%	43
35.	0.8	0.4	0	10	9	90%	42
36.	0.9	0.4	0	10	6	60%	44

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	0	10	1	10%	54
38.	0.2	0.5	0	10	1	10%	56
39.	0.3	0.5	0	10	2	20%	57
40.	0.4	0.5	0	10	3	30%	56
41.	0.5	0.5	0	10	7	70%	56
42.	0.6	0.5	0	10	6	60%	57
43.	0.7	0.5	0	10	6	60%	57
44.	0.8	0.5	0	10	10	100%	56
45.	0.9	0.5	0	10	10	100%	56
46.	0.1	0.6	0	10	0	0%	75
47.	0.2	0.6	0	10	1	10%	76
48.	0.3	0.6	0	10	2	20%	77
49.	0.4	0.6	0	10	7	70%	76
50.	0.5	0.6	0	10	6	60%	76
51.	0.6	0.6	0	10	6	60%	57
52.	0.7	0.6	0	10	6	60%	77
53.	0.8	0.6	0	10	10	100%	76
54.	0.9	0.6	0	10	10	100%	76
55.	0.1	0.7	0	10	1	10%	105
56.	0.2	0.7	0	10	1	10%	109
57.	0.3	0.7	0	10	2	20%	110
58.	0.4	0.7	0	10	6	60%	108
59.	0.5	0.7	0	10	6	60%	109
60.	0.6	0.7	0	10	6	60%	110
61.	0.7	0.7	0	10	6	60%	110
62.	0.8	0.7	0	10	10	100%	108
63.	0.9	0.7	0	10	10	100%	109
64.	0.1	0.8	0	10	1	10%	170
65.	0.2	0.8	0	10	1	10%	173
66.	0.3	0.8	0	10	3	30%	175
67.	0.4	0.8	0	10	4	40%	173
68.	0.5	0.8	0	10	6	60%	174
69.	0.6	0.8	0	10	6	60%	175
70.	0.7	0.8	0	10	6	60%	175
71.	0.8	0.8	0	10	10	100%	173
72.	0.9	0.8	0	10	10	100%	173

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	0	10	1	10%	353
74.	0.2	0.9	0	10	1	10%	366
75.	0.3	0.9	0	10	3	30%	363
76.	0.4	0.9	0	10	4	40%	366
77.	0.5	0.9	0	10	6	60%	368
78.	0.6	0.9	0	10	6	60%	370
79.	0.7	0.9	0	10	10	100%	364
80.	0.8	0.9	0	10	10	100%	365
81.	0.9	0.9	0	10	10	100%	366

Berdasarkan percobaan di atas diperoleh grafik sebagai berikut :



Gambar 4.22 Grafik percobaan dengan jari-jari 0 semua topologi

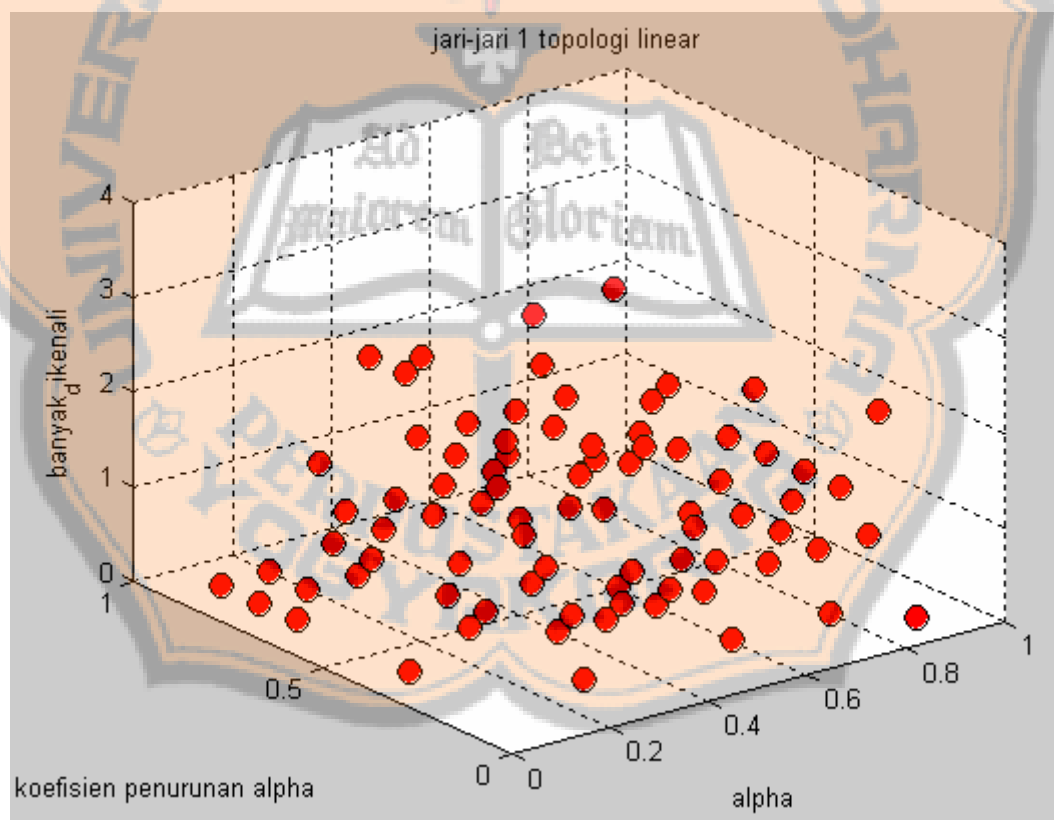
Tabel 2. Hasil percobaan dengan jari-jari 1 pada topologi linier

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	1	10	2	20%	18
2.	0.2	0.1	1	10	1	10%	18
3	0.3	0.1	1	10	1	10%	19
4.	0.4	0.1	1	10	1	10%	18
5.	0.5	0.1	1	10	2	20%	19
6.	0.6	0.1	1	10	1	10%	19
7.	0.7	0.1	1	10	1	10%	20
8.	0.8	0.1	1	10	1	10%	20
9.	0.9	0.1	1	10	0	0%	19
10.	0.1	0.2	1	10	1	10%	24
11.	0.2	0.2	1	10	4	40%	25
12.	0.3	0.2	1	10	0	0%	26
13.	0.4	0.2	1	10	1	10%	26
14.	0.5	0.2	1	10	1	10%	25
15.	0.6	0.2	1	10	0	0%	26
16.	0.7	0.2	1	10	1	10%	27
17.	0.8	0.2	1	10	0	0%	26
18.	0.9	0.2	1	10	2	20%	26
19.	0.1	0.3	1	10	1	10%	33
20.	0.2	0.3	1	10	2	20%	34
21	0.3	0.3	1	10	1	10%	34
22.	0.4	0.3	1	10	2	20%	34
23.	0.5	0.3	1	10	2	20%	34
24.	0.6	0.3	1	10	1	10%	35
25.	0.7	0.3	1	10	1	10%	35
26.	0.8	0.3	1	10	1	10%	37
27.	0.9	0.3	1	10	1	10%	36
28.	0.1	0.4	1	10	0	0%	43
29.	0.2	0.4	1	10	1	10%	43
30.	0.3	0.4	1	10	2	20%	43
31.	0.4	0.4	1	10	0	0%	45
32.	0.5	0.4	1	10	0	0%	44
33.	0.6	0.4	1	10	0	0%	45
34.	0.7	0.4	1	10	0	0%	45
35.	0.8	0.4	1	10	2	20%	47
36.	0.9	0.4	1	10	1	10%	48

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	1	10	1	10%	57
38.	0.2	0.5	1	10	3	30%	56
39.	0.3	0.5	1	10	0	0%	57
40.	0.4	0.5	1	10	1	10%	57
41.	0.5	0.5	1	10	1	10%	60
42.	0.6	0.5	1	10	0	0%	58
43.	0.7	0.5	1	10	2	20%	59
44.	0.8	0.5	1	10	0	0%	65
45.	0.9	0.5	1	10	1	10%	58
46.	0.1	0.6	1	10	1	10%	77
47.	0.2	0.6	1	10	1	10%	80
48.	0.3	0.6	1	10	1	10%	80
49.	0.4	0.6	1	10	1	10%	77
50.	0.5	0.6	1	10	0	0%	79
51.	0.6	0.6	1	10	1	10%	80
52.	0.7	0.6	1	10	1	10%	78
53.	0.8	0.6	1	10	1	10%	79
54.	0.9	0.6	1	10	1	10%	82
55.	0.1	0.7	1	10	0	0%	111
56.	0.2	0.7	1	10	1	10%	120
57.	0.3	0.7	1	10	1	10%	110
58.	0.4	0.7	1	10	1	10%	114
59.	0.5	0.7	1	10	1	10%	111
60.	0.6	0.7	1	10	2	20%	114
61.	0.7	0.7	1	10	1	10%	122
62.	0.8	0.7	1	10	1	10%	112
63.	0.9	0.7	1	10	0	0%	115
64.	0.1	0.8	1	10	0	0%	173
65.	0.2	0.8	1	10	0	0%	180
66.	0.3	0.8	1	10	0	0%	178
67.	0.4	0.8	1	10	2	20%	186
68.	0.5	0.8	1	10	1	10%	184
69.	0.6	0.8	1	10	1	10%	178
70.	0.7	0.8	1	10	1	10%	179
71.	0.8	0.8	1	10	0	0%	179
72.	0.9	0.8	1	10	1	10%	183

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	1	10	0	0%	360
74.	0.2	0.9	1	10	0	0%	373
75.	0.3	0.9	1	10	1	10%	370
76.	0.4	0.9	1	10	2	20%	373
77.	0.5	0.9	1	10	1	10%	375
78.	0.6	0.9	1	10	1	10%	377
79.	0.7	0.9	1	10	1	10%	378
80.	0.8	0.9	1	10	1	10%	380
81.	0.9	0.9	1	10	2	20%	381

Berdasarkan percobaan di atas diperoleh grafik sebagai berikut :



Gambar 4.23 Grafik percobaan dengan jari-jari 1 topologi linier

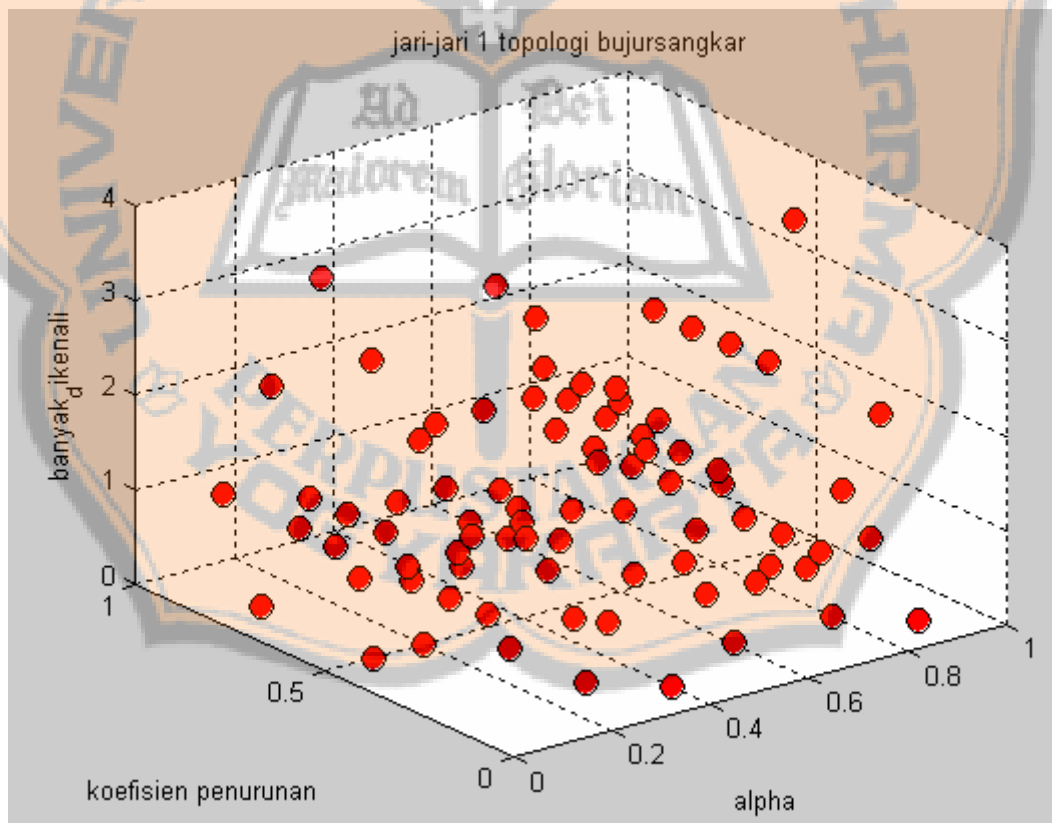
Tabel 3. Hasil percobaan dengan jari-jari 1 pada topologi bujursangkar

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	1	10	2	20%	18
2.	0.2	0.1	1	10	1	10%	18
3.	0.3	0.1	1	10	2	20%	18
4.	0.4	0.1	1	10	0	0%	18
5.	0.5	0.1	1	10	2	20%	19
6.	0.6	0.1	1	10	1	10%	19
7.	0.7	0.1	1	10	1	10%	20
8.	0.8	0.1	1	10	1	10%	20
9.	0.9	0.1	1	10	0	0%	19
10.	0.1	0.2	1	10	1	10%	24
11.	0.2	0.2	1	10	4	40%	25
12.	0.3	0.2	1	10	0	0%	26
13.	0.4	0.2	1	10	1	10%	26
14.	0.5	0.2	1	10	1	10%	25
15.	0.6	0.2	1	10	0	0%	26
16.	0.7	0.2	1	10	1	10%	27
17.	0.8	0.2	1	10	0	0%	26
18.	0.9	0.2	1	10	2	20%	26
19.	0.1	0.3	1	10	1	10%	33
20.	0.2	0.3	1	10	2	20%	34
21.	0.3	0.3	1	10	1	10%	34
22.	0.4	0.3	1	10	2	20%	34
23.	0.5	0.3	1	10	2	20%	34
24.	0.6	0.3	1	10	1	10%	35
25.	0.7	0.3	1	10	1	10%	35
26.	0.8	0.3	1	10	4	40%	34
27.	0.9	0.3	1	10	1	10%	35
28.	0.1	0.4	1	10	1	10%	43
29.	0.2	0.4	1	10	1	10%	44
30.	0.3	0.4	1	10	0	0%	43
31.	0.4	0.4	1	10	1	10%	44
32.	0.5	0.4	1	10	0	0%	44
33.	0.6	0.4	1	10	2	20%	44
34.	0.7	0.4	1	10	0	0%	47
35.	0.8	0.4	1	10	0	0%	46
36.	0.9	0.4	1	10	0	0%	47

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	1	10	0	0%	56
38.	0.2	0.5	1	10	0	0%	56
39.	0.3	0.5	1	10	1	10%	63
40.	0.4	0.5	1	10	1	10%	57
41.	0.5	0.5	1	10	1	10%	60
42.	0.6	0.5	1	10	2	20%	59
43.	0.7	0.5	1	10	1	10%	58
44.	0.8	0.5	1	10	1	10%	58
45.	0.9	0.5	1	10	2	20%	58
46.	0.1	0.6	1	10	1	10%	76
47.	0.2	0.6	1	10	1	10%	76
48.	0.3	0.6	1	10	2	20%	77
49.	0.4	0.6	1	10	2	20%	77
50.	0.5	0.6	1	10	2	20%	79
51.	0.6	0.6	1	10	2	20%	78
52.	0.7	0.6	1	10	1	10%	78
53.	0.8	0.6	1	10	1	10%	79
54.	0.9	0.6	1	10	2	20%	79
55.	0.1	0.7	1	10	1	10%	109
56.	0.2	0.7	1	10	1	10%	113
57.	0.3	0.7	1	10	1	10%	110
58.	0.4	0.7	1	10	1	10%	111
59.	0.5	0.7	1	10	3	30%	111
60.	0.6	0.7	1	10	2	20%	112
61.	0.7	0.7	1	10	1	10%	112
62.	0.8	0.7	1	10	1	10%	116
63.	0.9	0.7	1	10	2	20%	113
64.	0.1	0.8	1	10	0	0%	170
65.	0.2	0.8	1	10	1	10%	173
66.	0.3	0.8	1	10	0	0%	175
67.	0.4	0.8	1	10	0	0%	176
68.	0.5	0.8	1	10	0	0%	177
69.	0.6	0.8	1	10	0	0%	178
70.	0.7	0.8	1	10	1	10%	179
71.	0.8	0.8	1	10	1	10%	179
72.	0.9	0.8	1	10	2	20%	180

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	1	10	1	10%	367
74.	0.2	0.9	1	10	2	20%	366
75.	0.3	0.9	1	10	3	30%	370
76.	0.4	0.9	1	10	2	20%	373
77.	0.5	0.9	1	10	1	10%	375
78.	0.6	0.9	1	10	0	0%	377
79.	0.7	0.9	1	10	0	0%	378
80.	0.8	0.9	1	10	1	10%	379
81.	0.9	0.9	1	10	1	10%	381

Berdasarkan percobaan di atas diperoleh grafik sebagai berikut :



Gambar 4.24 Grafik percobaan dengan jari-jari 1 topologi bujursangkar

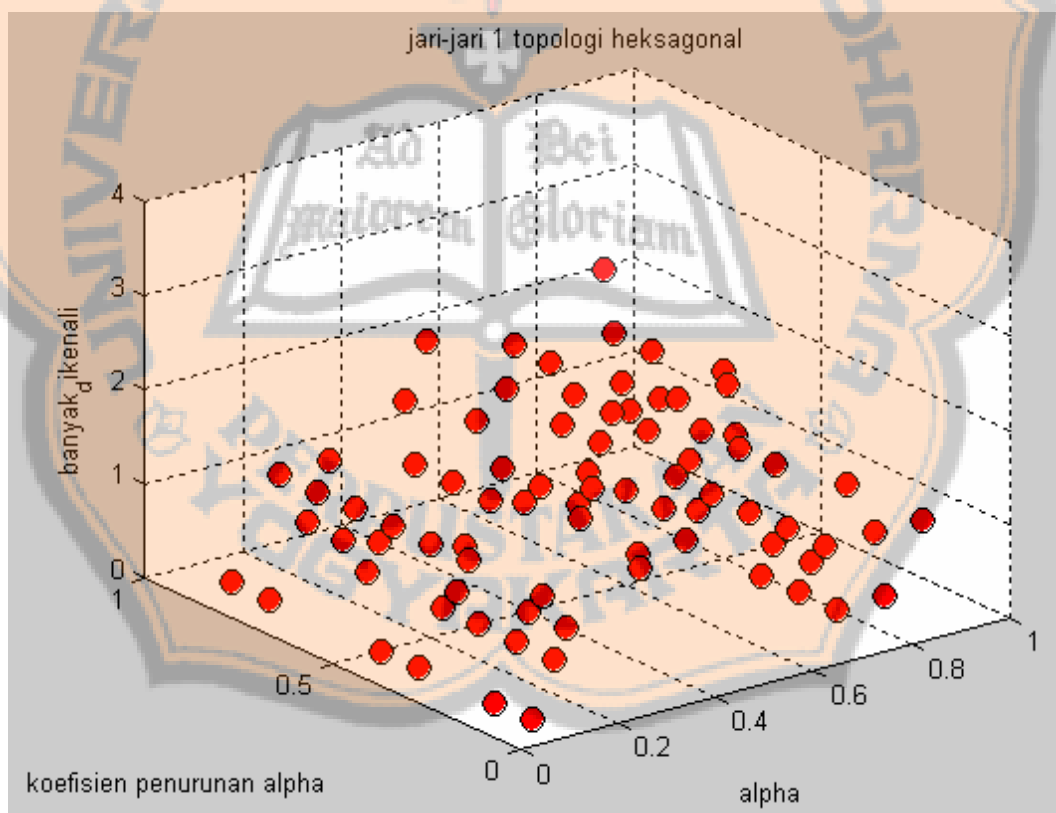
Tabel 4. Hasil percobaan dengan jari-jari 1 pada topologi heksagonal

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	1	10	0	0%	18
2.	0.2	0.1	1	10	2	20%	18
3	0.3	0.1	1	10	3	30%	18
4.	0.4	0.1	1	10	3	30%	19
5.	0.5	0.1	1	10	3	30%	19
6.	0.6	0.1	1	10	2	20%	19
7.	0.7	0.1	1	10	1	10%	19
8.	0.8	0.1	1	10	1	10%	18
9.	0.9	0.1	1	10	1	10%	19
10.	0.1	0.2	1	10	0	0%	25
11.	0.2	0.2	1	10	1	10%	25
12.	0.3	0.2	1	10	2	20%	26
13.	0.4	0.2	1	10	1	10%	26
14.	0.5	0.2	1	10	2	20%	26
15.	0.6	0.2	1	10	2	20%	26
16.	0.7	0.2	1	10	1	10%	27
17.	0.8	0.2	1	10	0	0%	27
18.	0.9	0.2	1	10	0	0%	26
19.	0.1	0.3	1	10	1	10%	33
20.	0.2	0.3	1	10	3	30%	33
21	0.3	0.3	1	10	0	0%	35
22.	0.4	0.3	1	10	4	40%	34
23.	0.5	0.3	1	10	3	30%	34
24.	0.6	0.3	1	10	2	20%	34
25.	0.7	0.3	1	10	1	10%	35
26.	0.8	0.3	1	10	0	0%	35
27.	0.9	0.3	1	10	1	10%	34
28.	0.1	0.4	1	10	0	0%	45
29.	0.2	0.4	1	10	1	10%	44
30.	0.3	0.4	1	10	0	0%	44
31.	0.4	0.4	1	10	0	0%	46
32.	0.5	0.4	1	10	3	30%	45
33.	0.6	0.4	1	10	1	10%	44
34.	0.7	0.4	1	10	1	10%	45
35.	0.8	0.4	1	10	0	0%	45
36.	0.9	0.4	1	10	0	0%	45

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	1	10	0	0%	56
38.	0.2	0.5	1	10	1	10%	57
39.	0.3	0.5	1	10	0	0%	59
40.	0.4	0.5	1	10	0	0%	57
41.	0.5	0.5	1	10	1	10%	57
42.	0.6	0.5	1	10	1	10%	59
43.	0.7	0.5	1	10	1	10%	59
44.	0.8	0.5	1	10	2	20%	62
45.	0.9	0.5	1	10	0	0%	62
46.	0.1	0.6	1	10	1	10%	76
47.	0.2	0.6	1	10	1	10%	79
48.	0.3	0.6	1	10	0	0%	77
49.	0.4	0.6	1	10	1	10%	77
50.	0.5	0.6	1	10	1	10%	78
51.	0.6	0.6	1	10	1	10%	80
52.	0.7	0.6	1	10	0	0%	80
53.	0.8	0.6	1	10	0	0%	80
54.	0.9	0.6	1	10	1	10%	80
55.	0.1	0.7	1	10	1	10%	111
56.	0.2	0.7	1	10	1	10%	111
57.	0.3	0.7	1	10	2	20%	110
58.	0.4	0.7	1	10	1	10%	111
59.	0.5	0.7	1	10	1	10%	111
60.	0.6	0.7	1	10	2	20%	118
61.	0.7	0.7	1	10	1	10%	112
62.	0.8	0.7	1	10	1	10%	116
63.	0.9	0.7	1	10	0	0%	115
64.	0.1	0.8	1	10	0	0%	170
65.	0.2	0.8	1	10	1	10%	173
66.	0.3	0.8	1	10	0	0%	175
67.	0.4	0.8	1	10	1	10%	176
68.	0.5	0.8	1	10	0	0%	177
69.	0.6	0.8	1	10	2	20%	178
70.	0.7	0.8	1	10	1	10%	179
71.	0.8	0.8	1	10	1	10%	180
72.	0.9	0.8	1	10	1	10%	183

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	1	10	0	0%	360
74.	0.2	0.9	1	10	1	10%	366
75.	0.3	0.9	1	10	1	10%	377
76.	0.4	0.9	1	10	0	0%	373
77.	0.5	0.9	1	10	2	20%	375
78.	0.6	0.9	1	10	1	10%	377
79.	0.7	0.9	1	10	0	0%	378
80.	0.8	0.9	1	10	1	10%	380
81.	0.9	0.9	1	10	1	10%	381

Berdasarkan percobaan di atas diperoleh grafik sebagai berikut :



Gambar 4.25 Grafik percobaan dengan jari-jari 1 topologi heksagonal

4.4. Pembahasan

- Percobaan dengan Pola Uji Kelompok 1

0	1	2	3	4
5	6	7	8	9

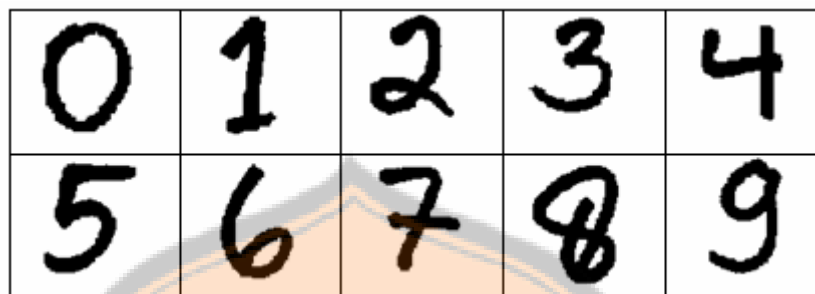
Gambar 4.22 Pola Uji Kelompok 1

Berdasarkan hasil percobaan kelompok angka pertama dapat diperoleh kesimpulan bahwa persentase dikenali terbanyak terdapat pada percobaan dengan jari-jari 0 untuk semua topologi. Persentase dikenali maksimal dari jari-jari 0 tersebut adalah 100% melebihi hasil percobaan dengan jari-jari 1 topologi linear, bujursangkar, dan heksagonal yang memiliki persentase dikenali maksimal 40%. Kombinasi alpha dan koefisien penurunan alpha pada jari-jari 0 semua topologi yang menghasilkan persentasi dikenali 100% tersebut adalah sebagai berikut :

Tabel 5. Kombinasi input yang menghasilkan persentase dikenali terbaik untuk kelompok angka pertama

Alpha	Koefisien penurunan alpha	Jari-jari	Jumlah pola uji	Banyak dikenali	Persentase dikenali
0.8	0.5	0	10	10	100%
0.9	0.5	0	10	10	100%
0.8	0.6	0	10	10	100%
0.9	0.6	0	10	10	100%
0.8	0.7	0	10	10	100%
0.9	0.7	0	10	10	100%
0.8	0.8	0	10	10	100%
0.9	0.8	0	10	10	100%
0.7	0.9	0	10	10	100%
0.8	0.9	0	10	10	100%
0.9	0.9	0	10	10	100%

- Percobaan dengan Pola Uji Kelompok 2



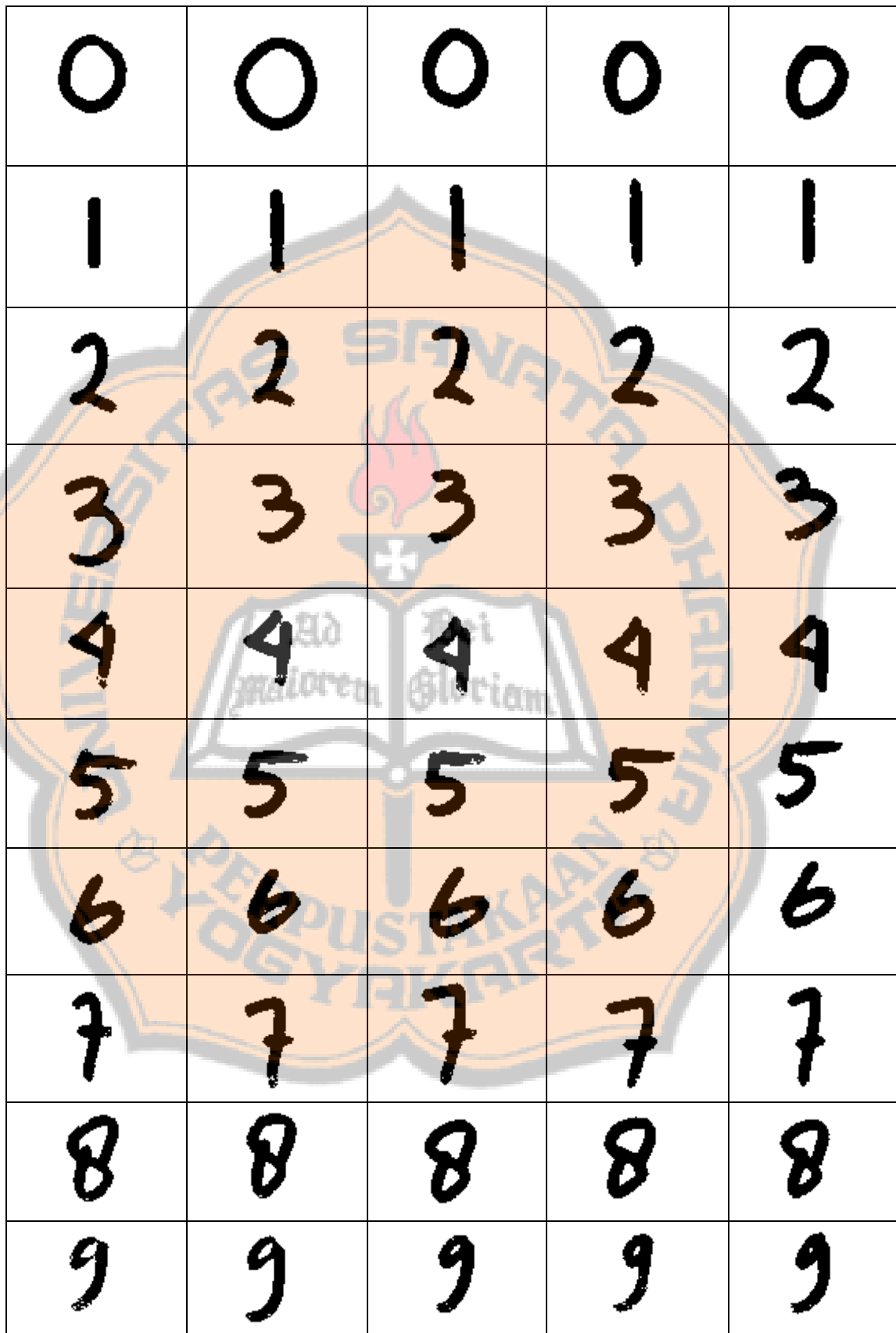
Gambar 4.23 Pola Uji Kelompok 2

Pada percobaan dengan menggunakan kelompok angka kedua juga diperoleh hasil bahwa persentase dikenali terbaik pada saat input jari-jarinya 0. Hanya saja persentase dikenali maksimal yang diperoleh tidak sebanyak pada percobaan dengan kelompok pola pertama.

Tabel 6. Kombinasi input yang menghasilkan persentase dikenali terbaik untuk kelompok angka kedua

Alpha	Koefisien penurunan alpha	Jari-jari	Jumlah pola uji	Banyak dikenali	Persentase dikenali
0.9	0.1	0	10	5	50%
0.9	0.3	0	10	5	50%
0.8	0.4	0	10	5	50%
0.7	0.5	0	10	5	50%
0.8	0.5	0	10	5	50%
0.9	0.6	0	10	5	50%
0.8	0.7	0	10	5	50%
0.9	0.7	0	10	5	50%
0.8	0.8	0	10	5	50%
0.9	0.8	0	10	5	50%
0.7	0.9	0	10	5	50%
0.8	0.9	0	10	5	50%
0.9	0.9	0	10	5	50%

Hasil tersebut dipengaruhi oleh struktur pixel Pola Uji yang memiliki banyak perbedaan dengan struktur pixel pada Pola *Template* berikut.



0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9

Gambar 4.24 Pola Template



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Beberapa kesimpulan yang dapat diambil dari penulisan tugas akhir ini adalah:

1. Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen dapat digunakan untuk pengenalan pola angka 0 sampai 9.
2. Berdasarkan hasil penelitian dan percobaan yang telah dilakukan dapat diperoleh kesimpulan bahwa pengenalan pola angka 0 sampai 9 menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen memiliki tingkat keberhasilan hingga 100% dengan input :
 - Pola Template pada percobaan ini.
 - Pola Uji kelompok 1 pada percobaan ini.
 - Jari-jari 0 semua topologi.
 - Kombinasi Alpha dan Koefisien Penurunan Alpha :

Alpha	Koefisien Penurunan Alpha
0.8	0.5
0.9	0.5
0.8	0.6
0.9	0.6
0.8	0.7
0.9	0.7
0.8	0.8
0.9	0.8
0.7	0.9
0.8	0.9
0.9	0.9

3. Modifikasi bobot dipengaruhi oleh topologi dan jari-jari.
4. Percobaan dengan pola Uji kelompok 1, jari-jari 1 topologi linear, bujursangkar, dan heksagonal memiliki persentase dikenali maksimal 40 %.

5.2. Saran

Pada pengembangan selanjutnya proses pelatihan pada sistem pengenalan pola angka 0 sampai 9 menggunakan Jaringan Syaraf Tiruan dengan Model Jaringan Kohonen tersebut dapat lebih disempurnakan lagi. Pengembangan yang bisa dilakukan :

1. Memilih pola template yang lebih universal sehingga dapat digunakan untuk aplikasi pengenalan pola angka yang lebih luas penggunaannya. Contoh penggunaan pengenalan pola angka yang lebih luas adalah pada pengenalan kodepos.
2. Menampilkan visualisasi perubahan bobot yang terjadi agar user dapat mengetahui lebih jelas tentang proses yang sedang berlangsung.

DAFTAR PUSTAKA

E.Dayhoff, Judith. (1990). *Neural Network Architectures An Introduction*. New York : Van Nostrand Reinhold.

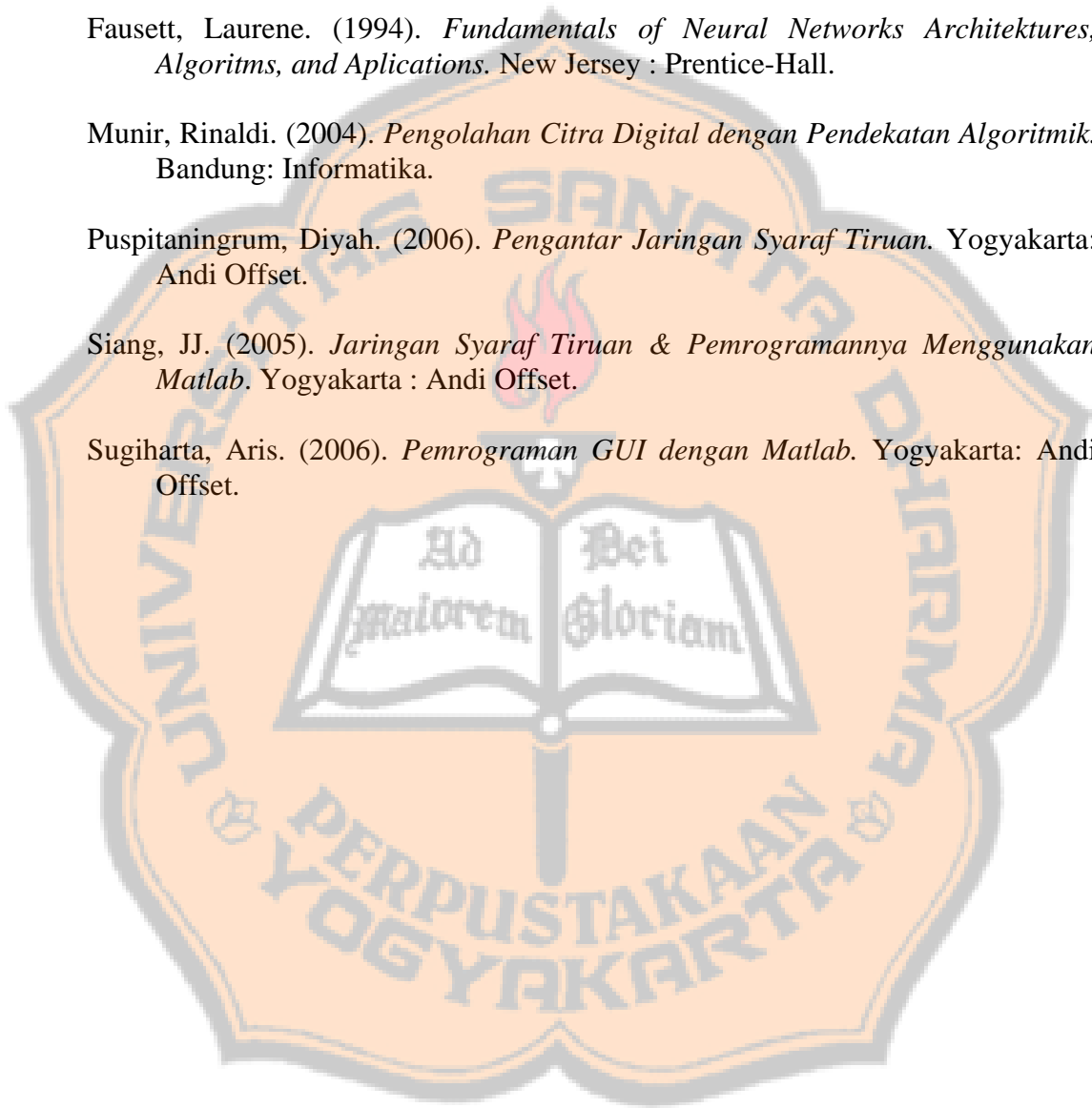
Fausett, Laurene. (1994). *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*. New Jersey : Prentice-Hall.

Munir, Rinaldi. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.

Puspitaningrum, Diyah. (2006). *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Andi Offset.

Siang, JJ. (2005). *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*. Yogyakarta : Andi Offset.

Sugiharta, Aris. (2006). *Pemrograman GUI dengan Matlab*. Yogyakarta: Andi Offset.



LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

PERCOBAAN PENGENALAN ANGKA 0 SAMPAI 9
Jari-jari 0 semua topologi

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	0	10	0	0%	17
2.	0.2	0.1	0	10	0	0%	17
3.	0.3	0.1	0	10	1	10%	18
4.	0.4	0.1	0	10	3	30%	18
5.	0.5	0.1	0	10	2	20%	19
6.	0.6	0.1	0	10	3	30%	18
7.	0.7	0.1	0	10	2	20%	18
8.	0.8	0.1	0	10	3	30%	18
9.	0.9	0.1	0	10	5	50%	18
10.	0.1	0.2	0	10	0	0%	24
11.	0.2	0.2	0	10	0	0%	24
12.	0.3	0.2	0	10	1	10%	25
13.	0.4	0.2	0	10	2	20%	25
14.	0.5	0.2	0	10	3	30%	25
15.	0.6	0.2	0	10	2	20%	25
16.	0.7	0.2	0	10	2	20%	25
17.	0.8	0.2	0	10	3	30%	25
18.	0.9	0.2	0	10	4	40%	25
19.	0.1	0.3	0	10	0	0%	31
20.	0.2	0.3	0	10	0	0%	33
21.	0.3	0.3	0	10	0	0%	33
22.	0.4	0.3	0	10	2	20%	33
23.	0.5	0.3	0	10	2	20%	33
24.	0.6	0.3	0	10	4	40%	33
25.	0.7	0.3	0	10	5	50%	33
26.	0.8	0.3	0	10	4	40%	33
27.	0.9	0.3	0	10	5	50%	33
28.	0.1	0.4	0	10	0	0%	42
29.	0.2	0.4	0	10	0	0%	43
30.	0.3	0.4	0	10	0	0%	43
31.	0.4	0.4	0	10	1	10%	43
32.	0.5	0.4	0	10	3	30%	43
33.	0.6	0.4	0	10	2	20%	43
34.	0.7	0.4	0	10	4	40%	43
35.	0.8	0.4	0	10	5	50%	42
36.	0.9	0.4	0	10	3	30%	44

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	0	10	0	0%	54
38.	0.2	0.5	0	10	0	0%	56
39.	0.3	0.5	0	10	0	0%	57
40.	0.4	0.5	0	10	4	40%	56
41.	0.5	0.5	0	10	2	20%	56
42.	0.6	0.5	0	10	4	40%	57
43.	0.7	0.5	0	10	5	50%	56
44.	0.8	0.5	0	10	5	50%	56
45.	0.9	0.5	0	10	4	40%	57
46.	0.1	0.6	0	10	0	0%	73
47.	0.2	0.6	0	10	0	0%	76
48.	0.3	0.6	0	10	0	0%	77
49.	0.4	0.6	0	10	3	30%	76
50.	0.5	0.6	0	10	4	40%	76
51.	0.6	0.6	0	10	4	40%	77
52.	0.7	0.6	0	10	3	30%	77
53.	0.8	0.6	0	10	4	40%	77
54.	0.9	0.6	0	10	5	50%	76
55.	0.1	0.7	0	10	0	0%	107
56.	0.2	0.7	0	10	0	0%	109
57.	0.3	0.7	0	10	0	0%	110
58.	0.4	0.7	0	10	2	20%	108
59.	0.5	0.7	0	10	3	30%	109
60.	0.6	0.7	0	10	4	40%	110
61.	0.7	0.7	0	10	3	30%	110
62.	0.8	0.7	0	10	5	50%	108
63.	0.9	0.7	0	10	5	50%	109
64.	0.1	0.8	0	10	0	0%	170
65.	0.2	0.8	0	10	1	10%	173
66.	0.3	0.8	0	10	0	0%	172
67.	0.4	0.8	0	10	2	20%	173
68.	0.5	0.8	0	10	4	40%	174
69.	0.6	0.8	0	10	3	30%	175
70.	0.7	0.8	0	10	4	40%	175
71.	0.8	0.8	0	10	5	50%	173
72.	0.9	0.8	0	10	5	50%	173

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	0	10	0	0%	353
74.	0.2	0.9	0	10	0	0%	366
75.	0.3	0.9	0	10	0	0%	363
76.	0.4	0.9	0	10	2	20%	366
77.	0.5	0.9	0	10	3	30%	368
78.	0.6	0.9	0	10	4	40%	370
79.	0.7	0.9	0	10	5	50%	364
80.	0.8	0.9	0	10	5	50%	365
81.	0.9	0.9	0	10	5	50%	366



LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

PERCOBAAN PENGENALAN ANGKA 0 SAMPAI 9
Jari-jari 1 topologi linear

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	1	10	2	20%	18
2.	0.2	0.1	1	10	1	10%	18
3.	0.3	0.1	1	10	1	10%	19
4.	0.4	0.1	1	10	1	10%	18
5.	0.5	0.1	1	10	1	10%	19
6.	0.6	0.1	1	10	0	0%	19
7.	0.7	0.1	1	10	0	0%	19
8.	0.8	0.1	1	10	0	0%	20
9.	0.9	0.1	1	10	0	0%	19
10.	0.1	0.2	1	10	1	10%	25
11.	0.2	0.2	1	10	0	0%	25
12.	0.3	0.2	1	10	1	10%	25
13.	0.4	0.2	1	10	1	10%	26
14.	0.5	0.2	1	10	0	0%	25
15.	0.6	0.2	1	10	0	0%	26
16.	0.7	0.2	1	10	1	10%	27
17.	0.8	0.2	1	10	1	10%	27
18.	0.9	0.2	1	10	1	10%	26
19.	0.1	0.3	1	10	0	0%	33
20.	0.2	0.3	1	10	2	20%	35
21.	0.3	0.3	1	10	0	0%	33
22.	0.4	0.3	1	10	1	10%	35
23.	0.5	0.3	1	10	1	10%	34
24.	0.6	0.3	1	10	1	10%	34
25.	0.7	0.3	1	10	1	10%	34
26.	0.8	0.3	1	10	1	10%	37
27.	0.9	0.3	1	10	2	20%	34
28.	0.1	0.4	1	10	1	10%	44
29.	0.2	0.4	1	10	0	0%	44
30.	0.3	0.4	1	10	0	0%	45
31.	0.4	0.4	1	10	1	10%	44
32.	0.5	0.4	1	10	2	20%	46
33.	0.6	0.4	1	10	0	0%	49
34.	0.7	0.4	1	10	0	0%	44
35.	0.8	0.4	1	10	0	0%	45
36.	0.9	0.4	1	10	0	0%	46

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	1	10	1	10%	57
38.	0.2	0.5	1	10	0	0%	58
39.	0.3	0.5	1	10	1	10%	59
40.	0.4	0.5	1	10	0	0%	57
41.	0.5	0.5	1	10	1	10%	61
42.	0.6	0.5	1	10	0	0%	58
43.	0.7	0.5	1	10	0	0%	59
44.	0.8	0.5	1	10	1	10%	61
45.	0.9	0.5	1	10	2	20%	61
46.	0.1	0.6	1	10	1	10%	80
47.	0.2	0.6	1	10	0	0%	76
48.	0.3	0.6	1	10	1	10%	78
49.	0.4	0.6	1	10	1	10%	77
50.	0.5	0.6	1	10	1	10%	78
51.	0.6	0.6	1	10	1	10%	78
52.	0.7	0.6	1	10	1	10%	80
53.	0.8	0.6	1	10	0	0%	80
54.	0.9	0.6	1	10	0	0%	79
55.	0.1	0.7	1	10	0	0%	109
56.	0.2	0.7	1	10	0	0%	113
57.	0.3	0.7	1	10	0	0%	110
58.	0.4	0.7	1	10	2	20%	111
59.	0.5	0.7	1	10	0	0%	111
60.	0.6	0.7	1	10	1	10%	112
61.	0.7	0.7	1	10	0	0%	112
62.	0.8	0.7	1	10	1	10%	116
63.	0.9	0.7	1	10	1	10%	113
64.	0.1	0.8	1	10	0	0%	170
65.	0.2	0.8	1	10	1	10%	173
66.	0.3	0.8	1	10	1	10%	175
67.	0.4	0.8	1	10	1	10%	176
68.	0.5	0.8	1	10	2	20%	187
69.	0.6	0.8	1	10	0	0%	191
70.	0.7	0.8	1	10	0	0%	179
71.	0.8	0.8	1	10	1	10%	179
72.	0.9	0.8	1	10	1	10%	180

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	1	10	0	0%	380
74.	0.2	0.9	1	10	1	10%	366
75.	0.3	0.9	1	10	1	10%	370
76.	0.4	0.9	1	10	0	0%	373
77.	0.5	0.9	1	10	1	10%	375
78.	0.6	0.9	1	10	0	0%	377
79.	0.7	0.9	1	10	0	0%	378
80.	0.8	0.9	1	10	0	0%	379
81.	0.9	0.9	1	10	1	10%	381



LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

PERCOBAAN PENGENALAN ANGKA 0 SAMPAI 9
Jari-jari 1 topologi bujursangkar

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	1	10	2	20%	18
2.	0.2	0.1	1	10	1	10%	18
3.	0.3	0.1	1	10	1	10%	19
4.	0.4	0.1	1	10	1	10%	18
5.	0.5	0.1	1	10	1	10%	19
6.	0.6	0.1	1	10	0	0%	19
7.	0.7	0.1	1	10	0	0%	20
8.	0.8	0.1	1	10	1	10%	20
9.	0.9	0.1	1	10	0	0%	19
10.	0.1	0.2	1	10	0	0%	24
11.	0.2	0.2	1	10	2	20%	25
12.	0.3	0.2	1	10	0	0%	26
13.	0.4	0.2	1	10	0	0%	26
14.	0.5	0.2	1	10	0	0%	25
15.	0.6	0.2	1	10	1	10%	26
16.	0.7	0.2	1	10	0	0%	27
17.	0.8	0.2	1	10	0	0%	26
18.	0.9	0.2	1	10	1	10%	26
19.	0.1	0.3	1	10	1	10%	33
20.	0.2	0.3	1	10	0	0%	35
21.	0.3	0.3	1	10	0	0%	33
22.	0.4	0.3	1	10	1	10%	35
23.	0.5	0.3	1	10	0	0%	35
24.	0.6	0.3	1	10	0	0%	35
25.	0.7	0.3	1	10	0	0%	34
26.	0.8	0.3	1	10	1	10%	34
27.	0.9	0.3	1	10	0	0%	36
28.	0.1	0.4	1	10	0	0%	45
29.	0.2	0.4	1	10	1	10%	43
30.	0.3	0.4	1	10	1	10%	44
31.	0.4	0.4	1	10	1	10%	46
32.	0.5	0.4	1	10	1	10%	44
33.	0.6	0.4	1	10	0	0%	45
34.	0.7	0.4	1	10	1	10%	45
35.	0.8	0.4	1	10	0	0%	44
36.	0.9	0.4	1	10	0	0%	46

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	1	10	1	10%	56
38.	0.2	0.5	1	10	1	10%	57
39.	0.3	0.5	1	10	0	0%	58
40.	0.4	0.5	1	10	0	0%	57
41.	0.5	0.5	1	10	1	10%	61
42.	0.6	0.5	1	10	1	10%	58
43.	0.7	0.5	1	10	1	10%	58
44.	0.8	0.5	1	10	0	0%	59
45.	0.9	0.5	1	10	1	10%	58
46.	0.1	0.6	1	10	0	0%	76
47.	0.2	0.6	1	10	0	0%	76
48.	0.3	0.6	1	10	0	0%	77
49.	0.4	0.6	1	10	1	10%	79
50.	0.5	0.6	1	10	0	0%	79
51.	0.6	0.6	1	10	1	10%	80
52.	0.7	0.6	1	10	0	0%	78
53.	0.8	0.6	1	10	0	0%	84
54.	0.9	0.6	1	10	1	10%	79
55.	0.1	0.7	1	10	1	10%	111
56.	0.2	0.7	1	10	1	10%	109
57.	0.3	0.7	1	10	0	0%	112
58.	0.4	0.7	1	10	0	0%	111
59.	0.5	0.7	1	10	1	10%	111
60.	0.6	0.7	1	10	0	0%	112
61.	0.7	0.7	1	10	0	0%	112
62.	0.8	0.7	1	10	0	0%	112
63.	0.9	0.7	1	10	0	0%	115
64.	0.1	0.8	1	10	1	10%	170
65.	0.2	0.8	1	10	1	10%	173
66.	0.3	0.8	1	10	1	10%	175
67.	0.4	0.8	1	10	0	0%	176
68.	0.5	0.8	1	10	0	0%	177
69.	0.6	0.8	1	10	0	0%	178
70.	0.7	0.8	1	10	0	0%	179
71.	0.8	0.8	1	10	1	10%	192
72.	0.9	0.8	1	10	0	0%	183

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	1	10	0	0%	360
74.	0.2	0.9	1	10	0	0%	366
75.	0.3	0.9	1	10	0	0%	370
76.	0.4	0.9	1	10	0	0%	373
77.	0.5	0.9	1	10	0	0%	375
78.	0.6	0.9	1	10	0	0%	377
79.	0.7	0.9	1	10	1	10%	378
80.	0.8	0.9	1	10	1	10%	379
81.	0.9	0.9	1	10	0	0%	381



LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

PERCOBAAN PENGENALAN ANGKA 0 SAMPAI 9
Jari-jari 1 topologi heksagonal

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
1.	0.1	0.1	1	10	0	0%	18
2.	0.2	0.1	1	10	1	10%	18
3.	0.3	0.1	1	10	0	0%	18
4.	0.4	0.1	1	10	0	0%	18
5.	0.5	0.1	1	10	0	0%	19
6.	0.6	0.1	1	10	0	0%	19
7.	0.7	0.1	1	10	0	0%	20
8.	0.8	0.1	1	10	2	20%	18
9.	0.9	0.1	1	10	0	0%	19
10.	0.1	0.2	1	10	0	0%	24
11.	0.2	0.2	1	10	0	0%	25
12.	0.3	0.2	1	10	0	0%	26
13.	0.4	0.2	1	10	0	0%	25
14.	0.5	0.2	1	10	1	10%	28
15.	0.6	0.2	1	10	1	10%	26
16.	0.7	0.2	1	10	0	0%	26
17.	0.8	0.2	1	10	1	10%	26
18.	0.9	0.2	1	10	0	0%	26
19.	0.1	0.3	1	10	0	0%	35
20.	0.2	0.3	1	10	0	0%	33
21.	0.3	0.3	1	10	1	10%	34
22.	0.4	0.3	1	10	0	0%	34
23.	0.5	0.3	1	10	1	10%	34
24.	0.6	0.3	1	10	1	10%	34
25.	0.7	0.3	1	10	1	10%	35
26.	0.8	0.3	1	10	1	10%	34
27.	0.9	0.3	1	10	1	10%	36
28.	0.1	0.4	1	10	0	0%	43
29.	0.2	0.4	1	10	0	0%	45
30.	0.3	0.4	1	10	0	0%	43
31.	0.4	0.4	1	10	1	10%	46
32.	0.5	0.4	1	10	0	0%	44
33.	0.6	0.4	1	10	0	0%	48
34.	0.7	0.4	1	10	0	0%	45
35.	0.8	0.4	1	10	1	10%	46
36.	0.9	0.4	1	10	0	0%	44

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
37.	0.1	0.5	1	10	0	0%	57
38.	0.2	0.5	1	10	2	20%	57
39.	0.3	0.5	1	10	0	0%	57
40.	0.4	0.5	1	10	0	0%	57
41.	0.5	0.5	1	10	1	10%	58
42.	0.6	0.5	1	10	0	0%	60
43.	0.7	0.5	1	10	0	0%	58
44.	0.8	0.5	1	10	0	0%	61
45.	0.9	0.5	1	10	0	0%	59
46.	0.1	0.6	1	10	0	0%	77
47.	0.2	0.6	1	10	0	0%	77
48.	0.3	0.6	1	10	0	0%	78
49.	0.4	0.6	1	10	0	0%	79
50.	0.5	0.6	1	10	2	20%	79
51.	0.6	0.6	1	10	0	0%	78
52.	0.7	0.6	1	10	0	0%	80
53.	0.8	0.6	1	10	0	0%	80
54.	0.9	0.6	1	10	1	10%	79
55.	0.1	0.7	1	10	0	0%	109
56.	0.2	0.7	1	10	1	10%	109
57.	0.3	0.7	1	10	0	0%	110
58.	0.4	0.7	1	10	0	0%	113
59.	0.5	0.7	1	10	1	0%	111
60.	0.6	0.7	1	10	0	0%	112
61.	0.7	0.7	1	10	0	0%	112
62.	0.8	0.7	1	10	1	10%	116
63.	0.9	0.7	1	10	1	10%	115
64.	0.1	0.8	1	10	0	0%	170
65.	0.2	0.8	1	10	1	10%	173
66.	0.3	0.8	1	10	0	0%	175
67.	0.4	0.8	1	10	1	10%	189
68.	0.5	0.8	1	10	0	0%	177
69.	0.6	0.8	1	10	0	0%	178
70.	0.7	0.8	1	10	0	0%	188
71.	0.8	0.8	1	10	0	0%	180
72.	0.9	0.8	1	10	0	0%	180

LAMPIRAN 1 Hasil Percobaan Dengan Pola Uji Kelompok 2

No	Alpha	Koef.turun	Jari-jari	Jml pola uji	Jml yg dikenali	Persentase dikenali	JML EPOCH
73.	0.1	0.9	1	10	0	0%	360
74.	0.2	0.9	1	10	0	0%	366
75.	0.3	0.9	1	10	0	0%	370
76.	0.4	0.9	1	10	0	0%	373
77.	0.5	0.9	1	10	0	0%	402
78.	0.6	0.9	1	10	1	10%	397
79.	0.7	0.9	1	10	1	10%	378
80.	0.8	0.9	1	10	0	0%	380
81.	0.9	0.9	1	10	0	0%	381

