

PENDETEKSIAN KANKER PAYUDARA DENGAN  
*MAMMOGRAM* MENGGUNAKAN  
JARINGAN SYARAF TIRUAN *BACKPROPAGATION*

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Komputer dalam bidang Ilmu Komputer

oleh:  
**LITA INDAH AYU**  
**0410963031-96**



PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2009

UNIVERSITAS BRAWIJAYA



ii

LEMBAR PENGESAHAN SKRIPSI

PENDETEKSIAN KANKER PAYUDARA DENGAN  
*MAMMOGRAM MENGGUNAKAN*  
*JARINGAN SYARAF TIRUAN BACKPROPAGATION*

Oleh:  
**LITA INDAH AYU**  
**0410963031-96**

Setelah dipertahankan di depan Majelis Pengaji  
Pada tanggal 14 Juli 2009  
dan dinyatakan memenuhi syarat untuk memperoleh gelar  
Sarjana dalam bidang Ilmu Komputer

Pembimbing I

**Drs. Muh. Arif Rahman, M.Kom**  
NIP. 131 971 481

Pembimbing II

**Drs. Marji, MT**  
NIP. 131 993 386

Mengetahui,  
Ketua Jurusan Matematika  
Fakultas MIPA Universitas Brawijaya

**Dr. Agus Suryanto, M.Sc**  
NIP. 132 126 049



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama :  
NIM :  
Jurusan :  
Penulis Skripsi berjudul

: Lita Indah Ayu  
: 0410963031  
: Matematika  
: Pendeksiian Kanker Payudara  
Dengan *Mammogram*  
Menggunakan Jaringan Syaraf  
Tiruan *Backpropagation*.

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 14 Juli 2009  
Yang menyatakan,

(Lita Indah Ayu)  
NIM. 0410963031



**PENDETEKSIAN KANKER PAYUDARA DENGAN  
MAMMOGRAM MENGGUNAKAN  
JARINGAN SYARAF TIRUAN *BACKPROPAGATION***

**ABSTRAK**

Kanker payudara menduduki urutan kedua terbanyak penyebab kematian pada wanita saat ini. Pencegahan kanker payudara dapat dilakukan secara dini dengan pemeriksaan *mammografi*. *Mammografi* merupakan pemeriksaan menggunakan sinar *rotgen* dosis rendah untuk melihat bagian dalam payudara. Hasil dari *mammografi* disebut *mammogram*, yang biasanya digunakan untuk membantu proses awal pendeksi dan *diagnosa* kanker payudara pada wanita.

Pada penelitian ini disajikan suatu metode untuk mendeksi kanker payudara dengan *mammogram* menggunakan algoritma Jaringan Syaraf Tiruan(JST) *backpropagation* dengan fungsi aktivasi biner sigmoid. Secara umum sistem ini memiliki 2 bagian pemrosesan. Proses pertama pengolahan citra *mammogram* yang menerima inputan sistem. Dalam sistem ini menggunakan 3 karakteristik data input, yaitu menggunakan inputan dari nilai *mean* dan *variance* dan perbandingan *mean* dan *variance*. Proses kedua adalah proses pelatihan dan pengenalan *mammogram* dengan Jaringan Syaraf Tiruan.

Untuk mendapatkan struktur JST yang terbaik, dilakukan pelatihan dengan beberapa parameter diantaranya, ukuran citra inputan, jumlah neuron pada *hidden layer* dan laju pembelajaran (*learning rate*). Tolak ukur keberhasilan sistem pendeksi kanker payudara ini adalah dengan mengitung jarak error terdekat dan tingkat keakuratan dalam pendeksi kanker payudara. Dari penelitian ini diperoleh struktur JST dengan jumlah neuron pada *input layer* sebanyak 600 unit, jumlah neuron pada *hidden layer* sebanyak 30 unit, jumlah neuron pada *output layer* sebanyak 1 unit, nilai *learning rate* sebesar 2.0 dan *max epoch* sebesar 5000.

JST yang terbentuk mampu mengenali citra *mammogram* dengan nilai keakuratan rata-rata untuk inputan dengan nilai *mean* 94.87%, *variance* 82.05% dan perbandingan *mean variance* 89.74%. Dari ketiga karakteristik data input yang digunakan pada sistem ini, data input menggunakan nilai *mean* menghasilkan pengenalan yang baik.



**BREAST CANCER DETECTION WITH MAMMOGRAM  
USING ARTIFICIAL NEURAL NETWORK  
BACKPROPAGATION ALGORITHM**

**ABSTRACT**

Currently, breast cancer gets a second position as the causal of mortality in women. Breast cancer prevention can be done as early as possible by conducting mammography examination. Mammography is an examination using roentgen rays with low dosage to observe inner part of breast. The result of mammography is called mammogram. It is usually used to help the early process of detection and diagnose of breast cancer on women.

In this research, it is presented a method to detect breast cancer that is mammogram using artificial neural network (ANN) backpropagation algorithm with sigmoid binary activation function. In general, this system has 2 processing units. The first unit is mammogram image processing that achieves system inputs. It uses 3 input characteristics: input from mean score and variance and comparison between mean and variance. The second one is an training and recognition process on mammogram with artificial neural network.

Training using some parameter is done to get the best ANN structure. Those parameters are input image size, the amount of neuron in hidden layer, and learning rate. Success-measuring rod of this breast cancer detection system is done by counting the narrowest error distance and its accuracy level. From this research, 600 units neuron at the input layer, 300 unit neurons at the hidden layer, and 1 unit neuron at the output layer in ANN structure are obtained with its learning rate is 2.0 and max epoch is 1000.

The formed ANN can recognize mammogram image with the average accuracy values for mean value input is 94.87%, 82.05% in variance, and 89.74% in mean variance comparison. From those three input data characteristics used in this system, input using mean value produces a good recognition.



## KATA PENGANTAR

*Alhamdulillahi rabbil 'alamin.* Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas segala rahmat dan limpahan hidayahNya, skripsi yang berjudul "**Pendeteksian Kanker Payudara dengan Mammogram Menggunakan Jaringan Syaraf Tiruan Backpropagation**" ini dapat diselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW, makhluk paling mulia yang senantiasa memberikan cahya petunjuk, dan atas keluarganya dan sahabat-sahabatnya.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Drs. Muh. Arif Rahman, M.Kom dan Drs. Marji, MT terima kasih atas semua waktu,saran, bantuan dan bimbingan yang telah diberikan.
2. Wayan Firdaus Mahmudy, SSi, MT selaku Ketua Program Studi Ilmu Komputer Unibraw Malang.
3. Kasyful Amron,ST selaku Penasihat Akademik.
4. Dr. Agus Suryanto, Msc selaku Ketua Jurusan Matematika Fakultas MIPA Unibraw Malang
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya
7. Ayah, Ibu dan keluarga. Terima kasih atas cinta, kasih sayang, doa, dukungan dan semangat yang tiada henti.
8. Mas Riphin, yang senantiasa memberikan semangat dan doanya. Ekak dan Mas Ady, terima kasih telah memberikan bantuan selama di Surabaya.
9. Sahabat Loelie serta teman-teman ilkomers '04. Terima kasih atas senyuman, semangat, dukungan, do'a dan hari-hari kita.

10. Pihak lain yang telah membantu terselesaikannya Skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga penulisan laporan tugas akhir ini bermanfaat bagi pembaca sekalian. Dengan tidak lupa atas kodratnya sebagai manusia, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, 14 Juli 2009

Penulis

UNIVERSITAS BRAWIJAYA



xii

**DAFTAR ISI**

Halaman

<b>HALAMAN JUDUL .....</b>	i
<b>HALAMAN PENGESAHAN .....</b>	iii
<b>HALAMAN PERNYATAAN .....</b>	v
<b>ABSTRAK .....</b>	vii
<b>KATA PENGANTAR .....</b>	xi
<b>DAFTAR ISI .....</b>	xiii
<b>DAFTAR GAMBAR .....</b>	xv
<b>DAFTAR TABEL .....</b>	xvii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Pelatihan.....	3
1.7 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Kanker payudara .....	5
2.2 <i>Mammografi</i> .....	5
2.2.1 Tipe <i>Mammografi</i>	6
2.2.2 <i>Mammogram Report</i> .....	10
2.3 Pengolahan Citra Digital .....	11
2.3.1 Representasi Warna Keabuan ( <i>Gray scale</i> ).....	12
2.3.2 <i>Thresholding</i> .....	12
2.3.3 Karakteristik Citra .....	12
2.4 Jaringan Syaraf Tiruan .....	13
2.4.1 Definisi Jaringan Syaraf Tiruan.....	13
2.4.2 Arsitektur Jaringan Syaraf Tiruan .....	14
2.4.3 Model Matematis.....	16
2.5 Pelatihan JST .....	17
2.6 <i>Backpropagation</i> (propagasi Balik) .....	18
2.6.1 Arsitektur <i>Backpropagation</i> .....	19
2.6.2 Fungsi Aktivasi.....	19
2.6.3 Normalisasi Data .....	20
2.6.4. Algoritma <i>Backpropagation</i> .....	21

2.7	Optimalitas Arsitektur <i>Backpropagation</i> .....	24	
2.7.1	Pemilihan Bobot dan Bias .....	24	
2.7.2	Jumlah Unit Tersembunyi.....	24	
2.7.3	Lama Iterasi.....	25	
<b>BAB III METODOLOGI DAN PERANCANGAN</b>			
3.1	Analisa Data .....	27	
3.2	Analisa Sistem .....	28	
3.2.1	Pengolahan Citra ( <i>preprocessing</i> ) .....	29	
3.2.2	Proses Jaringan Syaraf Tiruan .....	35	
3.2.2.1	Struktur Jaringan Syaraf Tiruan.....	35	
3.2.2.2	Proses Pelatihan.....	36	
3.2.2.3	Proses Pengenalan .....	40	
3.3	Perancangan Uji Coba .....	43	
3.3.1	Citra Uji.....	43	
3.3.2	Pengujian Pengenalan Kanker Payudara .....	43	
3.4	Contoh Perhitungan .....	44	
3.5.	Rancangan Antarmuka ( <i>interface</i> ) .....	55	
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN</b>			
4.1	Lingkungan Implementasi .....	57	
4.1.1	Lingkungan Perangkat Keras.....	57	
4.1.2	Lingkungan Perangkat Lunak.....	57	
4.2	Implementasi Program .....	57	
4.2.1	Implementasi Proses <i>Pre procesing</i> .....	57	
4.2.2	Implementasi Proses JST .....	61	
4.2.2.1	Implementasi Proses Pelatihan .....	62	
4.2.2.2.	Implementasi Proses Pengenalan.....	69	
4.3	Implementasi Antarmuka ( <i>Interface</i> ) .....	70	
4.4	Hasil dan pembahasan .....	73	
4.4.1	Hasil Percobaan .....	73	
4.4.2	Analisa Hasil percobaan.....	76	
4.4.3	Penetapan Struktur Jaringan Syaraf tiruan .....	83	
4.4.4	Hasil Pelatihan.....	84	
4.4.5	Hasil Pengenalan Kanker Payudara.....	88	
4.4.6	Uraian .....	92	
<b>BAB V KESIMPULAN DAN SARAN</b>			
5.1	Kesimpulan .....	95	
5.2	Saran .....	95	
<b>DAFTAR PUSTAKA</b> .....			97

**DAFTAR GAMBAR**

Halaman

Gambar 2.1	Pengambilan foto <i>Cranio-caudal (CC)</i> <i>dan mediolateral oblique (MLO).....</i>	7
Gambar 2.2	Posisi <i>Cranio-caudal (CC).....</i>	8
Gambar 2.3	Hasil mammografi <i>Cranio-caudal (CC).....</i>	8
Gambar 2.4	Posisi <i>mediolateral oblique (MLO).....</i>	8
Gambar 2.5	Hasil mammografi <i>mediolateral oblique(MLO).....</i>	9
Gambar 2.6	Foto <i>Mammogram Normal dan Foto Mammogram Kena Kanker.....</i>	9
Gambar 2.7	Foto <i>Mammogram Normal dan Foto Mammogram Kena Kanker.....</i>	9
Gambar 2.8	<i>Mammogram Report.....</i>	10
Gambar 2.9	Representasi citra kedalam matriks angka.....	11
Gambar 2.10	<i>Single layer network .....</i>	15
Gambar 2.11	<i>Multi layer network .....</i>	15
Gambar 2.12	<i>Reccurent network .....</i>	16
Gambar 2.13	Model Matematis .....	16
Gambar 2.14	Arsitektur <i>Backpropagation .....</i>	19
Gambar 3.1	Proses Data 2.....	28
Gambar 3.2	Diagram alur sistem secara umum.....	29
Gambar 3.3	Proses <i>Prepocesing .....</i>	30
Gambar 3.4	Proses <i>ConvertToImageBiner.....</i>	31
Gambar 3.5	Proses <i>ImageToMatrik .....</i>	32
Gambar 3.6	Proses <i>ImageToBlok.....</i>	33
Gambar 3.7	Proses Normalisasi.....	34
Gambar 3.8	Arsitektur JST Pendeksiyan Kanker Payudara.....	35
Gambar 3.9	Diagram Alir Proses pembelajaran JST .....	37
Gambar 3.10	Diagram Alir Proses <i>Feedforward.....</i>	38
Gambar 3.11	Diagram alir proses <i>Backpropagation.....</i>	41
Gambar 3.12	Diagram alir proses pengenalan.....	42
Gambar 3.13	Antarmuka Pelatihan.....	55
Gambar 3.14	Antarmuka Pengenalan.....	55
Gambar 4.1	Prosedur <i>ConvertToImageBiner .....</i>	58
Gambar 4.2	Fungsi <i>ImageToMeanMatrik .....</i>	59
Gambar 4.3	Fungsi <i>ImageToVarianMatrik .....</i>	60

Gambar 4.4 Prosedur Normalisasi Data .....	61
Gambar 4.5 Prosedur Inisialisasi Bobot .....	63
Gambar 4.6 Prosedur Pelatihan .....	64
Gambar 4.7 Prosedur <i>feedforward</i> .....	65
Gambar 4.8 Prosedur fungsi Hitung <i>Error</i> .....	65
Gambar 4.9 Prosedur Hitung Error di <i>Output Layer</i> .....	66
Gambar 4.10 Prosedur Hitung Error di <i>Hidden layer</i> .....	66
Gambar 4.11 Prosedur <i>Update</i> Bobot.....	67
Gambar 4.12 Prosedur Simpan.....	68
Gambar 4.13 Prosedur Pengenalan.....	70
Gambar 4.14 Antarmuka Utama.....	71
Gambar 4.15 Antarmuka Pelatihan .....	72
Gambar 4.16 Antarmuka Pengenalan.....	72
Gambar 4.17 Grafik Pengaruh Jumlah <i>Neuron</i> .....	74
Gambar 4.18 Grafik Pengaruh <i>Learning Rate</i> .....	75

UNIVERSITAS BRAWIJAYA



## DAFTAR TABEL

Halaman

Tabel 3.1	Pengaruh Karakteristik data input .....	43
Tabel 3.2	Pengaruh jumlah neuron terhadap waktu iterasi.....	43
Tabel 3.3	Pengaruh <i>Learning rate</i> pada kesalahan minimal .....	43
Tabel 3.4	Hasil pengujian.....	44
Tabel 3.5	Data input matrik 4 x 6.....	44
Tabel 3.6	Hasil Normalisasi .....	45
Tabel 3.7	Bobot awal ke lapisan <i>hidden</i> ( $v_{ij}$ ).....	45
Tabel 3.8	Bobot awal ke lapisan <i>output</i> ( $w_{jk}$ ).....	46
Tabel 3.9	Hitung keluaran unit <i>hidden</i> ( $Z_j$ ).....	46
Tabel 3.10	Hasil Aktivasi .....	46
Tabel 3.11	Selisih bobot $W_{jk}$ .....	47
Tabel 3.12	Faktor penimbang di unit <i>hidden</i> .....	47
Tabel 3.13	Aktivasi faktor Penimbang di unit <i>hidden</i> .....	47
Tabel 3.14	Perbaikan bobot penimbang $\Delta V_{ij}$ .....	48
Tabel 3.15	Bobot baru <i>hidden-output layer</i> .....	48
Tabel 3.16	Bobot baru <i>input-hidden layer</i> .....	49
Tabel 3.17	Bobot awal ke lapisan <i>hidden</i> ( $V_{ij}$ ) .....	50
Tabel 3.18	Bobot awal ke lapisan <i>output</i> ( $w_{jk}$ ).....	50
Tabel 3.19	Operasi pada <i>hidden</i> .....	51
Tabel 3.20	Hasil Aktivasi pada <i>Hidden</i> .....	51
Tabel 3.21	Selisih Bobot $W_{jk}$ .....	52
Tabel 3.22	Faktor Penimbang di unit <i>hidden</i> .....	52
Tabel 3.23	Aktivasi faktor Penimbang di unit <i>hidden</i> .....	52
Tabel 3.24	Perbaikan bobot penimbang $\Delta V_{ij}$ .....	53
Tabel 3.25	Bobot baru <i>hidden-output layer</i> .....	53
Tabel 3.26	Bobot baru <i>input-hidden layer</i> .....	54
Tabel 4.1	Pengaruh jumlah <i>neuron</i> .....	73
Tabel 4.2	Pengaruh <i>Learning rate</i> terhadap <i>epoch</i> .....	75
Tabel 4.3	Hasil Percobaan Pengaruh Ukuran Blok <i>Mammogram</i> .....	76
Tabel 4.4	Hasil pengenalan dengan <i>learning rate</i> 2.0 .....	77
Tabel 4.5	Hasil pengenalan dengan <i>learning rate</i> 1.4 .....	77
Tabel 4.6	Hasil pengenalan dengan <i>learning rate</i> 2.0 .....	78

Tabel 4.7	Hasil pengenalan dengan <i>learning rate</i> 1.4 .....	78
Tabel 4.8	Hasil pengenalan dengan <i>learning rate</i> 2.0 .....	79
Tabel 4.9	Hasil pengenalan dengan <i>learning rate</i> 1.4 .....	79
Tabel 4.10	Hasil pengenalan dengan <i>learning rate</i> 2.0 .....	80
Tabel 4.11	Hasil pengenalan dengan <i>learning rate</i> 1.4 .....	80
Tabel 4.12	Hasil pengenalan dengan <i>learning rate</i> 2.0 .....	81
Tabel 4.13	Hasil pengenalan dengan <i>learning rate</i> 1.4 .....	81
Tabel 4.14	Hasil pengenalan dengan <i>learning rate</i> 2.0 .....	82
Tabel 4.15	Hasil pengenalan dengan <i>learning rate</i> 1.4 .....	82
Tabel 4.16	Hasil percobaan uji kemampuan struktur jaringan ...	83
Tabel 4.17	Hasil pelatihan dan tes pengenalan dengan <i>mean value</i> pada payudara kiri .....	84
Tabel 4.18	Hasil pelatihan dan tes pengenalan dengan <i>mean value</i> pada payudara kanan .....	85
Tabel 4.19	Hasil pelatihan dan tes pengenalan dengan <i>variance value</i> pada payudara kiri .....	85
Tabel 4.20	Hasil pelatihan dan tes pengenalan dengan <i>variance value</i> pada payudara kanan .....	86
Tabel 4.21	Hasil pelatihan dan tes pengenalan dengan perbandingan <i>mean variance value</i> pada payudara kiri.....	87
Tabel 4.22	Hasil pelatihan dan tes pengenalan dengan perbandingan <i>variance value</i> pada payudara kanan .....	87
Tabel 4.23	Hasil Pengujian data yang sudah dilatih.....	88
Tabel 4.24	Hasil Pengujian data yang belum di latih.....	88
Tabel 4.25	Hasil Pengujian data yang belum di latih secara keseluruhan berdasarkan data input yang digunakan	89
Tabel 4.26	Hasil Pengujian data keseluruhan.....	90
Tabel 4.27	Hasil pengujian data keseluruhan berdasarkan data input yang digunakan .....	91
Tabel 4.28	Hasil Percobaan Data dari RS.Onkologi dengan Threshold=50 .....	91



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Berdasarkan laporan terbaru dari Cancer.org, kanker payudara menduduki urutan kedua terbanyak penyebab kematian pada wanita. Kanker payudara biasanya terjadi diatas usia 30 tahun, terbanyak usia 45 tahun ke atas. Sebagian besar kanker payudara ditemukan secara tidak sengaja oleh penderita sendiri atau pada waktu berkunjung ke dokter. Pencegahan kanker payudara dapat dilakukan secara dini, salah satunya dengan pemeriksaan menggunakan *mammografi*.

*Mammografi* adalah pemeriksaan menggunakan sinar *rotgen* dosis rendah untuk melihat bagian dalam payudara. Hasil dari *mammografi* disebut *mammogram*, yang biasanya digunakan untuk membantu proses awal pendekripsi dan *diagnosa* kanker payudara pada wanita. Hasil *mammografi* yang berupa *mammogram* digunakan untuk mendekripsi sejumlah kelainan pada payudara, seperti pengkapuran atau *calcification*. Proses pengkapuran adalah pengendapan kapur pada jaringan payudara, yang nampak seperti daerah putih pada foto *mammogram* [Siemens-Elema AB, Division SPS, 1999].

Walaupun teknologi *mammografi* telah banyak mengalami kemajuan, ada komunitas medis yang meragukan penggunaan *mammografi* karena tingkat kesalahan yang masih tinggi dan karena hasil radiasi pada gambar *mammogram* yang kurang jelas [Wartamedika, 2008].

Selama ini dalam melakukan identifikasi kanker payudara masih dilakukan secara manual, yaitu berdasarkan hasil keluaran *mammogram* jaringan payudara dan pengalaman para dokter radiolog. Sementara dalam proses identifikasi yang melibatkan tim dokter terkadang terdapat perbedaan *diagnosa*. Sehingga bagaimana komputer bisa melakukan pendekripsi kanker dengan *mammogram* untuk membantu pengenalan yang bisa digunakan sebagai *second opinion* pada pengambilan keputusan saat proses diagnosa. Jaringan Syaraf tiruan (*Artificial Neural Network*) dapat digunakan sebagai

alat bantu untuk mendeteksi kanker payudara berdasarkan jaringan yang terdapat pada *mammogram*.

Telah banyak dilakukan penelitian mengenai kanker payudara, misalnya, *Classification of breast MRI lesions using a backpropagation neural network (BNN)* [Arbach L., Stolpen A., and Reinhardt MJ], *Breast Cancer diagnosis using statistical Neural Network* [Tuba Kiyan ,Tulay Dirim, 2004] , *neural network approach for mammography diagnosis using wavelets feature* [Essam A. Rashed and Mohamad G. Awad, 2006], *Analisis Mammographic Microcalcifications Menggunakan Ciri-Ciri Struktur Citra Gray-Level* [Gunawan, 2005].

Pada penelitian ini disajikan metode pendekstian kanker payudara menggunakan jaringan syaraf tiruan. Jaringan syaraf tiruan adalah salah satu metode komputasi yang berusaha meniru cara kerja sistem syaraf otak yang ada pada manusia, yaitu dengan cara menyimpan informasi terhadap obyek-obyek yang telah dikenali, kemudian informasi yang telah disimpan tersebut nantinya akan digunakan untuk mengenali obyek-obyek berikutnya yang akan diujikan.

Jaringan Syaraf Tiruan memerlukan suatu proses pembelajaran. Berdasarkan cara pembelajarannya Jaringan Syaraf Tiruan dibagi menjadi dua yaitu *supervised learning* (terawasi) dan *unsupervised learning* (tidak terawasi). Pada jaringan syaraf tiruan *supervised learning* diberi *input* tertentu dan *outputnya* ditentukan oleh pengajarnya. Sebaliknya jaringan syaraf tiruan *unsupervised* secara mandiri jaringan syaraf tiruan akan mengatur keluarannya sesuai dengan aturan yang dimilikinya [Kristanto, 2004].

Dalam skripsi ini akan dibuat sebuah pendekstian kanker payudara menggunakan *mammogram* dengan Jaringan Syaraf Tiruan dengan algoritma pembelajaran *backpropagation*.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahannya yaitu:

1. Bagaimana merancang arsitektur Jaringan Syaraf Tiruan *Backpropagation* yang digunakan untuk mendeteksi kanker payudara.
2. Berapa tingkat keakuratan untuk pendekstian kanker payudara.

### 1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah disebutkan, maka pembahasan dibatasi pada:

1. Data yang digunakan adalah gambar *mammogram* yang berekstensi *bitmap* (.bmp).
2. Metode pengenalan pola jaringan menggunakan Jaringan Syaraf Tiruan dengan pelatihan menggunakan *backpropagation*.

### 1.4 Tujuan Penelitian

Tujuan dari skripsi ini adalah:

1. Membuat perangkat lunak yang mampu mendeteksi kanker pada payudara serta melakukan pelatihan dan pengujian terhadap sistem yang telah dibuat.
2. Mengetahui dan menghitung tingkat akurasi pendekripsi kanker payudara.

### 1.5 Manfaat Penelitian

Manfaat dari penulisan skripsi ini adalah agar teknologi pendekripsi kanker payudara dengan *mammogram* menggunakan algoritma jaringan syaraf tiruan propagasi balik (*backpropagation*) yang dibangun dapat digunakan, sehingga hasil pendekripsi kanker payudara dapat diperoses menjadi informasi yang diperlukan.

### 1.6 Metodologi Penelitian

1. Studi Literatur

Studi ini dilakukan dengan cara mencari sekaligus mempelajari beberapa literatur dan artikel mengenai kanker payudara dan memahami pola pada jaringan payudara yang normal dan *abnormal*. Disamping itu juga mempelajari metode yang digunakan untuk mendeteksi kanker payudara menggunakan Jaringan Syaraf Tiruan, sebagai acuan dalam perencanaan dan pembuatan skripsi.

2. Perancangan dan Pembuatan Program

Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut dalam suatu program komputer.

3. Uji coba dan analisa hasil implementasi  
Menguji perangkat lunak pada beberapa data yang berbeda dan menganalisa hasil dari implementasi tersebut kemudian melakukan evaluasi
4. Penyusunan Laporan  
Membuat laporan tertulis mengenai hasil tugas akhir ini.

## 1.7 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika penulisan sebagai berikut:

### 1. BAB I PENDAHULUAN

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

### 2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan kanker payudara dan proses deteksi kanker payudara menggunakan metode Jaringan Syaraf Tiruan jenis *Backpropagation*.

### 3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam pendekripsi kanker payudara..

### 4. BAB IV IMPLEMENTASI DAN UJI COBA SISTEM

Dalam bab ini dijelaskan mengenai implementasi program, uji coba dan analisanya

### 5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.

## DAFTAR PUSTAKA

## LAMPIRAN

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Kanker Payudara

Kanker Payudara adalah tumor ganas yang tumbuh di dalam jaringan payudara. Kanker bisa mulai tumbuh di dalam kelenjar susu, saluran susu, jaringan lemak maupun jaringan ikat pada payudara. Secara sederhana menjadi wanita dan menjadi tua merupakan resiko untuk terkena kanker payudara.

Kanker pada stadium awal jarang menimbulkan gejala, karena itu sangat penting untuk melakukan penyaringan. Beberapa prosedur yang digunakan untuk penyaringan kanker payudara adalah :

##### 1. SADARI (Pemeriksaan Payudara Sendiri).

Bagi wanita yang masih mengalami menstruasi, waktu yang paling tepat untuk melakukan SADARI adalah 7-10 hari sesudah hari 1 menstruasi. Bagi wanita pasca *menopause*, SADARI bisa dilakukan kapan saja, tetapi secara rutin dilakukan setiap bulan (misalnya setiap awal bulan).

##### 2. Mammografi.

Pada pemeriksaan dengan *mammografi* digunakan *sinar X* dosis rendah untuk menemukan daerah yang *abnormal* pada payudara. Para ahli menganjurkan kepada setiap wanita yang berusia diatas 40 tahun untuk melakukan *mammografi* secara rutin setiap 1-2 tahun dan pada usia 50 tahun ke atas *mammografi* dilakukan sekali/tahun.

##### 3. USG payudara.

USG digunakan untuk membedakan kista (kantung berisi cairan) dengan benjolan padat.

##### 4. Termografi.

Pada *termografi* digunakan suhu untuk menemukan kelainan pada payudara

[<http://www.cancerhelps.com/kanker-payudara.htm>].

#### 2.2 Mammografi

*Mammografi* adalah pemeriksaan menggunakan sinar rotgen dosis rendah untuk melihat bagian dalam payudara. Hasil *mammografi* yang berupa *mammogram* digunakan untuk mendeteksi

sejumlah kelainan pada payudara. Keadaan-keadaan yang ditemukan pada *mammogram* seperti *calcification* (kalsifikasi) dan *Mass* (massa).

Kalsifikasi adalah penumpukan mineral atau pengendapan zat kapur pada jaringan payudara, yang nampak seperti daerah putih pada foto *mammogram*. Ada 2 macam kalsifikasi yaitu *microcalcification* dan *macrocalcification*. *Microcalcification* merupakan bintik kecil berupa zat kapur yang ada dalam payudara, berukuran kurang lebih 1/50 inci. Apabila terdapat beberapa *microcalcification* pada suatu area dan membentuk sebuah *cluster* maka bisa diindikasikan sebagai sebuah kanker kecil. Sedangkan *macrocalcification* merupakan pengendapan zat kapur yang besar pada payudara yang biasanya sering dihubungkan dengan perubahan yang buruk pada payudara, seperti pengerasan nadi pada payudara, luka-luka dan peradangan.

*Massa* merupakan sekelompok sel yang bergerombol secara rapat disekeliling jaringan. Para radiolog sering menggunakan USG atau tindakan lebih lanjut untuk membantu membedakan antara massa yang padat dengan cair (bisul). Massa berupa kista, tumor jinak, atau kanker payudara.

Pada *mammogram* daerah kelenjar pada jaringan payudara ditampilkan dengan daerah putih. Secara umum wanita lebih muda lebih banyak memiliki jaringan pada payudara dibandingkan dengan wanita yang lebih tua. Kepadatan jaringan payudara bisa mempersulit untuk mendeteksi kalsifikasi dan massa dengan *mammografi*.

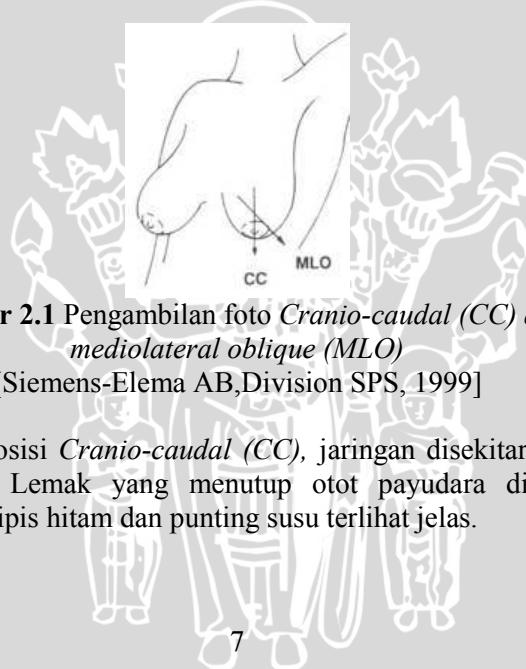
### 2.2.1 Tipe *Mammografi*

*Mammografi* tidak dapat menentukan apakah jaringan sudah berubah menjadi kanker, tetapi hanya memberikan kesimpulan bahwa suatu jaringan dicurigai telah terkena kanker. Untuk memastikannya diperlukan pemeriksaan selanjutnya seperti biopsi. Biopsi yaitu pengambilan jaringan payudara dengan jarum atau melalui pembedahan, kemudian diperiksa di bawah mikroskop. Beberapa tipe *mammografi* adalah:

- *Mammografi* skrining. *Mammografi* ini dilakukan pada wanita yang tidak menunjukkan tanda-tanda kanker payudara.

- *Mammografi* diagnostik. *Mammografi* yang digunakan untuk mendiagnosis gejala-gejala yang timbul pada payudara seperti benjolan, nyeri, penebalan puting susu, keluarnya cairan melalui puting susu, atau perubahan bentuk dan ukuran payudara.

*Mammografi* bisa menemukan 85-90% kanker payudara pada wanita usia diatas 50 tahun, bahkan bisa menemukan suatu gumpalan pada payudara sebelum dua tahun kanker bisa dirasakan oleh penderita. Saat gumpalan ditemukan, *mammografi* bisa mendeteksi gumpalan tersebut untuk menentukan kanker. Jika ditemukan kelainan pada payudara dengan *mammografi* maka akan dilakukan pendekstian lainnya dengan menggunakan *image* pada daerah payudara seperti USG atau biopsi lainnya. Seringkali USG digunakan untuk bantuan para radiolog atau ahli bedah untuk menentukan apakah payudara berisi sel kanker. Pemeriksaan dengan *mammografi* sangat sederhana, payudara pasien secara bergantian diletakkan dan ditekan mendatar antara 2 lembar alas, kemudian difoto *rontgen* dari atas ke bawah (*Cranio Caudal, CC*) dan miring (*mediolateral oblique,MLO*). Semakin datar posisi payudara,akan semakin bagus gambar yang dihasilkan[Siemens-Elema AB,Division SPS, 1999].



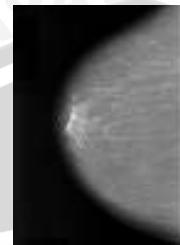
**Gambar 2.1** Pengambilan foto *Cranio-caudal (CC)* dan *mediolateral oblique (MLO)*

[Siemens-Elema AB,Division SPS, 1999]

Melalui posisi *Cranio-caudal (CC)*, jaringan disekitar payudara dapat difoto. Lemak yang menutup otot payudara ditampilkan sebagai garis tipis dan punting susu terlihat jelas.



**Gambar 2.2** Posisi *Cranio-caudal (CC)*  
[Siemens-Elema AB, Division SPS, 1999]



**Gambar 2.3** Hasil mammografi *Cranio-caudal (CC)*  
[Siemens-Elema AB, Division SPS, 1999]

Pada posisi *mediolateral oblique (MLO)*, pelat *mammografi* diletakkan pada sudut  $45^{\circ}$  dari payudara pasien. Pada gambar, otot dada sampai punting susu difoto miring dari atas, jaringan di sekitar ketiak juga tampak jelas. Hal ini mengetahui apakah sel kanker payudara sudah mulai menyebar ke kelenjar getah bening di sekitar ketiak atau belum.



**Gambar 2.4** Posisi *mediolateral oblique (MLO)*  
[Siemens-Elema AB, Division SPS, 1999]

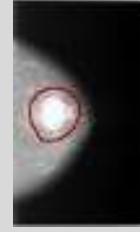


**Gambar 2.5** Hasil mammografi mediolateral oblique (*MLO*)  
[Siemens-Elema AB, Division SPS, 1999]

Pada Gambar 2.6 dan Gambar 2.7 merupakan perbedaan foto *mammogram* antara payudara normal dan yang terdapat kelainan atau terkena kanker.



a



b

**Gambar 2.6** a. Foto *Mammogram* Normal dan b. Foto *Mammogram* Kena Kanker  
[<http://marathon.csee.usf.edu/Mammography/Database.html>]



a



b

**Gambar 2.7** a. Foto *Mammogram* Normal dan b. Foto *Mammogram* Kena Kanker  
[Sumber : RS Onkologi Surabaya]

## 2.2.2 Mammogram Report

*Mammogram report* merupakan hasil laporan yang merupakan komponen utama pada proses *mammografi*. *Mammogram report* diberikan kepada pasien yang disertakan dengan hasil foto *mammogramnya* untuk mengetahui kelanjutan dari hasil *mammografi*. Pada *mammogram report* berisi lengkap data pasien, tipe *mammografi* yang digunakan, hasil pemeriksaan, keputusan kelanjutan pemeriksaan, anjuran, serta catatan dari hasil pemeriksaan. Lebih jelasnya bisa dilihat pada Gambar 2.8

*Mammogram* saja tidak bisa menentukan apakah seorang perempuan mempunyai kanker payudara atau tidak, tetapi dengan *mammogram report* kemungkinan bisa menunjukkan kelainan yang ada dan bisa menggambarkan kelainan tersebut.



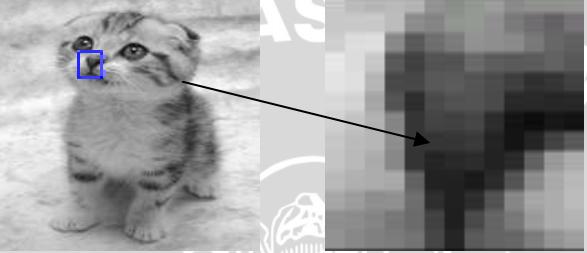
The image shows a scanned copy of a 'MAMMOGRAM REPORT' from RS.Oncology Surabaya. The report includes fields for patient information (Name: Sri Dwi Lestari, Age: 30, Address: Jl. Veteran No. 12, City: Surabaya, Province: East Java, Phone: 0812-3456789), referring doctor information (Name: dr. Agus Djoko, Address: Jl. Veteran No. 12, City: Surabaya, Specialization: Oncologist, Screening: Yes), examination date (11/05/2009), and a grid for assessing lesion location (Right). The report also contains sections for breast history, findings (e.g., no palpable abnormalities, no mammographic signs of breast cancer), and recommendations (e.g., refer to a breast specialist, follow-up mammogram in 6 months). A signature at the bottom right reads 'dr. Dwi Lestari, M.Kes'.

Gambar 2.8 Mammogram Report  
[Sumber :RS.Oncologi Surabaya]

### 2.3 Pengolahan Citra Digital

Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom, dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra, atau *pixel (picture elemen)* [Gonzales and Woods, 1992]. Pengolahan citra pada dasarnya dilakukan dengan cara memodifikasi setiap titik dalam citra tersebut sesuai kebutuhan[Balza, A dan Kartika, F.,2005].

Sebuah citra dalam komputer direpresentasikan dalam sebuah matriks yang berisikan angka-angka. Jika menyimpan citra kedalam sebuah file, maka yang disimpan ke dalam file tersebut adalah angka-angka yang diperoleh dari matriks kanvas seperti yang ditunjukkan pada Gambar 2.9.



201	188	181	185	180	147	140	149	155	138	144	144	145
199	200	201	188	139	132	147	150	143	123	112	102	117
207	221	222	136	90	111	125	145	140	138	122	104	97
231	219	200	90	65	84	84	107	95	92	92	99	89
227	223	181	74	72	89	92	86	77	63	50	55	65
217	211	166	85	47	75	82	83	75	42	42	39	40
208	195	179	131	54	68	66	72	46	21	15	24	19
198	187	181	141	53	54	55	59	37	21	37	66	90
195	184	170	134	52	38	42	45	35	43	98	152	172
186	175	171	169	100	34	34	27	44	85	139	170	184
167	156	142	144	112	48	32	46	84	133	166	172	186
142	139	131	120	108	67	30	76	102	123	153	171	178
145	134	128	125	117	70	38	91	101	105	125	146	157

**Gambar 2.9** Representasi citra kedalam matriks angka  
[Murni Aniati,dkk.,2005]

### 2.3.1 Representasi Warna Keabuan (*Gray Scale*)

*Grayscale* merupakan suatu format gambar pada sistem *digital image*. Pada suatu *image* format RGB (*Red, Green, Blue*) mewakili gambar pada sistem digital untuk setiap *pixel*, dimana masing-masing komponen warna diwakili dengan satu *byte*. Karena nilai variasi yang ada pada masing - masing komponen RGB adalah 0 sampai 255, maka total nilai variasi yang dihasilkan untuk sistem warna ini adalah  $256 * 256 * 256$ . Sistem gambar *grayscale* memerlukan satu *byte* (delapan *bit*) untuk penyimpanan data, mempunyai kemungkinan wama dan 0 (hitam) sampai 255 (putih), dalam hal ini nilai variasi RGB untuk sistem gambar *grayscale* adalah sama. Konversi dari sistem warna RGB menjadi *gray scale* ini ada beberapa macam, salah satunya menggunakan sistem YUV (sistem warna pada NTSC) yaitu dengan cara mengambil komponen Y (Illuminasi), komponen Y sendiri dapat diperoleh dari sistem warna RGB dengan konversi:

$$Y = (0.299 * \text{Red}) + (0.587 * \text{Green}) + (0.114 * \text{Blue}) \quad (2.1)$$

### 2.3.2 *Thresholding*

*Thresholding* digunakan untuk mengatur derajat keabuan yang ada pada suatu citra. Dengan *thresholding* maka derajat keabuan dapat diubah sesuai dengan keinginan. Teknik *thresholding* ini memisahkan bagian gambar yang sesuai dengan obyek yang diteliti dan latar belakangnya. Metode ini juga digunakan untuk mengkonversi data citra menjadi data biner dengan tujuan agar proses pengolahan citra dapat dilakukan dengan lebih mudah.

### 2.3.3 Karakteristik Citra

Kuantisasi rata-rata adalah suatu metode pada gambar dimana suatu gambar dibagi menjadi bagian-bagian gambar yang lebih kecil (*resampling*) dan nilai pada setiap bagian sampling diwakili oleh hasil rata-rata terhadap bagian sampling tersebut. Pada proses kuantisasi rata-rata ini digunakan gambar ukuran 200x300 dan ukuran sampling 10x10, maka akan diperoleh 20x30 bagian sampling. Disamping ukuran sampling di atas, pada penelitian ini digunakan juga ukuran sampling yang lainnya.

Kuantisasi rata-rata yang digunakan dalam proses pembentukan ciri, mempunyai persamaan sebagai berikut:

$$\bar{x} = \frac{\sum_{i=0}^{n_i} \sum_{j=0}^{n_j} x_{i,j}}{n_i \cdot n_j} \quad (2.2)$$

dengan,

$\bar{x}$  : nilai rata-rata dari setiap bagian sampling,

$x_{i,j}$  : nilai citra pada titik  $i,j$  dalam satu bagian sampling,

$n_i, n_j$  : masing-masing adalah ukuran baris dan kolom pada tiap bagian sampling.

Dalam menentukan nilai pada bagian sampling selain menggunakan hasil rata-rata, bisa juga dengan menggunakan varian. Varian untuk masing-masing sampling dapat dihitung dengan persamaan 2.3

$$\sigma^2 = \frac{\sum_{i=0}^{n_i} \sum_{j=0}^{n_j} (x_{i,j} - \bar{x})^2}{n_i \cdot n_j} \quad (2.3)$$

## 2.4 Jaringan Syaraf Tiruan

### 2.4.1 Definisi Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (*artificial neural networks*) atau disingkat JST adalah sistem komputasi dimana arsitekturnya dan operasinya diilhami dari pengetahuan tentang sel syaraf biologi dalam otak. Metode jaringan syaraf tiruan berusaha meniru cara kerja sistem syaraf otak yang ada pada manusia, yaitu dengan cara menyimpan informasi terhadap obyek-obyek yang telah dikenali, kemudian informasi yang telah disimpan tersebut nantinya akan digunakan untuk mengenali obyek-obyek berikutnya yang akan diujikan. Berdasarkan kemampuan yang dimiliki, Jaringan Syaraf Tiruan dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh, untuk menghasilkan *output* yang sempurna dari contoh atau *input* yang dimasukkan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan

karakteristik dari *input* yang akan disimpan kepadanya [Kristanto, 2004].

Jaringan syaraf tiruan terdiri dari sekumpulan unit pemrosesan yang dikenal sebagai unit *neuron*. Kumpulan unit *neuron* tersebut saling terhubung dan saling berkerja sama dalam mencari solusi dari sebuah permasalahan. Permasalahan yang cocok untuk ditangani oleh Jaringan saraf tiruan terbagi menjadi tiga kelompok, yaitu:

1. Permasalahan klasifikasi

Permasalahan penentuan kelas yang cocok untuk sebuah masukan dengan pola tertentu.

2. Prediksi pola

Permasalahan pembuatan suatu pola secara lengkap dari sebagian masukan pola dan memprediksi kecenderungan yang mungkin muncul dari data masukan tersebut.

3. Kompresi data

Permasalahan pengurangan jumlah bit dari suatu blok data untuk disimpan atau dikirim dalam batas-batas kesalahan yang diperkenankan. Sebuah pola tertentu dapat dikenali dari data masukan dan dijadikan basis penmbangkitan data keluaran dengan ukuran lebih kecil [Arif H., 2006].

#### 2.4.2 Arsitektur jaringan syaraf tiruan

Suatu jaringan syaraf minimum tersusun atas *input layer* dan *output layer*. Dalam beberapa tipe jaringan diantara *input layer* dan *output layer* terdapat *hidden layer*. *Input layer* merupakan aktifitas unit-unit lapisan masukan yang menunjukkan informasi dasar yang kemudian digunakan dalam jaringan syaraf tiruan. *Hidden layer* merupakan aktifitas setiap unit-unit lapisan tersembunyi ditentukan oleh aktifitas dari unit-unit masukan dan bobot dari koneksi antara unit-unit masukan dan unit-unit lapisan tersembunyi. Sedangkan *output layer* merupakan karakteristik dari unit-unit keluaran tergantung dari aktifitas unit-unit lapisan tersembunyi dan bobot antara unit-unit lapisan tersembunyi dan unit-unit keluaran.

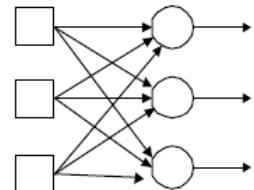
Hal ini berarti bahwa semua neuron pada *input layer* akan berhubungan ke semua neuron dalam *hidden layer* yang selanjutnya setiap *neuron* dalam *hidden layer* nantinya akan dihubungkan ke semua *neuron* di *output layer*. Pada setiap *layer* biasanya *neuron*

mempunyai fungsi aktivasi serta pola hubungan ke *neuron* lain yang sama.

Jaringan syaraf tiruan dapat dibagi menjadi beberapa kelompok. Menurut Siang J.J. (2005) berdasarkan jumlah *layernya* jaringan syaraf tiruan dibagi menjadi:

a. *Single-Layer Network*

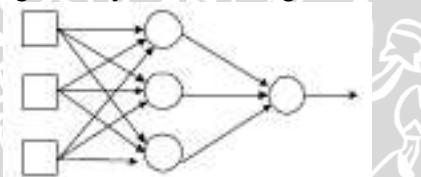
*Single layer network* ini pertama kali dirancang oleh Widrow dan Hoff pada tahun 1960. *Neural network* jenis ini memiliki koneksi pada *inputnya* secara langsung ke jaringan *outputnya*. Pada jaringan ini, semua unit input dihubungkan langsung ke unit *output*, meskipun dengan bobot yang berbeda-beda. *Single layer network* ini tidak mempunyai *hidden layer*.



Gambar 2.10 Single layer network [Saputra,2002]

b. *Multi layer network*

Jaringan ini merupakan perluasan dari *single layer network*. Pada jaringan ini selain terdapat unit *input* dan *output* juga terdapat unit lain yang biasanya disebut dengan *hidden layer*.

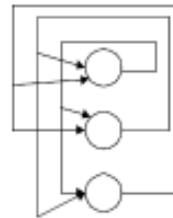


Gambar 2.11 Multi layer network [Saputra,2002]

*Multi layer network* ini dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan *single layer network*, meskipun kadang proses pelatihan lebih kompleks dan lama.

### c. Recurrent network

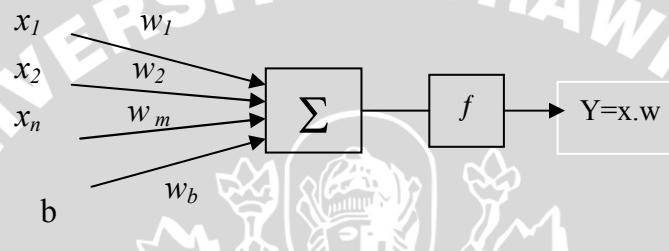
Model jaringan ini sama seperti *single layer network* maupun *multi layer network*. Hanya saja ada *neuron output* yang memberikan sinyal pada unit *input* (*feedback loop*).



Gambar 2.12 Recurrent network [Saputra,2002]

### 2.4.3 Model Matematis

Model JST merupakan kondisi topologi dari interkoneksi serta aturan yang ada di dalamnya.



Gambar 2.13. Model matematis JST

Vektor input  $\mathbf{x}$ , diberikan oleh  $x_1, x_2, \dots, x_n$ , adalah input stimulus yang mewakili atribut (*feature*). Nilai koneksi merupakan rangsangan yang memiliki bobot  $w_1, w_2, \dots, w_m$ , secara biologis memiliki analogi dengan kekuatan sinaptis yang dimiliki neuron [Kristanto, 2004].

Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan. Neuron terdiri dari 3 elemen pembentuk:

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot/ kekuatan yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memprlemah sinyal yang dibawahnya.

2. Suatu unit penjumlahan yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain atau tidak [Siang, 2005].

## 2.5 Pelatihan JST

Karakteristik dari JST adalah kemampuannya untuk belajar. Seperti halnya manusia untuk mengenali suatu benda, sistem juga perlu diberi pembelajaran. Namun kemampuan belajar JST bersifat terbatas, sehingga JST tidak dapat melakukan segalanya. Contohnya manusia dapat mempelajari sesuatu, menghafalnya, dan menalarakannya. Berbeda dengan JST, yang dilakukan tidak sampai pada penalaran. Sehingga dalam pembelajarannya dibutuhkan pengawasan atas kebenaran yang diajarkan. Kebenaran pengetahuan yang disimpan dalam sistem ditentukan oleh pengajarnya (manusia) [Kusumadewi, 2003].

Pembelajaran JST merupakan proses pencarian konfigurasi bobot yang sesuai dengan cara perkalian, penjumlahan dan aktivasi bobot dan *input*. Menurut Kosko (1990), JST dikelompokkan menjadi 2 bagian. Pertama bagaimana JST menyimpan pengetahuan atau *encode*, yang kedua bagaimana JST menanggapi dan memproses data yang masuk atau *decode*. Berdasarkan *encode* proses pembelajaran JST dibagi atas metode pembelajaran dibimbing atau *Supervised* dan tidak dibimbing atau *Unsupervised*. Sedangkan berdasarkan *decode* dibedakan menjadi umpan maju atau *feedforward* dan umpan balik atau *feedback* [Kristanto, 2004].

Pembelajaran terawasi atau *supervised learning* adalah proses pembelajaran menacari bobot koneksi yang optimal, dimana penyimpanan pengetahuan sistem harus dibimbing atau diawasi. Hasil pembelajaran akan disimpan atau tidak dapat ditentukan oleh pengajar. Setiap *input* telah diberi pasangan outputnya.

Sedangkan pembelajaran tak terawasi atau *unsupervised learning* merupakan proses pembelajaran yang penyimpanan pengetahuan sistem tidak harus dibimbing. Sistem bekerja sendiri mengatur keluaran sistem sesuai aturan yang dimilikinya [Kristanto, 2004].

## 2.6 Backpropagation (Propagasi balik)

Seperti halnya model JST lain, *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan (Siang,J.J, 2004).

Pelatihan jaringan syaraf tiruan dikatakan berhasil jika pelatihan *konvergen*, dan gagal jika pelatihan *divergen*. Suatu pelatihan dikatakan konvergen jika galat pada setiap iterasi pelatihan selalu mengecil, sampai pada titik dimana nilai bobot pada setiap *neuron* telah mencapai nilai yang paling baik untuk data pelatihan yang diberikan. Sebaliknya, pelatihan dikatakan *divergen* jika galat pada pelatihan tidak cenderung mengecil menuju sebuah titik tertentu.

Menurut Ibrahim Arif, algoritma *backpropagation* memiliki beberapa parameter yang dapat menentukan tingkat efektivitas pembelajaran jaringan syaraf tiruan. Parameter-parameter tersebut adalah:

### 1. Maximum epoch

Menentukan berapa iterasi pembelajaran yang akan dilakukan pada jaringan syaraf tiruan. Semakin besar *maximum epoch* maka tingkat kesalahan jaringan syaraf tiruan akan semakin menurun. Namun, penentuan *maximum epoch* yang terlalu besar akan menyebabkan jaringan syaraf tiruan terlalu mengikuti pola data pelatihan dan meningkatkan kesalahan yang mungkin terjadi ketika jaringan syaraf tiruan diberikan data masukan dari permasalahan sebenarnya. Hal ini dikenal dengan *overfitting*.

### 2. Laju pembelajaran ( $\alpha$ )

Laju pembelajaran menunjukkan seberapa cepat jaringan syaraf tiruan akan menyesuaikan diri dengan data pelatihan yang diterimanya.

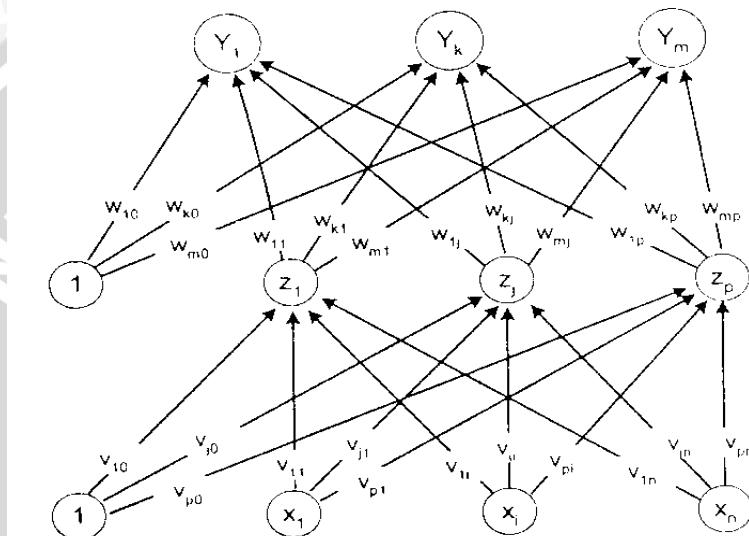
### 3. Momentum ( $\mu$ )

Momentum menunjukkan seberapa besar pelatihan pada iterasi pelatihan saat itu hendak dipengaruhi oleh iterasi pelatihan sebelumnya. Penambahan momentum dimaksukan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (*outlier*).

### 2.6.1 Arsitektur Backpropagation

Jaringan ini memiliki satu lapis masukan ditambah dengan sebuah bias, satu atau lebih lapisan tersembunyi (*hidden layer*) ditambah dengan sebuah bias, dan satu lapisan keluaran. Setiap lapis memiliki *neuron-neuron* (unit-unit). Di antara *neuron* pada satu lapis dengan *neuron* pada lapis berikutnya dihubungkan dengan model koneksi yang memiliki bobot-bobot (*weights*).

Berdasarkan Gambar 2.14,  $v_{ij}$  merupakan bobot dari neuron *input layer*  $x_i$  ke neuron *hidden layer*  $z_j$  ( $v_{j0}$  merupakan bobot yang menghubungkan bias dari neuron *input layer* ke neuron *hidden layer*). Sedangkan  $w_{kj}$  merupakan bobot dari neuron *hidden layer*  $z_j$  ke neuron *output layer*  $y_k$  ( $w_{k0}$  merupakan bobot dari bias di neuron *hidden layer* ke neuron *output layer*  $y_k$ )



Gambar 2.14 Arsitektur Backpropagation  
(Dhaneswara,Giri dan Moertini,SV,2004)

### 2.6.2 Fungsi Aktivasi

Fungsi aktifasi adalah fungsi matematis inputan agar mendekati nilai ambang tertentu. Fungsi aktifasi dalam JST berguna untuk menentukan nilai keluaran dari hasil kolinasi linier masukan dan bobot [Siang, 2005]. Jika nilai keluaran aktifasi berada dalam jangkauan fungsi, maka sinyal akan diteruskan. Terdapat beberapa

macam fungsi aktivasi. Fungsi aktivasi dari algoritma *Bacpropagation* harus berupa fungsi yang dapat diturunkan. Biasanya menggunakan fungsi aktivasi Sigmoid. Fungsi Sigmoid terbagi atas Binary dan Bipolar Sigmoid.

Fungsi aktivasi *Bipolar Sigmoid* mempunyai jangkauan -1 dan 1. Sedangkan *binary sigmoid* mempunyai jangkauan antara 0 dan 1 adalah fungsi yang sering digunakan. Penggunaan fungsi aktifasi harus sesuai dengan *input* dan *output*. Persamaan fungsi *binary sigmoid* [Bose, 2003]:

$$f(x) = \frac{1}{1 + e^{(-\alpha x)}} \quad (2.4)$$

fungsi turunannya adalah:

$$f'(x) = \alpha f(x)[1 - f(x)] \quad (2.5)$$

*Learning rate* ( $\alpha$ ) merupakan laju pembelajaran dan e adalah bilangan euler.

### 2.6.3 Normalisasi Data

Normalisasi data digunakan untuk menyamakan skala data ke dalam jangkauan nilai tertentu, misal 0 sampai dengan 1. Jika menggunakan fungsi sigmoid biner maka data harus ditransformasi karena range keluaran fungsi sigmoid biner adalah range [0,1]. Data akan ditransformasikan ke interval [0,1] tapi akan lebih baik jika ditransformasikan pada interval yang lebih kecil [0.1,0.9] mengingat fungsi sigmoid merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1[Siang,J.J, 2004].

Transformasi linier yang dipakai untuk mentransformasikan data ke interval [0.1,0.9] adalah:

$$x' = \frac{0.8(x - a)}{b - a} + 0.1 \quad (2.6)$$

Dimana a adalah data minimum dan b adalah data maksimum.

### 2.6.4 Algoritma Backpropagation

Pelatihan sebuah jaringan *backpropagation* terdiri dari tiga langkah yaitu pelatihan pola *input* secara *feedforward*, *backpropagation* dari kumpulan kesalahan dan penyesuaian bobot (Kristanto, 2004). Pelatihan dilakukan berulang-ulang dan berhenti jika telah mencapai batas iterasi maksimum yang ditentukan dan nilai *error* kurang dari *Mean Square Error* (MSE). Ketepatan algoritma *backpropagation* ditentukan dengan *Mean Square Error* (MSE). Semakin kecil nilai MSE maka dapat dianggap bahwa arsitektur jaringan semakin baik, demikian pula sebaliknya. MSE dihitung dengan Persamaan 2.7

$$\text{MSE} = \frac{1}{2} \sum_p (t_p - y_p)^2 \quad (2.7)$$

dimana p adalah jumlah *neuron* (unit) *output*, t adalah target dan y adalah output.

Algoritma pelatihan *backpropagation* adalah sebagai berikut :

1. Inisialisasi bobot

Bobot awal ditentukan secara acak dengan nilai sekecil mungkin, misalkan [-0.05, 0.05].

2. Inisialisasi *error* target dan jumlah iterasi maksimum.

3. Selama iterasi kurang dari iterasi maksimum dan *error* melebihi MSE.

- a. Lakukan *feedforward* untuk setiap pasangan pelatihan:

1. Tiap-tiap unit *input* ( $x_i$ ,  $i=1,2,\dots,n$ ) menerima sinyal dan menyalurkan sinyal ini ke semua unit lapisan tersembunyi.
2. Tiap-tiap unit tersembunyi ( $z_j$ ,  $j=1,\dots,p$ ) menjumlahkan perkalian nilai input dan bobot sinyal input dengan Persamaan 2.8.

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.8)$$

Gunakan fungsi aktivasi untuk menghitung *outputnya* yaitu:

$$z_j = f(z\_in_j) \quad (2.9)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (lapisan *output*).

3. Tiap-tiap unit *output* ( $y_k$ ;  $k=1,\dots,m$ ) akan menjumlahkan bobot sinyal output dengan Persamaan 2.10.

$$y\_in_k = v_{0k} + \sum_i z_j w_{jk} \quad (2.10)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*, ditunjukkan oleh Persamaan 2.11.

$$y_k = f(y\_in_k) \quad (2.11)$$

- b. Lakukan *Backpropagation* untuk masing-masing pasangan pelatihan :

1. Tiap-tiap unit *output* ( $y_k; k=1,\dots,m$ ) menerima target yang bersesuaian dengan pola input pelatihan, hitung informasi kesalahan (Persamaan 2.12).

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (2.12)$$

kemudian hitung koreksi bobot, digunakan untuk memperbarui  $w_{jk}$  dengan Persamaan 2.13.

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.13)$$

hitung koreksi bias untuk memperbaiki bobot bias ( $w_{0k}$ ) dengan Persamaan 2.14.

$$\Delta w_{0k} = \alpha \delta_k \quad (2.14)$$

kemudian kirimkan nilai  $\delta_k$  ke seluruh unit yang berada pada lapisan dibawahnya (*backward*).

2. Tiap-tiap unit tersembunyi ( $Z_j$ ;  $j=1,\dots,p$ ) menjumlahkan *input delta* dari unit lapisan atasnya.

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (2.15)$$

kalikan fungsi ini dengan turunan fungsi aktivasi untuk menghitung informasi kesalahan (Persamaan 2.16).

$$\delta_j = \delta\_in_j f'(y\_in_j) \quad (2.16)$$

hitung koreksi bobot untuk memperbaiki  $v_{ij}$  (Persamaan 2.17).  $\Delta v_{ij} = \alpha \delta_j x_i$

$$(2.17)$$

hitung koreksi bobot bias untuk memperbaiki  $v_{0j}$  dengan Persamaan 2.18.

$$\Delta v_{0j} = \alpha \delta_j \quad (2.18)$$

c. Perbaiki Bobot dan bias

Tiap-tiap unit *output* ( $y_k, k=1..m$ ) memperbaiki bobot dan bias ( $j = 0, \dots, p$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.19)$$

Tiap-tiap unit tersembunyi ( $Z; 1, \dots, p$ ) memperbaiki bobot dan bias ( $i; 0, \dots, n$ )

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.20)$$

4. Jika iterasi mencapai maksimum maka berhenti.

Keterangan dari simbol :

$t$  = *Output* vektor target,  $t = (t_1, \dots, x_k, \dots, x_m)$

$\delta_k$  = Informasi tentang kesalahan pada unit  $y_k$  yang disebarluaskan kembali ke unit tersembunyi

$\delta_j$  = Informasi tentang kesalahan dari lapisan *output* ke unit tersembunyi  $z_j$

$\alpha$  = *learning rate*

$X_i$  = unit *input* i

$v_{0j}$  = Bobot awal bias pada lapisan tersembunyi j

$v_{ij}$  = Bobot menuju hidden

$Z_j$  = Unit tersembunyi j

$Z_{inj}$  = *Input* jaringan ke  $z_j$

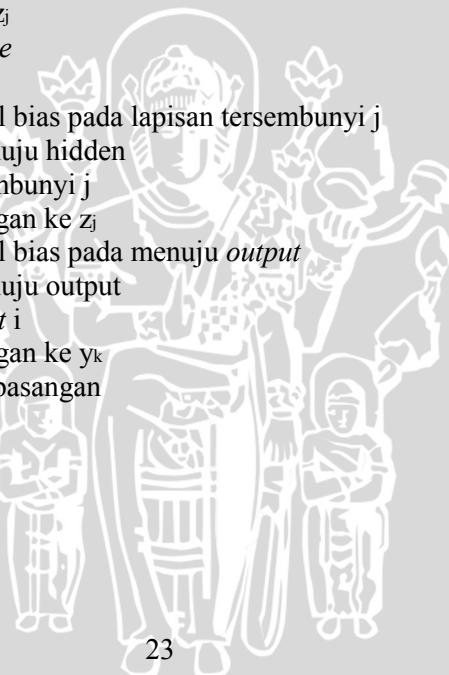
$w_{0k}$  = Bobot awal bias pada menuju *output*

$w_{jk}$  = Bobot menuju output

$y_k$  = Unit *output* i

$Y_{-ink}$  = *Input* jaringan ke  $y_k$

$E$  = Error tiap pasangan



## 2.7 Optimalitas Arsitektur *Backpropagation*

### 2.7.1 Pemilihan Bobot dan Bias

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Bobot yang menghasilkan nilai turunan fungsi aktivasi yang terlalu kecil sebaiknya dihindari, demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil.

Nguyen dan Widrow (1990) dalam Siang,J.J (2005) mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat.

Bila diketahui:  $n$  = jumlah unit masukan

$P$  = jumlah unit tersembunyi

$$\beta = \text{faktor skala} = 0.7\sqrt{n}$$

Algoritma Penentuan Bobot dan Bias:

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

2. Hitung

$$\|v_j\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2} \quad (2.21)$$

3. Bobot yang dipakai sebagai inisialisasi

$$v_{ij} = \frac{\beta v_{ij}(\text{lama})}{\|v_j\|} \quad (2.22)$$

4. Bias yang dipakai sebagai inisialisasi  $v_{j0}$  = bilangan acak antara  $-\beta$  dan  $\beta$ .

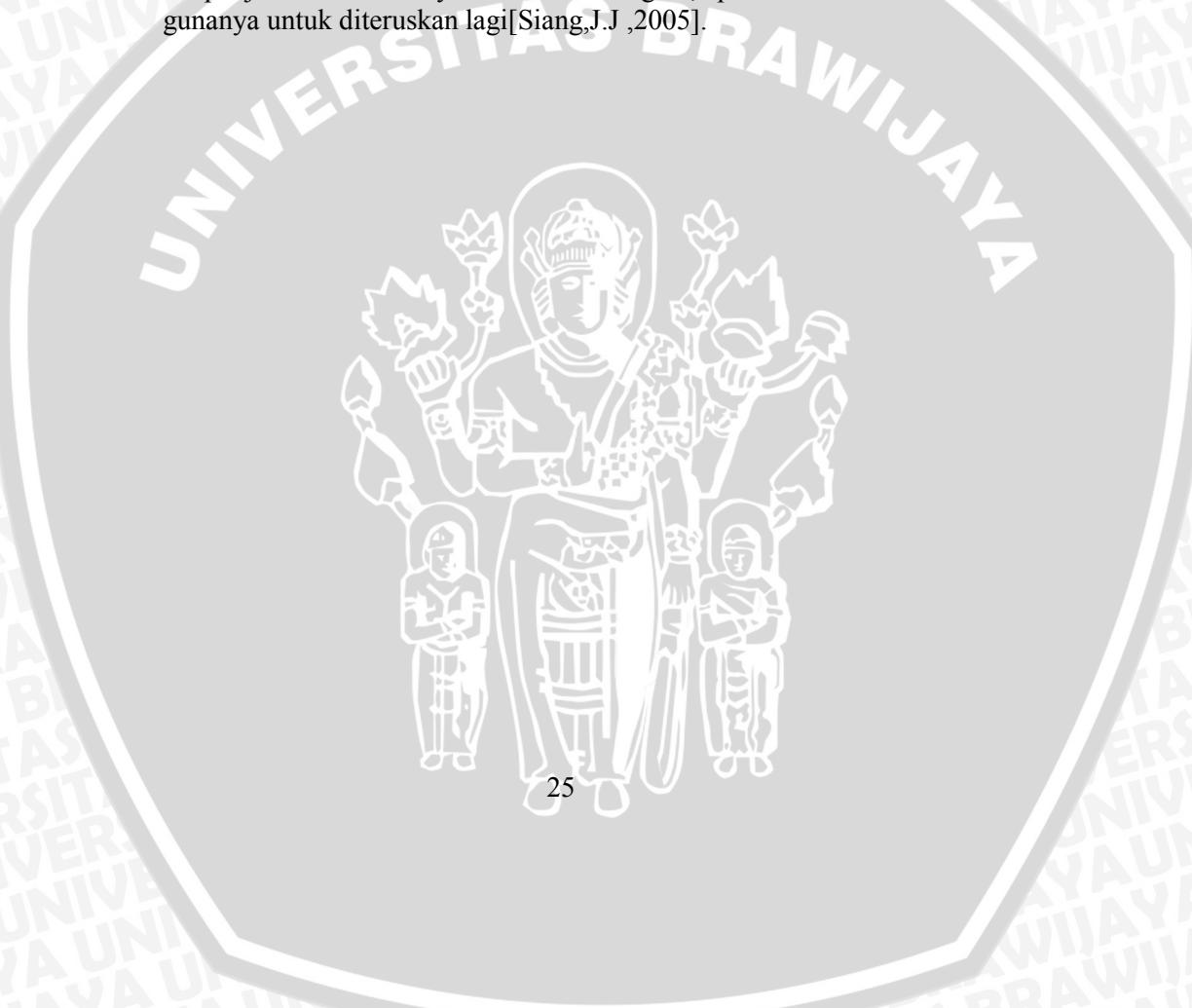
### 2.7.2 Jumlah Unit Tersembunyi

Menentukan jumlah lapis pada *hidden layer* dan menentukan jumlah unit per layernya sangat sulit. Hasil teoritis yang didapat menunjukkan bahwa jaringan dengan sebuah *hidden layer* sudah cukup bagi *backpropagation* untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Tetapi penambahan jumlah *hidden layer* kadangkala membuat pelatihan menjadi lebih mudah (Siang, J.J, 2005).

### 2.7.3 Lama Iterasi

Tujuan utama penggunaan *Backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat.

Umumnya data dibagi menjadi 2 bagian saling asing, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan (misal setiap 10 epoch), kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya sudah meningkat, pelatihan tidak ada gunanya untuk diteruskan lagi [Siang,J.J ,2005].



UNIVERSITAS BRAWIJAYA



26

### BAB III

## METODOLOGI DAN PERANCANGAN SISTEM

Dalam bab ini akan dijelaskan mengenai metode penelitian dan perancangan sistem untuk pendekripsi kanker payudara dengan *mammogram* dengan menggunakan metode jaringan syaraf tiruan (JST) *backpropagation*. Dalam menyelesaikan permasalahannya memiliki langkah-langkah sebagai berikut:

#### 3.1 Analisa Data

Dalam skripsi ini digunakan foto jaringan payudara kanan dan kiri yang di foto dengan posisi *Cranio Caudal(CC)* dan *mediolateral oblique(MLO)*. Data yang digunakan dalam penelitian ini ada 2 macam, yaitu terdiri dari:

- a. Data 1 dari *DDSM database of the USF (University of South Florida)*. Data ini tersedia homepages :

[<http://marathon.csee.usf.edu/Mammography/Database.html>]

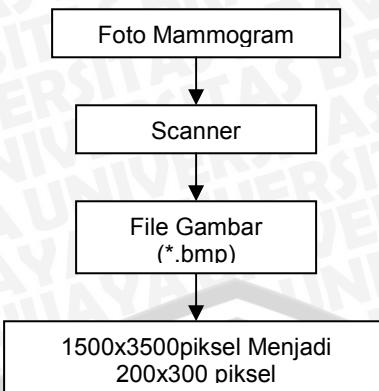
- 42 data dengan payudara kanan.
- 36 data dengan payudara kiri.

- b. Data 2 dari Rumah Sakit Onkologi Surabaya.

- 8 data pasien dengan payudara normal.
- 12 data pasien dengan payudara kanker.

Seluruh data yang di dapat sebelumnya sudah *divalidasi* menurut jenisnya, yaitu payudara normal dan payudara terkena kanker. Dari data yang telah *divalidasi* nantinya akan dikelompokkan menjadi 2 bagian, yaitu kelompok data *training* dan data *recognize*.

Untuk data hasil keluaran *mammografi* yang berasal dari Rumah Sakit akan di proses dahulu seperti pada Gambar 3.1, sehingga data akan disimpan pada file berekstensi \*.bmp. Hasil keluaran *mammografi* yang berupa foto *mammogram* di scan menggunakan Canon Scan 300 ex dengan spesifikasi 300 dpi. Scanner dijalankan dengan bantuan sistem operasi windows, kemudian disimpan di dalam hardisk dengan format file .bmp. Dengan menggunakan *Microsoft Office Picture Manager*, foto yang semula berukuran 1500 x 3500 piksel akan diubah ukurannya menjadi 200 x 300 piksel agar tampilan foto pada software tidak terlalu besar.



Gambar 3.1 Proses Data 2

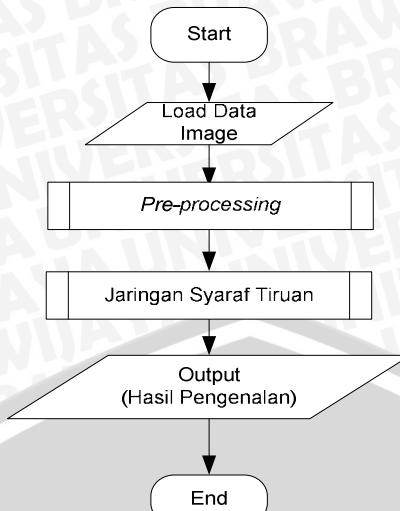
Setelah didapat file gambar yang diinginkan maka semua data baik data dari rumah sakit ataupun dari internet di kelompokkan sesuai dengan jenisnya. Setiap data juga haruslah melalui tahapan-tahapan terlebih dahulu sehingga bisa menjadi *inputan* yang baik dalam JST. Adapun *inputan* yang dapat diterima dengan baik oleh JST adalah berupa kumpulan data numerik [Nugraha, 2002]. Dengan demikian, bagaimana mengkonversikan suatu citra digital menjadi kumpulan data numerik yang konsisten.

### 3.2 Analisa Sistem

Sistem yang dibangun merupakan implementasi aplikasi Jaringan Syaraf Tiruan untuk mengenali gambar *mammogram*. Pada sistem ini akan digunakan 2 macam data yang berbeda.

Secara umum sistem ini memiliki 2 bagian pemrosesan. Proses pertama pengolahan *image* yang menerima *inputan* sistem. Proses kedua adalah proses pelatihan dan pengenalan gambar *mammogram* dengan Jaringan Syaraf tiruan.

Proses pengolahan *image* terdapat proses *preprocessing* gambar *mammogram*. Pada proses *preprocessing*, gambar dengan ukuran 200x300 piksel akan dijadikan matrik yang berukuran lebih kecil. Sistem secara umum digambarkan seperti pada gambar 3.2



Gambar 3.2 Diagram alur sistem secara umum

### 3.2.1 Pengolahan Citra (*Pre processing*)

Proses *pre-processing* merupakan proses mengolah citra sehingga menghasilkan citra yang sesuai dengan kebutuhan. Pengambilan data berupa gambar *mammogram* yang tersimpan dalam file gambar berekstensi file bitmap. Citra dengan ukuran 200x300 piksel akan dibagi menjadi beberapa blok (menjadi beberapa sampling) dan nilai setiap bagian blok diwakili oleh nilai karakteristik dari nilai *mean* maupun *variance*. Proses *preprocessing* secara umum di tunjukkan pada Gambar 3.3.

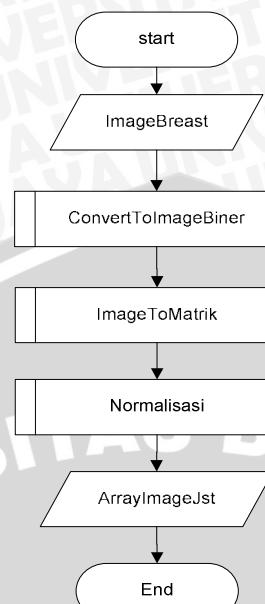
#### 1. Load Data

Dalam proses ini citra akan dimasukkan ke dalam sistem. Untuk memasukkan citra ke dalam sistem yaitu, dengan membuka *file* citra hasil *scan* atau hasil *download* dari intenet yang sudah ada pada komputer.

#### 2. Proses *ConvertToImageToBiner*

Setelah citra diload, kemudian dilakukan proses *ConvertToImagebiner*, yaitu proses perbandingan antara nilai *grayscale* setiap *pixel* dengan nilai *threshold* 220. Penggunaan nilai 220 ini berdasarkan beberapa percobaan. Jika nilai *grayscale* lebih besar atau sama dengan nilai *threshold*, *pixel* tersebut diberi

nilai 1 (warna putih), sedangkan jika nilai *grayscale* kurang dari nilai *threshold*, *pixel* tersebut diberi nilai 0 (warna hitam). Flowchart *ConvertToImageBiner* ditunjukkan pada Gambar 3.4.

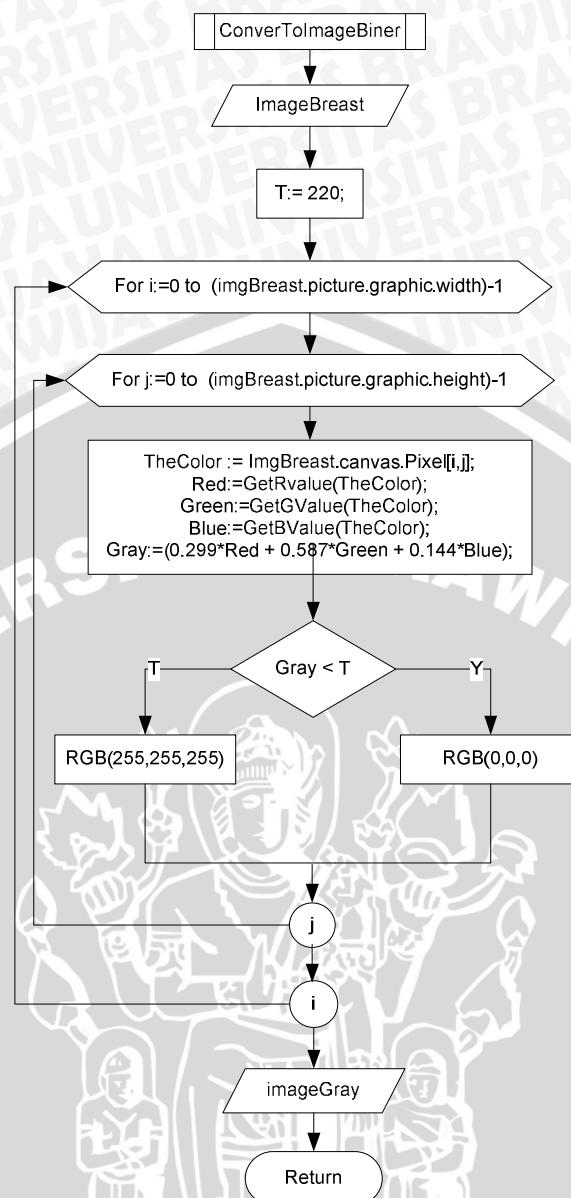


Gambar 3.3 Proses Preprocessing

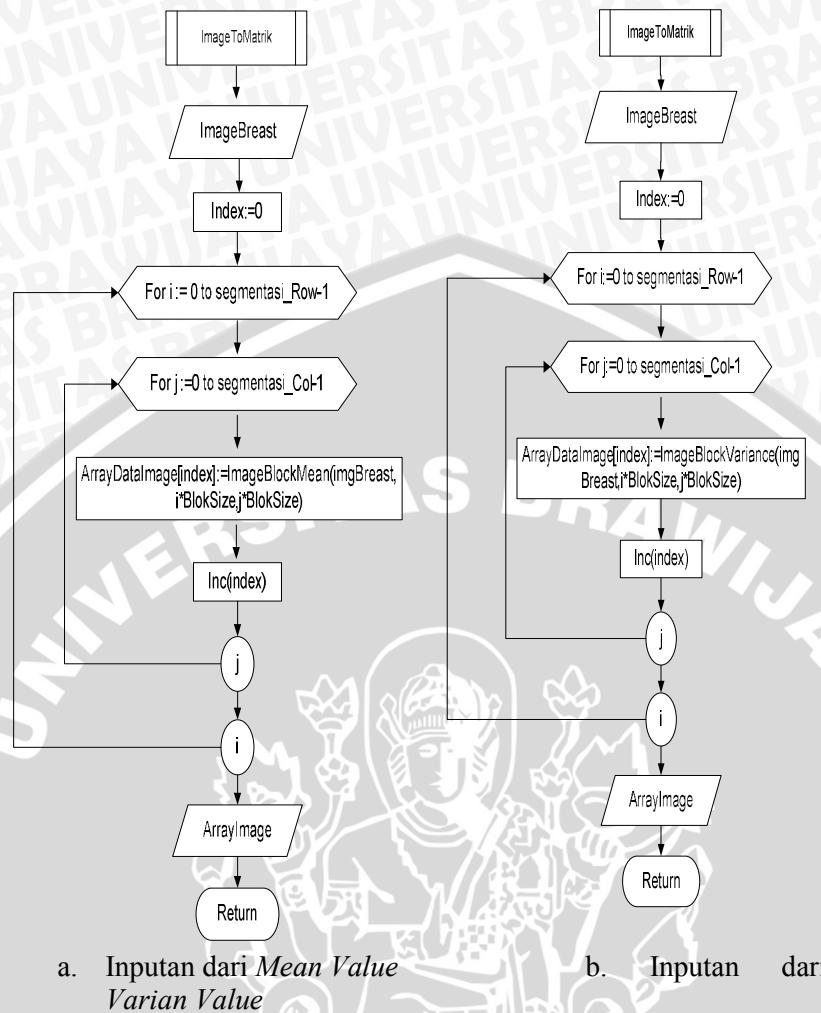
### 3. Proses *ImageToMatrik*

Sebelum proses mengubah citra ke dalam bentuk matrik, dilakukan beberapa proses *resampling* yaitu proses untuk membagi gambar yang berukuran 200x300 menjadi beberapa blok. Dari gambar 200x300 akan dibagi menjadi blok berukuran NxN dimana nilai N dilakukan beberapa percobaan. Nilai tiap blok akan dihitung sesuai nilai karakteristik data input yang digunakan. Dari nilai tersebut akan mewakili nilai tiap blok sehingga keseluruhan nilai tiap blok akan membentuk matrik  $(200/N) \times (300/N)$ .

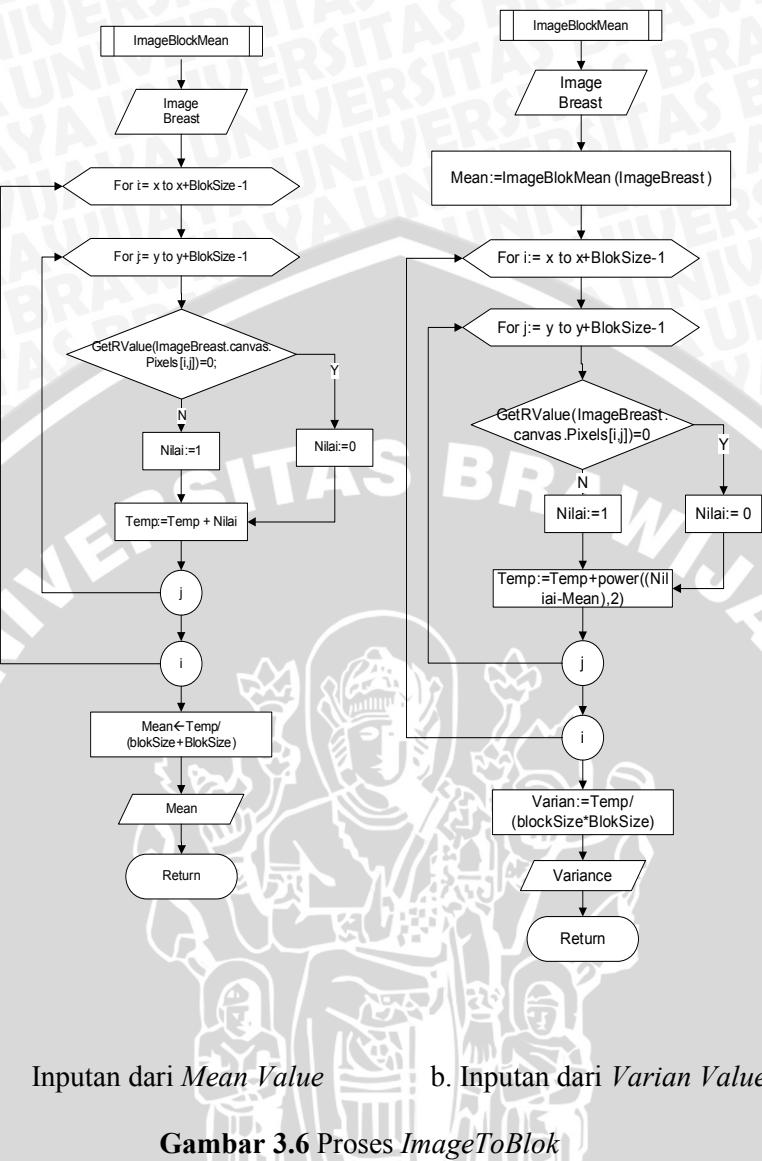
Pada proses *preprocessing* citra disini menggunakan 3 karakteristik data *input* yang berbeda, yaitu *mean*, *variance* dan perbandingan *mean* dan *variance*.



Gambar 3.4 Proses `ConverToImageBiner`



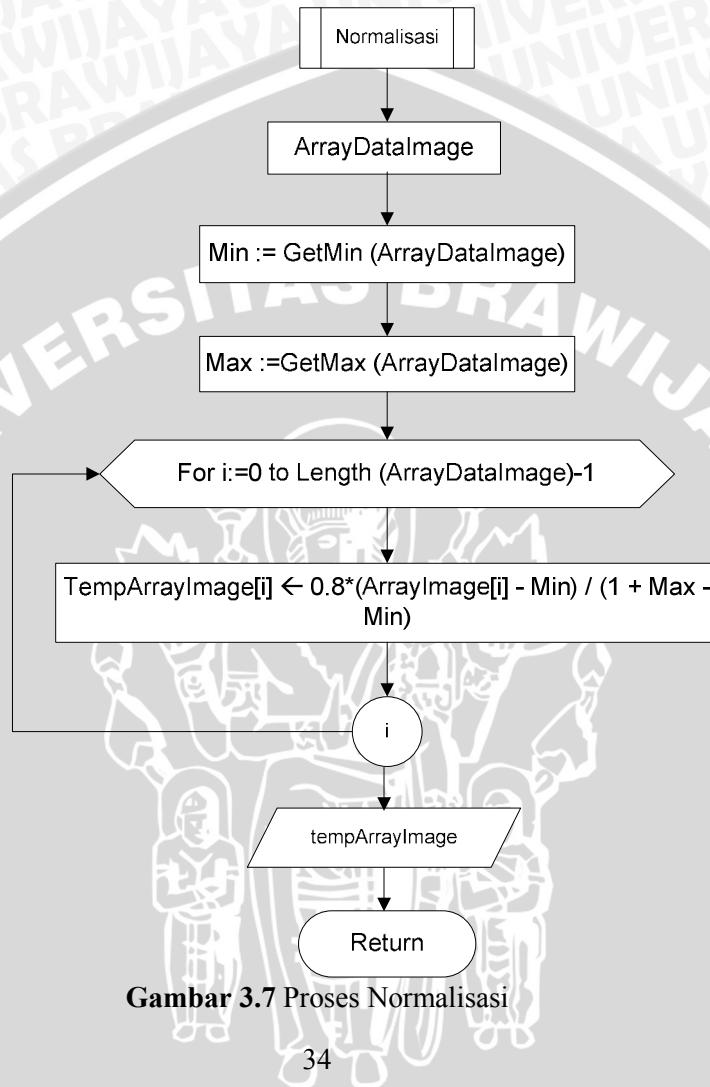
**Gambar 3.5 Proses *ImageToMatrik***



Gambar 3.6 Proses ImageToBlok

#### 4. Normalisasi

Setelah proses *preprocessing* dilakukan, maka langkah selanjutnya dilakukan proses normalisasi, yaitu mengkonversi nilai intensitas *grayscale* gambar sehingga memiliki range 0,1 sampai 0,9. Dari hasil normalisasi nantinya akan dijadikan *inputan* ke dalam Jaringan Syarat Tiruan untuk proses pelatihan dan pengenalan. Proses ini ditunjukkan pada Gambar 3.7.



**Gambar 3.7** Proses Normalisasi

### 3.2.2 Proses Jaringan Syaraf Tiruan

Proses jaringan syaraf tiruan dilakukan setelah proses pengolahan citra selesai dilakukan. Proses ini dapat dibagi menjadi proses membuat struktur jaringan, proses pelatihan, dan proses pengenalan.

#### 3.2.2.1. Struktur Jaringan Syaraf Tiruan

Jaringan yang akan dibangun meliputi tiga buah *layer*, yaitu *input layer*, satu *hidden layer*, dan *output layer*. *Input layer* terdiri dari  $M \times N$  buah neuron yang merupakan hasil pada saat normalisasi. Misalkan nilai  $M \times N$  adalah  $20 \times 30$ , maka terdapat 600 buah neuron pada *input layer*. Dan untuk *output* terdapat 1 *output layer*, yaitu untuk mengenali kanker dan normal. Sedangkan jumlah neuron pada *hidden layer* adalah  $n$ , dimana nilai  $n$  nantinya akan diujicobakan beberapa nilai. Hal ini dilakukan untuk mendapatkan arsitektur jaringan yang sesuai untuk melakukan pengenalan.



Gambar 3.8 Arsitektur JST Pendekstian Kanker Payudara

### 3.2.2.2. Proses Pelatihan

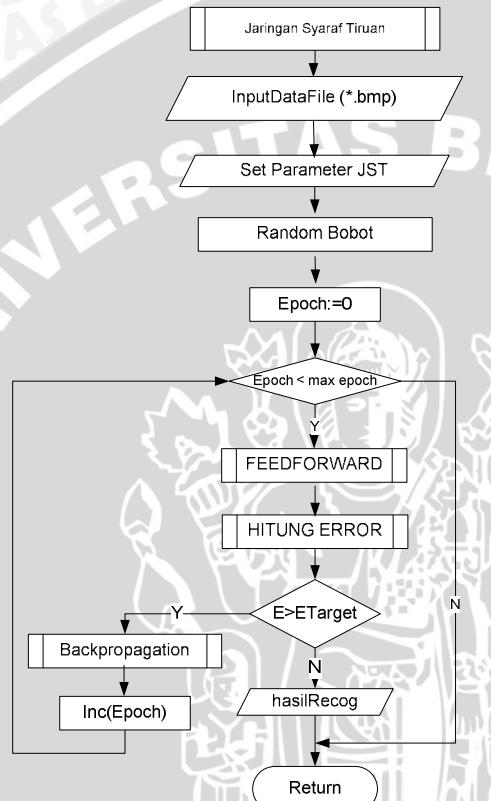
Algoritma pelatihan *backpropagation* pada dasarnya terbagi menjadi 3 langkah, yaitu: langkah maju (*feedforward*), propagasi balik (*backpropagation*) dan perubahan bobot. Secara keseluruhan langkah-langkah pelatihan pada pendekripsi kanker payudara menggunakan algoritma *backpropagation* diilustrasikan pada Gambar 3.9 dan langkah-langkahnya adalah sebagai berikut:

1. Mulai.
2. Masukkan data *input* yang akan dilatihkan. Set parameter JST yaitu kesalahan yang ditargetkan, iterasi maksimal, *neuron hidden*, *learning rate* dan *epoch*.
3. Inisialisasi bobot awal secara random, bobot secara random berkisar antara -0,05 sampai 0,05.
4. Setting parameter iterasi awal (*epoch*) sama dengan 0.
5. Selama *epoch* kurang dari maksimal iterasi lakukan langkah 6, jika tidak lompat ke langkah 11.
6. Setelah seluruh koneksi jaringan terisi bobot, lakukan *feedforward*.
7. Lakukan perhitungan kesalahan antara pola *output* JST dan pola target.
8. Periksa apakah kesalahan *output* lebih besar dari kesalahan yang ditargetkan? Jika ya, maka lakukan langkah 9, jika tidak, maka lakukan langkah 10.
9. Lakukan langkah *backpropagation*.
10. Proses berhenti, bobot akhir pelatihan disimpan pada dataset, data siap digunakan untuk proses pengenalan.
11. Selesai.

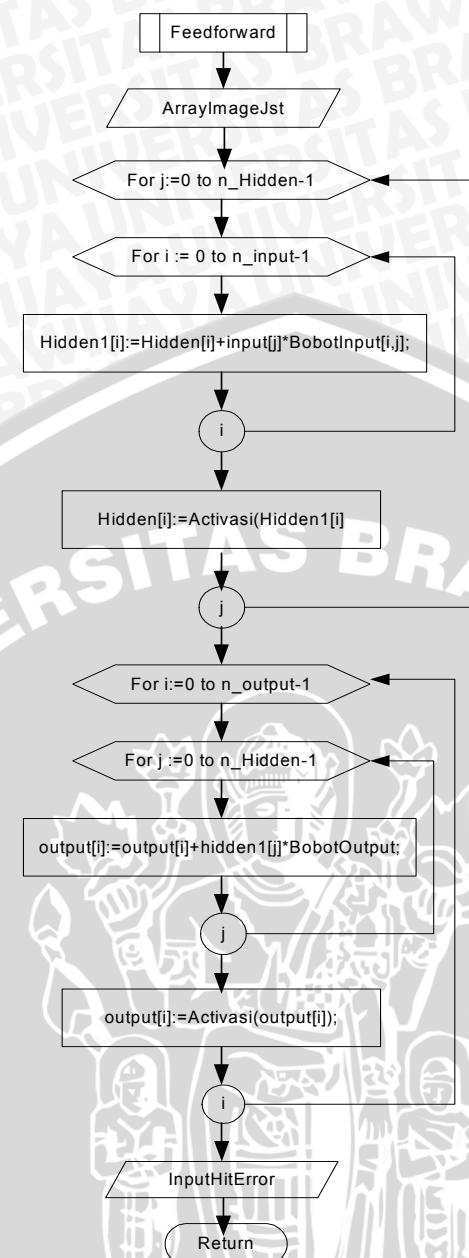
Pada proses pelatihan JST terdapat sub proses *feedforward* dan *backpropagation*. Untuk proses *feedforward* dapat dilihat pada Gambar 3.10 dan langkah-langkah *feedforward* adalah sebagai berikut:

1. Mulai
2. Kalikan seluruh data *input* pada *neuron\_input* dengan bobot random pada masing-masing bobot koneksi bobot *Input* yang terhubung dengan *neuron input*. Kemudian jumlahkan seluruh vektor bobot yang menuju neuron *hidden* yang sama.

3. Lakukan aktivasi hasil penjumlahan tersebut pada masing-masing *neuron*, sehingga *output* pada lapis ini berada pada kisaran 0 dan 1.
4. Kalikan seluruh data hasil aktivasi masing-masing neuron lapis *hidden* pada *neuron\_hidden* dengan bobot pada masing-masing koneksi bobot *Output* yang terhubung dengan *neuron* pada lapis *hidden*. Kemudian jumlahkan seluruh vektor bobot yang menuju *neuron output* yang sama.
5. Lakukan aktivasi hasil penjumlahan tersebut pada masing-masing neuron di lapisan *output*, sehingga *output* pada lapis ini berada pada kisaran 0 dan 1.
6. Selesai.



Gambar 3.9 Diagram Alir Proses pembelajaran JST



Gambar 3.10 Diagram Alir Proses *Feedforward*

Untuk menghitung kesalahan pada proses *feedforward*, maka dilakukan proses *hitungerror*. Fungsi untuk menghitung *error* sebagai berikut :

```
function Error(a : Real) : Real;  
begin  
Result := RoundTo(Power((target - a), 2)*0.5,-5);  
end;
```

Sedangkan untuk langkah *backpropagation* adalah perhitungan informasi kesalahan pada tiap *neuron* pada masing-masing lapisan dimulai dari kesalahan pada lapis *output* hingga lapis *hidden* terdekat dengan lapis *input*. Informasi kesalahan berguna untuk menghitung faktor peubah bobot yang akan digunakan untuk perbaikan bobot lama. Algoritma *backpropagation* diperlihatkan pada diagram alir pada Gambar 3.11

Langkah-langkah *backpropagation*:

1. Mulai
2. Pada lapisan *output*. Pertama hitung selisih antara target pelatihan dengan *output*. Kalikan selisih ini dengan *output* yang telah diaktifasi dengan fungsi turunan aktifasi. Hasil perkalian ini merupakan faktor kesalahan pada lapis *output* ini akan digunakan untuk menghitung faktor kesalahan pada lapisan *hidden* dan untuk menghitung faktor peubah bobot pada vektor bobot menuju *output*.
3. Hitung besar faktor peubah bobot baru pada tiap synapsis lapisan *output*.
4. Perbaiki bobot lama pada tiap synapsis menuju *output* dengan menjumlahkan bobot lama *bobotOutput* dengan peubah bobot yang telah dihitung pada langkah 3.
5. Pada lapisan *hidden*. Untuk menghitung faktor kesalahan masing-masing neuron lapisan *hidden* dilakukan. Masing-masing faktor kesalahan di *output* kalikan dengan bobot lama yang terkoneksi dengan neuron lapisan *output*. Kemudian hasil perkalian pada seluruh koneksi yang terhubung dengan masing-masing neuron pada lapis *hidden* akan dijumlahkan. Faktor kesalahan pada neuron lapis *hidden* akan digunakan untuk menghitung peubah bobot pada koneksi dari lapisan *input* menuju lapisan *hidden*.

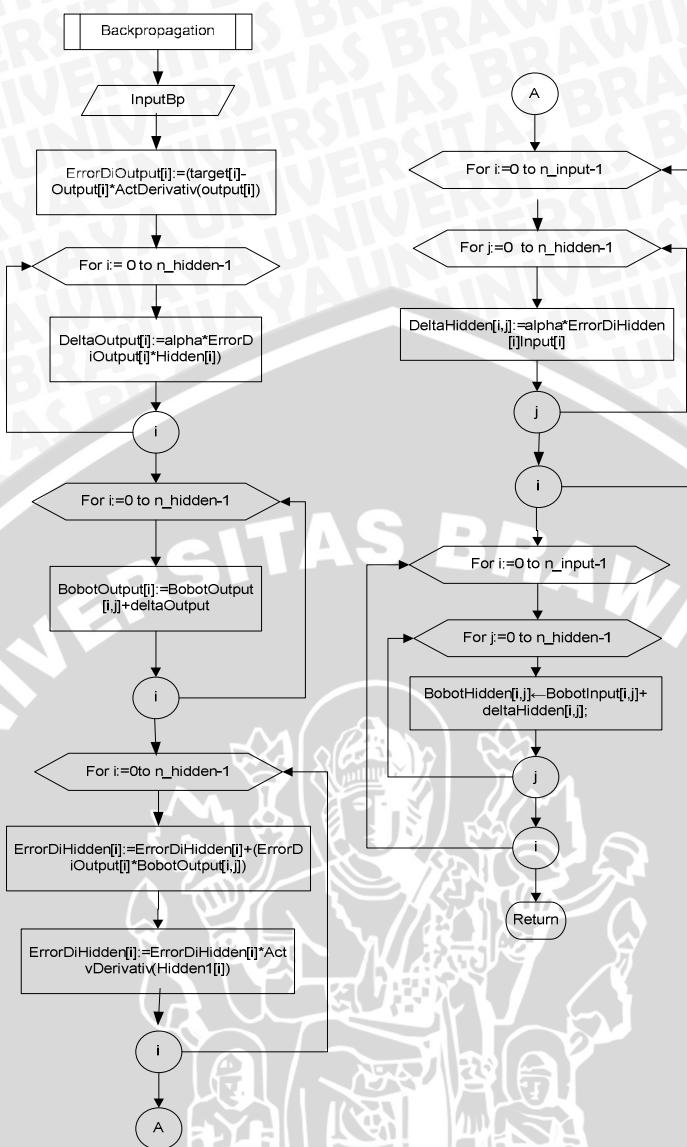
6. Hitung besar faktor peubah bobot baru pada tiap koneksi pada lapisan *hidden*:
7. Perbaiki bobot lama pada tiap koneksi dari lapisan *input* menuju lapisan *hidden* dengan menjumlahkan bobot lama bobot *input* dengan peubah bobot yang telah dihitung pada langkah 3.
8. Selesai. Bobot baru akan digunakan untuk proses *feedforward* kedua dan seterusnya hingga bobot optimal didapatkan.

### 3.2.2.3. Proses Pengenalan

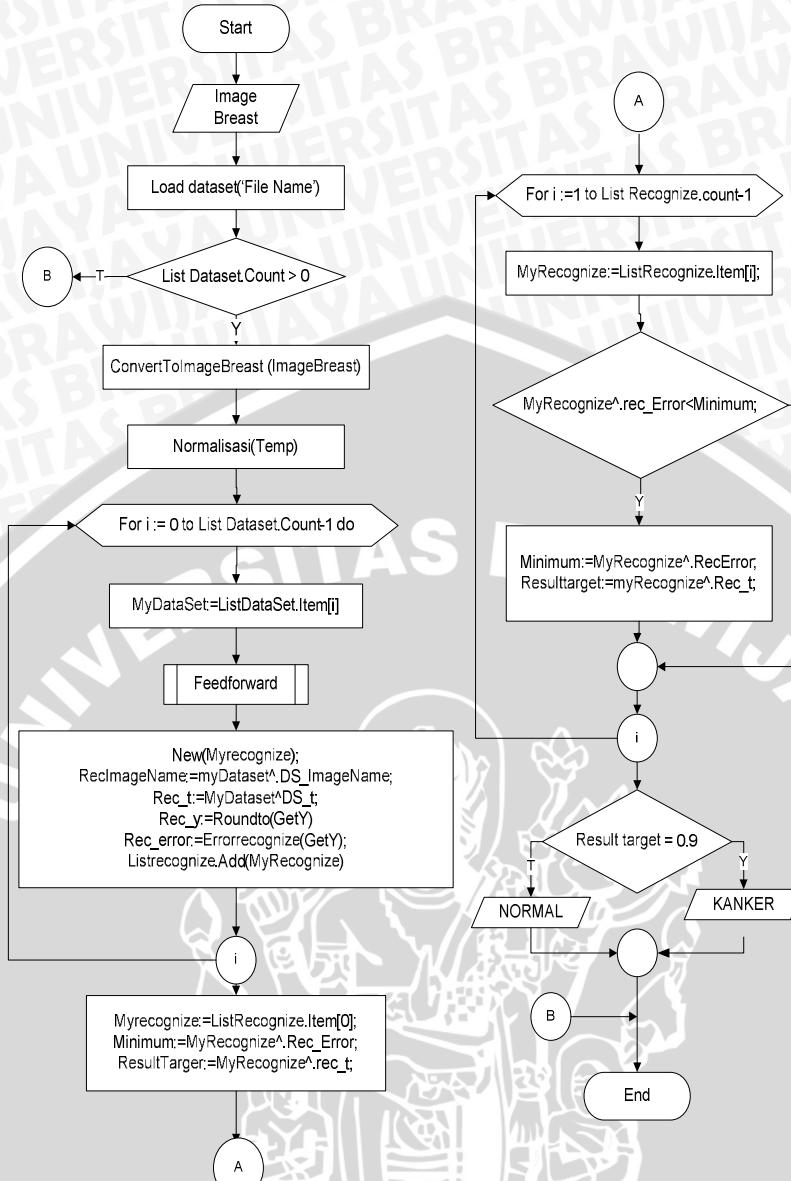
Setelah proses pelatihan selesai dilakukan, maka proses pengenalan dapat dilakukan. Pada proses pengenalan, konfigurasi bobot tiap data pada *dataset* akan diuji dalam proses pengenalan. Setiap bobot data pada *dataset* akan dicobakan pada pola data masukan dengan *feedforward* atau dicocokkan satu persatu dengan tiap data pada dataset. Pencocokan pola dideteksi dengan pemilihan kesalahan atau jaraknya. Sistem akan memutuskan hasil pengenalannya pada data dengan jarak target-output terkecil. Jika kesalahan terkecil terjadi pada *dataset* data ke i maka data akan dikenali dengan data ke i.

Langkah-langkah proses pengenalan secara keseluruhan adalah sebagai berikut:

1. Mulai
2. Input data image *mammogram*
3. load data training pada dataset. Cek apakah ada dataset yang tersimpan. Jika Ya lakukan langkah 4, jika tidak selesai.
4. Lakukan convert Image to biner kemudian di normalisasi.
5. Lakukan *feedforward* untuk mendapatkan *output* pengenalan.
6. hitung kesalahan antara target dengan *output* pengenalan.
7. Apakah jarak output dan target minimum? Jika minimum apakah target bernilai 0.9? Jika YA, lakukan langkah 9. Jika tidak lakukan langkah 10
8. Data diidentifikasi kanker.
9. Data diidentifikasi kena normal.
10. Selesai.



Gambar 3.11 Diagram alir proses *Backpropagation*



Gambar 3.12 Diagram alir proses pengenalan

### 3.3 Perancangan Uji Coba

#### 3.3.1 Citra uji

- Spesifikasi data yang akan diujikan yaitu:
1. data berupa gambar *mammogram* berekstensi *.bmp*.

2. Ukuran  $200 \times 300$  piksel

3. Sumber data berasal dari :

- <http://marathon.csee.usf.edu/Mammography/Database.html>
- Dari RS. Onkologi

#### 3.3.2 Pengujian pengenalan kanker payudara

Tahap terpenting dalam proses pengenalan adalah mendapatkan struktur jaringan saraf yang efisien. Hal yang perlu ditentukan dalam pembentukan jaringan syaraf yang efisien adalah nilai *learning rate*, jumlah *neuron hidden* (tersembunyi), *epoch* dan target kesalahan. Selain pengaruh parameter pada jaringan syaraf tiruan, dilakukan juga pengujian terhadap karakteristik data input dan pembentukan sampling (blok).

##### - Pengaruh Ukuran Blok data input

**Tabel 3.1** Pengaruh Ukuran Blok data *input*

Data	Ukuran Blok	Hasil

##### - Pengaruh Jumlah Neuron

Untuk mengetahui pengaruh neuron terhadap banyaknya iterasi, dilakukan pengujian dengan menentukan jumlah *neuron* yang diujikan. Jumlah iterasi maksimal ditetapkan 5000 iterasi. Bentuk dari hasil percobaan tersebut akan disajikan sebagai berikut:

**Tabel 3.2** Pengaruh jumlah neuron terhadap waktu iterasi

Jumlah <i>neuron</i>	Error minimal	Waktu

##### - Pengaruh Learning Rate

**Tabel 3.3** Pengaruh *Learning rate* pada kesalahan minimal

Learning rate	Epoch ke-

Dari jaringan yang terbentuk nantinya akan dilakukan proses pelatihan dengan pemilihan pola *input* yang berbeda. Hasil pelatihan nantinya akan menunjukkan bahwa nilai *output* mendekati dengan target.

**Tabel 3.4** Hasil pengujian

Jenis Data	Hasil Pengenalan	Keakuratan

### 3.4 Contoh Perhitungan

Pada subbab ini tidak menggunakan perhitungan asli secara manual, namun memakai ukuran sederhana sehingga mudah dalam memahami maksud dari sistem kerja perangkat lunak ini. Dalam perhitungan manual ini menggunakan contoh dari gambar berukuran 200x300 yang dibagi menjadi 50x50 blok. Tiap blok dihitung nilai rata-ratanya sehingga menjadi matrik berukuran 4x6, yang nilainya mewakili dari nilai *mean* dari tiap blok.

- **Pola data 1**

#### Iterasi 1

**Tabel 3.5** Data input matrik 4 x 6

	1	2	3	4
1	104.56	138.61	138.51	135.61
2	171.95	235.16	180.68	161.67
3	135.81	131.67	157.71	156.23
4	114.71	15.99	45.5	59.58
5	12.89	1.07	0	0
6	0	0	0	0

Selanjutnya data inputan berupa matrik berukuran 4x6 diubah menjadi array satu dimensi, kemudian dinormalisasi pada interval [0.1, 0.9] karena fungsi aktivasi yang digunakan adalah fungsi sigmoid, dimana fungsi ini merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 atau 1. Proses normalisasi dihitung menggunakan rumus pada persamaan 2.13.

**Tabel 3.6** Hasil Normalisasi

x1'	x2'	x3'	x4'	x5'	x6'	x7'	x8'	x9'	x10'	x11'	x12'
0.46	0.57	0.57	0.56	0.68	0.9	0.71	0.65	0.56	0.55	0.64	0.63
x13'	x14'	x15'	x16'	x17'	x18'	x19'	x20'	x21'	x22'	x23'	x24'
0.49	0.15	0.25	0.3	0.14	0.1	0.1	0.1	0.1	0.1	0.1	0.1

**Inisialisasi****Tabel 3.7** Bobot awal ke lapisan hidden ( $v_{ij}$ )

	Z1	Z2	Z3	Z4	Z5
X1	-0.01	-0.01	-0.03	-0.03	-0.03
X2	0.03	-0.04	-0.05	-0.03	0.02
X3	0.01	-0.02	-0.04	-0.04	0.02
X4	0.05	0.01	0.02	-0.02	0.00
X5	0.00	0.03	0.02	0.01	0.02
X6	0.03	-0.02	0.00	0.04	0.04
X7	-0.02	0.00	0.04	-0.04	0.04
X8	-0.01	0.01	0.03	0.02	0.04
X9	-0.02	0.00	0.00	0.02	0.02
X10	0.00	-0.01	0.03	0.01	-0.04
X11	0.03	0.00	0.03	0.05	-0.02
X12	-0.04	0.00	-0.02	-0.01	-0.01
X13	-0.01	-0.01	-0.01	0.02	0.05
X14	0.01	-0.03	0.00	0.02	0.05
X15	0.01	0.04	0.05	-0.01	0.02
X16	-0.04	-0.02	-0.02	-0.02	-0.01
X17	0.03	0.04	-0.02	-0.02	0.00
X18	-0.01	-0.03	-0.05	-0.03	-0.03
X19	-0.03	0.03	-0.04	0.04	-0.05
X20	0.04	0.00	-0.03	0.02	-0.05
X21	0.02	-0.02	-0.04	-0.01	-0.02
X22	0.01	-0.02	0.01	-0.03	0.01
X23	0.04	0.00	-0.01	0.03	0.02
X24	0.05	0.04	0.00	0.05	0.01

$$V_o = 0.005$$

**Tabel 3.8** Bobot awal ke lapisan *output* ( $w_{jk}$ )

	y
Z1	0.45
Z2	0.32
Z3	-0.40
Z4	0.46
Z5	0.27

$$W_0 = 0.001$$

**Langkah 5 :** Hitung keluaran unit *hidden* ( $Z_j$ )

Masing-masing unit *hidden* menjumlahkan bobot sinyalnya, pada langkah ini menggunakan persamaan 2.15, hasilnya ditunjukkan pada Tabel 3.9.

**Tabel 3.9** Operasi pada *hidden*

J	Zin
1	0.0435
2	-0.0246
3	0.0185
4	0.0184
5	0.1110

Menghitung aktivasi  $z_{in}$  menggunakan persamaan 2.16 dan hasilnya pada tabel 3.10.

**Tabel 3.10** Hasil Aktivasi Operasi *hidden*

j	Z
1	0.0497
2	-0.0281
3	0.0212
4	0.0210
5	0.1267

**Langkah 6:** Hitung keluaran unit  $Y_k$

Masing-masing unit *output* menjumlahkan bobot sinyal *input*. Untuk menghitung nilai keluaran menggunakan persamaan 2.18.

$$y_{in} = 0.0327$$

Hasil aktivasinya :  $y = 0.049747$

**Langkah 7:** Hitung faktor  $\delta$  di unit output

Setiap unit *output* menerima vektor hasil yang diinginkan ( $t$ ) untuk data masukan tersebut.

$$\delta = (t-y) * f'(y)) = 0.975715$$

Hitung nilai koreksi bobotnya dengan  $\alpha=0.01$  dengan menggunakan persamaan 2.20 dan hasilnya pada tabel 3.11.

**Tabel 3.11** Selisih Bobot Wjk

	$\Delta W$
1	0.0005
2	-0.0003
3	0.0002
4	0.0002
5	0.0012

**Langkah 8:** Hitung kesalahan di unit *hidden*

Setiap unit *hidden* menjumlahkan delta masukannya (dari unit-unit pada lapisan diatasnya). Dihitung dengan persamaan 2.22

**Tabel 3.12** Faktor Penimbang di unit *hidden*

j	$\Delta \delta_{in}$
1	0.439072
2	0.312229
3	-0.39029
4	0.448829
5	0.263443

Faktor kesalahan  $\delta$  di unit *hidden***Tabel 3.13** Aktivasi faktor Penimbang di unit *hidden*

J	$\delta$
1	0.546547
2	0.372869
3	-0.47737
4	0.560787
5	0.310753

Suku perubahan bobot ke unit *hidden* dihitung dengan persamaan 2.24.

**Tabel 3.14** Perbaikan bobot penimbang  $\Delta V_{ij}$ 

V <sub>ij</sub>	1	2	3	4	5
1	0.002514	0.001715	-0.0022	0.00258	0.001429
2	0.003115	0.002125	-0.00272	0.003196	0.001771
3	0.003115	0.002125	-0.00272	0.003196	0.001771
4	0.003061	0.002088	-0.00267	0.00314	0.00174
5	0.003717	0.002536	-0.00325	0.003813	0.002113
6	0.004919	0.003356	-0.0043	0.005047	0.002797
7	0.00388	0.002647	-0.00339	0.003982	0.002206
8	0.003553	0.002424	-0.0031	0.003645	0.00202
9	0.003061	0.002088	-0.00267	0.00314	0.00174
10	0.003006	0.002051	-0.00263	0.003084	0.001709
11	0.003498	0.002386	-0.00306	0.003589	0.001989
12	0.003443	0.002349	-0.00301	0.003533	0.001958
13	0.002678	0.001827	-0.00234	0.002748	0.001523
14	0.00082	0.000559	-0.00072	0.000841	0.000466
15	0.001366	0.000932	-0.00119	0.001402	0.000777
16	0.00164	0.001119	-0.00143	0.001682	0.000932
17	0.000765	0.000522	-0.00067	0.000785	0.000435
18	0.000547	0.000373	-0.00048	0.000561	0.000311
19	0.000547	0.000373	-0.00048	0.000561	0.000311
20	0.000547	0.000373	-0.00048	0.000561	0.000311
21	0.000547	0.000373	-0.00048	0.000561	0.000311
22	0.000547	0.000373	-0.00048	0.000561	0.000311
23	0.000547	0.000373	-0.00048	0.000561	0.000311
24	0.000547	0.000373	-0.00048	0.000561	0.000311

**Langkah 9 :** hitung semua perubahan bobot di unit *output*

Setelah semua pola dihitung, lakukan perubahan bobot (dilakukan pada setiap iterasi) yaitu:

**Table 3.15** Bobot baru *hidden-output layer*

j	W <sub>k</sub>
1	0.4495
2	0.3203
3	-0.4002
4	0.4598
5	0.2688

Perubahan bobot unit *hidden*

Table 3.16 Bobot baru *input-hidden layer*

	Z1	Z2	Z3	Z4	Z5
1	-0.0125	-0.0117	-0.0278	-0.0326	-0.0314
2	0.0269	-0.0421	-0.0473	-0.0332	0.0182
3	0.0069	-0.0221	-0.0373	-0.0432	0.0182
4	0.0469	0.0079	0.0227	-0.0231	-0.0017
5	-0.0037	0.0275	0.0232	0.0062	0.0179
6	0.0251	-0.0234	0.0043	0.0350	0.0372
7	-0.0239	-0.0026	0.0434	-0.0440	0.0378
8	-0.0136	0.0076	0.0331	0.0164	0.0380
9	-0.0231	-0.0021	0.0027	0.0169	0.0183
10	-0.0030	-0.0121	0.0326	0.0069	-0.0417
11	0.0265	-0.0024	0.0331	0.0464	-0.0220
12	-0.0434	-0.0023	-0.0170	-0.0135	-0.0120
13	-0.0127	-0.0118	-0.0077	0.0173	0.0485
14	0.0092	-0.0306	0.0007	0.0192	0.0495
15	0.0086	0.0391	0.0512	-0.0114	0.0192
16	-0.0416	-0.0211	-0.0186	-0.0217	-0.0109
17	0.0292	0.0395	-0.0193	-0.0208	-0.0004
18	-0.0105	-0.0304	-0.0495	-0.0306	-0.0303
19	-0.0305	0.0296	-0.0395	0.0394	-0.0503
20	0.0395	-0.0004	-0.0295	0.0194	-0.0503
21	0.0195	-0.0204	-0.0395	-0.0106	-0.0203
22	0.0095	-0.0204	0.0105	-0.0306	0.0097
23	0.0395	-0.0004	-0.0095	0.0294	0.0197
24	0.0495	0.0396	0.0005	0.0494	0.0097

**Itersai 2****Tabel 3.17** Bobot awal ke lapisan *hidden* ( $v_{ij}$ )

	Z1	Z2	Z3	Z4	Z5
1	-0.0125	-0.0117	-0.0278	-0.0326	-0.0314
2	0.0269	-0.0421	-0.0473	-0.0332	0.0182
3	0.0069	-0.0221	-0.0373	-0.0432	0.0182
4	0.0469	0.0079	0.0227	-0.0231	-0.0017
5	-0.0037	0.0275	0.0232	0.0062	0.0179
6	0.0251	-0.0234	0.0043	0.0350	0.0372
7	-0.0239	-0.0026	0.0434	-0.0440	0.0378
8	-0.0136	0.0076	0.0331	0.0164	0.0380
9	-0.0231	-0.0021	0.0027	0.0169	0.0183
10	-0.0030	-0.0121	0.0326	0.0069	-0.0417
11	0.0265	-0.0024	0.0331	0.0464	-0.0220
12	-0.0434	-0.0023	-0.0170	-0.0135	-0.0120
13	-0.0127	-0.0118	-0.0077	0.0173	0.0485
14	0.0092	-0.0306	0.0007	0.0192	0.0495
15	0.0086	0.0391	0.0512	-0.0114	0.0192
16	-0.0416	-0.0211	-0.0186	-0.0217	-0.0109
17	0.0292	0.0395	-0.0193	-0.0208	-0.0004
18	-0.0105	-0.0304	-0.0495	-0.0306	-0.0303
19	-0.0305	0.0296	-0.0395	0.0394	-0.0503
20	0.0395	-0.0004	-0.0295	0.0194	-0.0503
21	0.0195	-0.0204	-0.0395	-0.0106	-0.0203
22	0.0095	-0.0204	0.0105	-0.0306	0.0097
23	0.0395	-0.0004	-0.0095	0.0294	0.0197
24	0.0495	0.0396	0.0005	0.0494	0.0097

**Tabel 3.18** Bobot awal ke lapisan *output* ( $w_{jk}$ )

	Y
Z1	0.4495
Z2	0.3203
Z3	-0.4002
Z4	0.4598
Z5	0.2688

Wo = 0.001

**Langkah 5 :** hitung keluaran unit *hidden* ( $Z_j$ )

Masing-masing unit *hidden* menjumlahkan bobot sinyalnya, pada langkah ini menggunakan persamaan 2.15, hasilnya ditunjukkan pada Tabel 3.19.

**Tabel 3.19** Operasi pada *hidden*

J	Zin
1	0.0145
2	-0.0443
3	0.4380
4	-0.0112
5	0.0947

Aktivasi  $z_{in}$

**Tabel 3.20** Hasil Aktivasi pada *Hidden*

J	Z
1	0.0166
2	-0.0507
3	0.4873
4	-0.0128
5	0.1082

**Langkah 6:** Hitung keluaran unit  $Y_k$

Masing-masing unit *output* menjumlahkan bobot sinyal *input*.

$$y_{in} = -0.179588$$

Hasil aktivasinya :  $y = -0.204460$

**Langkah 7:** Hitung faktor  $\delta$  di unit *output*

Setiap unit *output*  $y$  menerima vektor hasil yang diinginkan ( $t$ ) untuk data masukan tersebut.

$$\delta = (t-y) * f'(y) = 1.35254$$

Hitung nilai koreksi bobotnya dengan  $\alpha=0.01$

**Tabel 3.21** Selisih Bobot W<sub>jk</sub>

	$\Delta W$
1	0.000224
2	-0.00069
3	0.006591
4	-0.00017
5	0.001463

Langkah 8: Hitung kesalahan di unit *hidden*

Setiap unit *hidden* menjumlahkan delta masukannya (dari unit-unit pada lapisan diatasnya).

**Tabel 3.22** Faktor Penimbang di unit *hidden*

j	$\Delta \delta_{in}$
1	0.607967
2	0.433219
3	-0.54129
4	0.621898
5	0.363563

Faktor kesalahan  $\delta$  di unit *hidden*

**Tabel 3.23** Aktivasi faktor Penimbang di unit *hidden*

j	$\delta$
1	0.695408
2	0.495141
3	-0.56926
4	0.71137
5	0.414238

Suku perubahan bobot ke unit *hidden* seperti pada tabel 3.24

**Tabel 3.24** Perbaikan bobot penimbang  $\Delta V_{ij}$ 

V <sub>ij</sub>	1	2	3	4	5
1	0.003199	0.002278	-0.002619	0.003272	0.001905
2	0.003964	0.002822	-0.003245	0.004055	0.002361
3	0.003964	0.002822	-0.003245	0.004055	0.002361
4	0.003894	0.002773	-0.003188	0.003984	0.002320
5	0.004729	0.003367	-0.003871	0.004837	0.002817
6	0.006259	0.004456	-0.005123	0.006402	0.003728
7	0.004937	0.003516	-0.004042	0.005051	0.002941
8	0.004520	0.003218	-0.003700	0.004624	0.002693
9	0.003894	0.002773	-0.003188	0.003984	0.002320
10	0.003825	0.002723	-0.003131	0.003913	0.002278
11	0.004451	0.003169	-0.003643	0.004553	0.002651
12	0.004381	0.003119	-0.003586	0.004482	0.002610
13	0.003407	0.002426	-0.002789	0.003486	0.002030
14	0.001043	0.000743	-0.000854	0.001067	0.000621
15	0.001739	0.001238	-0.001423	0.001778	0.001036
16	0.002086	0.001485	-0.001708	0.002134	0.001243
17	0.000974	0.000693	-0.000797	0.000996	0.000580
18	0.000695	0.000495	-0.000569	0.000711	0.000414
19	0.000695	0.000495	-0.000569	0.000711	0.000414
20	0.000695	0.000495	-0.000569	0.000711	0.000414
21	0.000695	0.000495	-0.000569	0.000711	0.000414
22	0.000695	0.000495	-0.000569	0.000711	0.000414
23	0.000695	0.000495	-0.000569	0.000711	0.000414
24	0.000695	0.000495	-0.000569	0.000711	0.000414

**Langkah 9 :** hitung semua perubahan bobot di unit *output*

Setelah semua pola dihitung, lakukan perubahan bobot (dilakukan pada setiap iterasi) yaitu:

**Table 3.25** Bobot baru *hidden-output layer*

	W <sub>i</sub>
1	0.449276
2	0.320985
3	-0.40679
4	0.459973
5	0.267337

Perubahan bobot unit *hidden*

Table 3.26 Bobot baru *input-hidden layer*

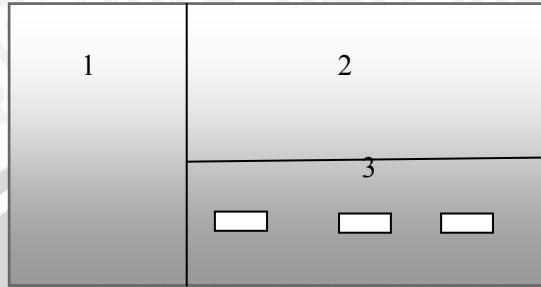
	Z1	Z2	Z3	Z4	Z5
1	-0.015699	-0.013978	-0.025181	-0.035872	-0.033305
2	0.022936	-0.044922	-0.044055	-0.037255	0.015839
3	0.002936	-0.024922	-0.034055	-0.047255	0.015839
4	0.043006	0.005127	0.025888	-0.027084	-0.004020
5	-0.008429	0.024133	0.027071	0.001363	0.015083
6	0.018841	-0.027856	0.009423	0.028598	0.033472
7	-0.028837	-0.006116	0.047442	-0.049051	0.034859
8	-0.018120	0.004382	0.036800	0.011776	0.035307
9	-0.026994	-0.004873	0.005888	0.012916	0.015980
10	-0.006825	-0.014823	0.035731	0.002987	-0.043978
11	0.022049	-0.005569	0.036743	0.041847	-0.024651
12	-0.047781	-0.005419	-0.013414	-0.017982	-0.014610
13	-0.016107	-0.014226	-0.004911	0.013814	0.046470
14	0.008157	-0.031343	0.001554	0.018133	0.048879
15	0.006861	0.037862	0.052623	-0.013178	0.018164
16	-0.043686	-0.022585	-0.016892	-0.023834	-0.012143
17	0.028226	0.038807	-0.018503	-0.021796	-0.000980
18	-0.011195	-0.030895	-0.048931	-0.031311	-0.030714
19	-0.031195	0.029105	-0.038931	0.038689	-0.050714
20	0.038805	-0.000895	-0.028931	0.018689	-0.050714
21	0.018805	-0.020895	-0.038931	-0.011311	-0.020714
22	0.008805	-0.020895	0.011069	-0.031311	0.009286
23	0.038805	-0.000895	-0.008931	0.028689	0.019286
24	0.048805	0.039105	0.001069	0.048689	0.009286

Setelah dilakukan perbaikan bobot sampai max epoch dan Error Min, maka hasil perubahan bobot akan disimpan untuk digunakan pada saat ada data baru yang akan dilakukan pengenalan. Pencocokan pola dideteksi dengan pemilihan kesalahan terkecil. Jika kesalahan terkecil terjadi pada *dataset* data ke i maka data akan dikenali dengan data ke i.

### 3.5 Rancangan Antarmuka (*Interface*)

Pada antarmuka pelatihan (gambar 3.13) terdiri dari beberapa bagian antara lain:

1. Menu untuk menentukan parameter-parameter yang digunakan untuk pelatihan.
2. Tampilan Input gambar.
3. Tombol menu untuk *open* gambar, pelatihan dan pengenalan.



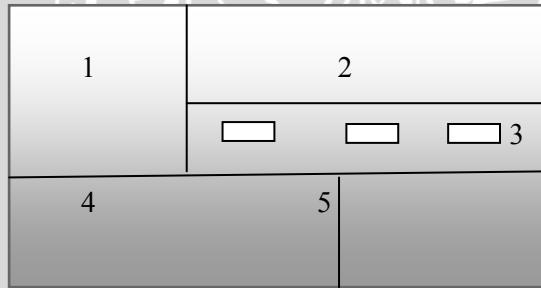
Gambar 3.13 Antarmuka Pelatihan

Pada antarmuka pengenalan seperti pada gambar

3.14 terdiri

dari beberapa bagian antara lain:

1. Menu untuk masukkan parameter-parameter.
2. Tampilan *Input* gambar
3. Tampilan tombol menu untuk *open* gambar, pelatihan dan pengenalan.
4. Tampilan hasil uji pengenalan
5. Tampilan hasil pengenalan ‘KANKER’ atau ‘NORMAL’.



Gambar 3.14 Antarmuka Pengenalan



## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### 4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan untuk pengembangan sistem pendekripsi kanker payudara.

##### 4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem pendekripsi kanker payudara dengan *mammogram* menggunakan JST *Backpropagation* adalah

1. Prosesor Intel Celeron 2.0 Ghz
2. Memori 512 MB
3. Harddisk dengan kapasitas 80 GB
4. Monitor 14"
5. Keyboard
6. Mouse
7. Scanner Canon Scan 300 ex dengan

##### 4.1.2 Lingkungan Perangkat Lunak

Perangkat keras yang digunakan dalam pengembangan sistem kanker payudara dengan *mammogram* menggunakan JST *Backpropagation* ini adalah

1. Sistem Operasi Windows XP
2. Borland Delphi 7
3. Text editor Notepad

#### 4.2 Implementasi Program

##### 4.2.1 Implementasi proses *Pre processing*

Berdasarkan analisa dan perancangan proses yang terdapat dalam sub bab 3.2.1, maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut. Inisialisasi awal yang digunakan dalam proses *pre processing* adalah:

```
const BLOCK_SIZE = 10;
const SEGMENTATION_ROW = 30;
const SEGMENTATION_COL = 20;

type
    ArrayDataImage = Array[0..(SEGMENTATION_ROW * SEGMENTATION_COL)-1] of Real;
```

`BLOCK_SIZE` adalah ukuran blok yang digunakan untuk membagi image menjadi ukuran  $10 \times 10$ . `SEGMENTATION_ROW` merupakan ukuran baris untuk image baru setelah dibagi dengan `BLOCK_SIZE` sedangkan `SEGMENTATION_COL` merupakan ukuran kolom untuk

```
procedure ConvertToImageBiner(Var imgBreast : TImage);
var
    i, j : Integer;
    Red, Green, Blue, Gray : Byte;
    theColor : TColor;
    T : Integer;
begin

    T := 220;

    for i := 0 to imgBreast.Picture.Graphic.Width - 1 do
    begin
        for j := 0 to imgBreast.Picture.Graphic.Height - 1 do
        begin
            theColor := imgBreast.Canvas.Pixels[i,j];
            Red := GetRValue(theColor);
            Green := GetGValue(theColor);
            Blue := GetBValue(theColor);

            Gray := Round(0.299 * Red + 0.587 * Green +
                          0.114 * Blue);

            if (Gray < T) then
                imgBreast.Canvas.Pixels[i,j] := RGB(0,0,0)
            else
                imgBreast.Canvas.Pixels[i,j] := RGB(255,255,255);
        end;
    end;
end;
```

`image` baru setelah dibagi dengan `BLOCK_SIZE`.

**Gambar 4.1** Prosedur ConvertToImageBiner

Pada proses *pre-processing*, langkah pertama yang dilakukan adalah menentukan *threshold* yang akan digunakan untuk mengubah gambar menjadi gambar hitam dan putih. Dalam implementasinya ukuran *threshold* diset = 220. Prosedur untuk mengubah gambar menjadi hitam dan putih ditunjukkan pada Gambar 4.1

```
function ImageBlockMean(imgBreast : TImage; x : Integer; y : Integer) : Real;
var
  i, j, temp, Nilai : Integer;
begin
  temp := 0;
  for i := x to (x + BLOCK_SIZE - 1) do
  begin
    for j := y to (y + BLOCK_SIZE - 1) do
    begin
      if (IntToStr(Round(GetRValue(imgBreast.Canvas.Pixels[i,j]))) = '0') then
        Nilai := 0
      else
        Nilai := 1;

      temp := temp + Nilai;
    end;
  end;
  Result := temp / (BLOCK_SIZE * BLOCK_SIZE);
end;

function ImageToMeanMatrix(imgBreast : TImage) : ArrayDataImage;
var
  i, j, index : Integer;
  temp : ArrayDataImage;
begin
  index := 0;
  for i := 0 to SEGMENTATION_ROW - 1 do
  begin
    for j := 0 to SEGMENTATION_COL - 1 do
    begin
      temp[index] := RoundTo(ImageBlockMean
                             (imgBreast, i*(BLOCK_SIZE),
                             j*(BLOCK_SIZE)), -2);
      Inc(index);
    end;
  end;
  Result := temp;
end;
```

Gambar 4.2 Fungsi ImageToMeanMatrik

```
function ImageBlockVariance(imgBreast : TImage; x : Integer;
y : Integer) : Real;
var
  i, j, Nilai : Integer;
  Mean, temp : Real;
begin
  temp := 0;
  Mean := ImageBlockMean(imgBreast, x, y);
  for i := x to (x + BLOCK_SIZE - 1) do
    begin
      for j := y to (y + BLOCK_SIZE - 1) do
        begin
          if(IntToStr(Round(GetRValue(imgBreast.
            Canvas.Pixels[i,j]))) = '0') then
            Nilai := 0
          else
            Nilai := 1;

          temp := temp + Power((Nilai - Mean), 2);
        end;
    end;
  Result := temp / (BLOCK_SIZE * BLOCK_SIZE);
end;

function ImageToVarianceMatrix(imgBreast : TImage) :
ArrayDataImage;
var
  i , j, index : Integer;
  temp : ArrayDataImage;
begin
  index := 0;
  for i := 0 to SEGMENTATION_ROW - 1 do
    begin
      for j := 0 to SEGMENTATION_COL - 1 do
        begin
          temp[index] := RoundTo(ImageBlockVariance
            (imgBreast, i*(BLOCK_SIZE),
             j*(BLOCK_SIZE)), -2);
          Inc(index);
        end;
    end;
  Result := temp;
end;
```

**Gambar 4.3 Fungsi ImageToVarianMatrik**

Pada proses selanjutnya yaitu menentukan ukuran blok, dalam implementasinya ukuran blok di set menjadi 10x10. Sehingga ukuran tiap blok dari gambar 200x300 disegmentasi menjadi 20x30. Setelah

dilakukan penentuan ukuran blok, untuk masing-masing blok dihitung nilai *mean* dan *variancanya*. Kemudian dari nilai *mean* dan *variance* tiap blok akan dijadikan matrik menjadi 20x30. Fungsi untuk menentukan nilai *mean* ke dalam matrik ditunjukkan pada Gambar 4.2 sedangkan nilai *variance* ke dalam matrik ditunjukkan pada Gambar 4.3.

Langkah selanjutnya dilakukan kukan proses normalisasi untuk mendapatkan nilai intensitas *grayscale* gambar sehingga memiliki range 0,1 sampai 0,9. Dari hasil normalisasi nantinya akan dijadikan inputan ke dalam JST yang digunakan untuk pelatihan dan pengenalan. Prosedur normalisasi ditunjukkan pada Gambar 4.4

```
procedure Normalisation(var tempArray : ArrayDataImage);
var
    i : Integer;
    Min, Max : Real;
begin
    Min := getMinimum(tempArray);
    Max := getMaximum(tempArray);

    for i := 0 to Length(tempArray) - 1 do
    begin
        tempArray[i] := RoundTo(((0.8*(tempArray[i] -
            Min)) / (1 + Max - Min)) + 0.1, -
            2);
    end;
```

Gambar 4.4 Prosedur Normalisasi Data

#### 4.2.2 Implementasi Proses JST

Inti dari proses pelatihan dan pengenalan pada gambar kanker payudara ini terletak pada pemanfaatan bobot, target dan gambar yang disimpan dalam dataset. Struktur data penyimpanan data dalam dataset tersebut berbentuk record. Pada proses pelatihan ini memiliki 4 macam dataset, masing-masing mewakili data yang disimpan sesuai jenisnya yaitu mean payudara kiri.txt, mean payudara kanan.txt, varian payudara kiri.txt, varian payudara kanan.txt, mean variance payudara kiri dan mean variance payudara kanan. List program struktur data tersebut adalah sebagai berikut:

```
type
    ArrayDuaDimensi = Array of Array of Real;
    ArraySatuDimensi = Array of Real;
```

```
type
  ListDS = ^DataSet;
  DataSet = record
    DS_ImageName : String;
    DS_p          : Integer;
    DS_t          : Real;
    DS_Vij        : ArrayDuaDimensi;
    DS_Voj        : ArraySatuDimensi;
    DS_Wjk        : ArraySatuDimensi;
    DS_Wok        : Real;
  end;
```

*DS\_ImageName* digunakan untuk menyimpan nama file gambar pelatihan dalam dataset. *DS\_p* merupakan penyimpanan jumlah *neuron hidden* yang dimasukkan *user* tiap gambar pada dataset. *DS\_t* merupakan penyimpanan target *output* yang dimasukkan *user* tiap gambar pada dataset. *DS\_Vij* tempat penyimpanan bobot koneksi dari *input* ke lapisan *hidden* pada dataset, sedangkan *DS\_Voj* tempat penyimpanan bias pada unit *input*. *DS\_Wjk* tempat penyimpanan bobot koneksi dari lapisan *hidden* menuju lapisan *output* pada dataset, sedangkan *DS\_Wok* tempat penyimpanan bias pada unit *output* pada dataset.

Setiap gambar yang dilatih akan mempunyai struktur data sesuai jenisnya yaitu *Imagename*, *p*, *t*, *Vij*, *Voj*, *Wjk* dan *Wok*. Seluruh data adalah hasil proses pelatihan yang disimpan pada dataset sesuai jenisnya. Data yang akan disimpan akan digunakan dalam proses pengenalan.

#### 4.2.2.1 Implementasi Proses Pelatihan

Langkah pertama adalah *user* memilih unit yang akan dilatih dan memasukan data-data yang akan dipakai untuk proses pelatihan. Data-data tersebut meliputi data jumlah *neuron hidden layer*, *learningrate*, iterasi maksimum, error toleransi dan target.

Langkah kedua ialah inisialisasi bobot untuk menentukan nilai awal bobot dari *input layer* ke *hidden layer* dan dari *hidden layer* ke *output layer*. Bobot diinisialisasi secara acak dengan nilai antara -0.05 sampai 0.05. Prosedurnya ditunjukkan pada Gambar 4.5

```
procedure InitializationWeight();
var
    i , j : Integer;
    tempBeta : Real;
begin
    tempBeta := Beta(neuron_input, neuron_hidden);

    Randomize;
    for i :=0 to neuron_input - 1 do
    begin
        for j := 0 to neuron_hidden - 1 do
        begin
            array_BobotOutput[i,j] := RandomValue(0.05);
        end;
    end;

    for i := 0 to neuron_hidden - 1 do
    begin
        array_Voj[i] := RandomValue(tempBeta);
    end;

    for i := 0 to neuron_input - 1 do
    begin
        for j := 0 to neuron_hidden - 1 do
        begin
            array_BobotOutput[i,j] := RoundTo(((tempBeta *
                array_BobotOutput[i,j]) / GetVj(j)), -4);
        end;
    end;

    for i := 0 to neuron_hidden - 1 do
    begin
        array_bobothidden[i] := RoundTo(RandomValue(0.05), -4);
    end;

    Wok := RoundTo(RandomValue(0.05), -4);
end;
```

Gambar 4.5 Prosedur Inisialisasi Bobot

Dalam proses pelatihan terdapat beberapa prosedur yang dijalankan diantaranya adalah *feedforward*, *backpropagation*, perhitungan error dan perubahan bobot. Gambar 4.6 menunjukan prosedur pelatihan.

```
begin

    ConvertToImageBiner(imgBreast);

    temp := ImageToMeanMatrix(imgBreast);

    Normalisation(temp);
    with formMenu do
        begin
            n := SEGMENTATION_ROW * SEGMENTATION_COL;
            p := StrToInt(editNeuronHidden.Text);
            l := StrToFloat(editLearningRate.Text);
            m := StrToInt(editIteration.Text);
            er := StrToFloat(editError.Text);
            t := StrToFloat(editTarget.Text);
        end;

    SetParameter(n, p, l, m, er, t);
    InitialitationBackpropagation(temp);
    InitialitationWeight();

    epoch := 0;

    while (epoch < m) do
    begin
        FeedForward();

        tempError := Error(GetY);
        if (tempError > er) then
        begin
            BackPropagation();
        end
        else
        begin
            break;
        end;

        Inc(epoch);
    end;
end;
```

Gambar 4.6 Prosedur Pelatihan

Prosedur *feedforward* merupakan fungsi untuk mengaktifkan neuron-neuron pada *hidden layer* dan *output layer*. Prosedurnya dapat dilihat pada Gambar 4.7.

```
function GetZ_in(j : Integer) : Real;
var
  i : Integer;
  temp : Real;
begin
  temp := 0;
  for i := 0 to neuron_input - 1 do
  begin
    temp := temp + (array_X[i] * array_Vij[i,j]);
  end;

  Result := RoundTo((array_Voj[j] + temp),-4) ;
end;
function GetY_in() : Real;
var
  i : Integer;
begin
  Y_in := 0;
  for i := 0 to neuron_hidden - 1 do
  begin
    Y_in := Y_in + (array_Z[i] * array_Wjk[i]);
  end;

  Result := RoundTo((Wok + Y_in),-4);
end;
procedure FeedForward();
var
  i : Integer;
begin
  for i := 0 to neuron_hidden - 1 do
  begin
    array_Z[i] := Activation(GetZ_in(i));
  end;

  Y := RoundTo(Activation(GetY_in()),-4);
end;
```

**Gambar 4.7** Prosedur *feedforward*

Nilai *output* (*y*) yang diperoleh dari proses *feedforward*, digunakan untuk menghitung informasi error pada *output layer*. Selanjutnya dapat dihitung nilai error. Fungsi untuk menghitung error adalah sebagai berikut:

```
function Error(a : Real) : Real;
begin
  Result := RoundTo(Power((target - a),2)*0.5, -5);
end;
```

**Gambar 4.8** Prosedur fungsi Hitung *Error*

Selanjutnya dihitung informasi kesalahan pada *output layer* (Gambar 4.9) dan *hidden layer* (Gambar 4.10)

```
function ErrorDiOutput(a : Real) : Real;
begin
    Result := RoundTo((target - a) *
                       ActivationDerivative(a), -4);
end;

procedure KoreksiBobotWjk();
var
    i : Integer;
begin
    for i := 0 to neuron_hidden - 1 do
    begin
        array_DeltaBobotOutput[i] := RoundTo((learning_rate *
                                              ErrorDiOutput(GetY) * array_Z[i]), -4);
    end
end;
```

Gambar 4.9 Prosedur Hitung Error di *Output Layer*

```
procedure ErrorDiHidden();
var
    i : Integer;
begin
    for i := 0 to neuron_hidden - 1 do
    begin
        array_ErrorDiHidden[i]:=RoundTo((ErrorDiOutput(Y)*
                                         array_bobotHidden[i]*ActivationDerivative(array_Zin[i])), -4);
    end
end;

procedure DeltaBobotVij();
var
    i, j : Integer;
begin
    for i := 0 to neuron_input - 1 do
    begin
        for j := 0 to neuron_hidden - 1 do
        begin
            array_DeltaBobotHidden[i,j] := RoundTo((learning_rate *
                                              array_ErrorDiHidden[j] * array_X[i]), -4);
        end;
    end;
end;
```

Gambar 4.10 Prosedur Hitung Error di *Hidden layer*

Hasil dari error di *output layer* dan *hidden layer* digunakan untuk memperbaiki bobot. Prosedur pengupdatetan bobot ditunjukkan pada Gambar 4.11.

```
procedure UpdateBobotVij();
var
  i, j : Integer;
begin
  for i := 0 to neuron_input - 1 do
    begin
      for j := 0 to neuron_hidden - 1 do
        begin
          array_bobotOutput[i,j]:=RoundTo((array_BobotOutput[i,j] +
            array_DeltaBobotHidden[i,j]),-4);
        end;
      end;
    end;
  procedure UpdateBobotWjk();
  var
    i : Integer;
  begin
    for i := 0 to neuron_hidden - 1 do
      begin
        array_bobotHidden[i] := RoundTo((array_bobotHidden[i] +
          array_DeltaBobotOutput[i]),-4);
      end
    end;
end;
```

Gambar 4.11 Prosedur *Update Bobot*

Prosedur *update bobot* digunakan untuk menghitung bobot baru. *Array\_Vij* adalah variabel untuk menyimpan bobot dari *input layer* menuju *hidden layer*. Sedangkan *array\_Wjk* adalah variabel untuk menyimpan bobot dari *hidden layer* menuju *output layer*. Nilai bobot hasil pelatihan akan disimpan dalam file text (\*.txt) sesuai jenis *inputan* yang dipilih. Bobot ini akan digunakan pada proses pengenalan. Prosedur untuk menyimpan bobot adalah:

```
procedure TformImage.SaveToFileText(FileText : String);
var
  F : TextFile;
  str_Vij : String;
  str_Voj : SString;
  str_Wjk : String;
  str_Wok : String;
  str_t : String;
  Path : String;
  i, j : Integer;
begin
  str_Vij := NullAsStringValue;
  str_Voj := NullAsStringValue;
  str_Wjk := NullAsStringValue;
  str_Wok := NullAsStringValue;
  Path := ExtractFilePath(Application.ExeName);
  FileText := Path + FileText;
```

### Lanjutan Prosedur Simpan

```
for i := 0 to n - 1 do
begin
    for j := 0 to p - 1 do
    begin
        str_Vij := str_Vij + ' ' + FloatToStr
                    (GetArray_BobotOutput[i,j]);
    end;
end;
for i := 0 to p - 1 do
begin
    str_Voj := str_Voj + ' ' + FloatToStr
                    (GetArray_Voj() [i]);
end;
for i := 0 to p - 1 do
begin
    str_Wjk := str_Wjk + ' ' + FloatToStr (GetBobotHidden[i]);
end;
str_Wok := FloatToStr (GetWok());
str_t := FloatToStr (RoundTo(t, -1));

if (FileExists (FileText)) then
begin
    AssignFile (F, FileText);
    Append (F);
    Writeln (F, FileImage);
    Writeln (F, p);
    Writeln (F, str_t);
    Writeln (F, str_Vij);
    writeln (F, str_Voj);
    Writeln (F, str_Wjk);
    Writeln (F, str_Wok);
    Flush (F);
    CloseFile (F);
end
else
begin
    AssignFile (F, FileText);
    ReWrite (F);
    Writeln (F, FileImage);
    Writeln (F, p);
    Writeln (F, str_t);
    Writeln (F, str_Vij);
    writeln (F, str_Voj);
    Writeln (F, str_Wjk);
    Writeln (F, str_Wok);
    Flush (F);
    CloseFile (F);
end;
end;
```

**Gambar 4.12** Prosedur Simpan

#### 4.2.2.2 Implementasi Proses Pengenalan

Proses ini dilakukan untuk mendeteksi kanker payudara menggunakan mammogram. Proses yang dilakukan adalah melakukan *feedforward* sehingga didapatkan nilai keluaran (Y). Bobot yang digunakan adalah bobot dari hasil pelatihan. Selanjutnya dihitung jarak error terdekat dengan data yang tersimpan pada dataset. Proses pengenalan ditunjukkan pada Gambar 4.13.

```
procedure TForm1.Image1.BtnRecognizeClick(Sender: TObject);
var
  i : Integer;
  temp : ArrayDataImage;
  resultTarget : Real;
  Minimum : Real;
  Value : String;
begin
  Clear();
  ListRecognize.Clear;
  if (formMenu.rbMean.Checked = True) and (rbLeft.Checked = True) then
    LoadFromFileText('Mean Payudara Kiri.txt')
  else if (formMenu.rbMean.Checked = True) and (rbRight.Checked = True) then
    LoadFromFileText('Mean Payudara Kanan.txt')
  else if (formMenu.rbVariance.Checked = True) and (rbLeft.Checked = True) then
    LoadFromFileText('Variance Payudara Kiri.txt')
  else if (formMenu.rbVariance.Checked = True) and (rbRight.Checked = True) then
    LoadFromFileText('Variance Payudara Kanan.txt');
  if (ListDataSet.Count > 0) then
  begin
    ConvertToImageBiner(imgBreast);
    if (formMenu.rbMean.Checked) then
      temp := ImageToMeanMatrix(imgBreast)
    else if (formMenu.rbVariance.Checked) then
      temp := ImageToVarianceMatrix(imgBreast);
    else if (formMenu.rbMeanVariance.Checked) then
      temp := ImageToMeanVarianceMatrix(imgBreast);

    Normalisation(temp);
    MyDataSet := ListDataSet.Items[i];

    with formMenu do
    begin
      n := SEGMENTATION_ROW * SEGMENTATION_COL;
      p := MyDataSet^.DS_p;
      l := StrToFloat(editLearningRate.Text);
      m := StrToInt(editIteration.Text);
      e := StrToFloat(editError.Text);
      t := MyDataSet^.DS_t;
```

## Lanjutan Prosedure Pengenalan

```
.....  
    end;  
    SetParameter(n, p, l, m, e, t);  
    InitialitationBackpropagation(temp);  
  
    SetData (MyDataSet^.DS_Vij, MyDataSet^.DS_Voj,  
    MyDataSet^.DS_Wjk, MyDataSet^.DS_Wok);  
    FeedForward();  
    New (MyRecognize);  
    MyRecognize^.Rec_ImageName := MyDataSet^.DS_ImageViewName;  
    MyRecognize^.Rec_t := MyDataSet^.DS_t;  
    MyRecognize^.Rec_error := ErrorRecognize(GetY);  
    ListRecognize.Add(MyRecognize);  
    end;  
  
    MyRecognize := ListRecognize.Items[0];  
    Minimum := MyRecognize^.Rec_error;  
    resultTarget := MyRecognize^.Rec_t;  
    for i := 1 to ListRecognize.Count - 1 do  
    begin  
        MyRecognize := ListRecognize.Items[i];  
        if (MyRecognize^.Rec_error <= Minimum) then  
        begin  
            Minimum := MyRecognize^.Rec_error;  
            resultTarget := MyRecognize^.Rec_t;  
        end;  
    end;  
    resultTarget := RoundTo(resultTarget,-1);  
    lblTempTarget.Caption := FloatToStr(resultTarget);  
    if (FloatToStr(resultTarget) = '0.9') then  
        formSummary.lblResult.Caption := Kondisi[1]  
    else if (FloatToStr(resultTarget) = '0.1') then  
        formSummary.lblResult.Caption := Kondisi[0];  
    end;  
end.
```

Gambar 4.13 Prosedur Pengenalan

### 4.3 Implementasi Antarmuka (*Interface*)

Berdasarkan rancangan antarmuka pada sub bab 3.5, dihasilkan antarmuka untuk pendekripsi kanker payudara. Tampilan utama dari implementasi program ini adalah *form* yang digunakan untuk pelatihan sekaligus pengenalan. Pada *form* ini terdapat 3 *form* yang digunakan, yaitu *form image*, *form menu* dan *form summary*. Tampilannya ditunjukkan pada gambar 4.14.



Gambar 4.14 Antarmuka Utama

Pada *form menu* digunakan untuk menentukan dan mengatur parameter JST. *Form image* digunakan untuk menentukan gambar yang digunakan untuk pelatihan atau pengenalan, tempat tombol *open*, *training* dan *recognice*. Serta menyediakan tempat untuk nilai awal data *input* yang sudah dinormalisasi.

Langkah pertama yang perlu dilakukan adalah menentukan arsitektur jaringan dan mengatur nilai parameter yang akan digunakan dalam proses pelatihan dan pengenalan, kemudian mengatur nilai gambar yang digunakan dalam pelatihan serta memilih bagian payudara yang akan di deteksi.

Pada Gambar 4.14 terlihat bahwa jaringan terbentuk dengan *neuron hidden* 30, *learningrate* 2.0, *Max iterasi* 5000 *epoch* dan *error tolerance* 0. Setelah parameter jaringan ditentukan, maka langkah selanjutnya menentukan *image valuenya* yaitu berdasarkan nilai *mean*, *variance* atau perbandingan *mean variance*, kemudian menentukan bagian payudara yang akan dideteksi, yaitu *left breast* dan *right breast*. Setelah semua parameter ditentukan maka proses pelatihan dapat dilakukan.

User memilih tombol *open* yang merupakan tempat masukkan gambar yang akan dilatih. Setelah gambar dimasukkan, kemudian user memilih tombol *Training*, maka proses pelatihan akan diproses. Pada proses pelatihan akan ditunjukkan hasil pelatihan seperti pada gambar 4.15



Gambar 4.15 Antarmuka Pelatihan

Pada pelatihan akan ditampilkan nilai awal data *input* kemudian nilai *error minimal* pada tiap iterasi atau epoch dan nilai output. Pada Gambar 4.15, pelatihan berhenti pada epoch ke-8 dengan nilai kesalahan sebesar nol. Dari antarmuka tersebut terlihat hasil pelatihan menghasilkan nilai *output* yang sama dengan target. Ini adalah indikasi bahwa proses pelatihan berhasil. Setelah proses pelatihan selesai, maka bobot secara otomatis akan disimpan pada dataset.

Setelah pelatihan selesai, untuk mengetahui uji hasil pembelajaran dapat dilakukan dengan menekan tombol recognize (lihat gambar 4.16).



Gambar 4.16 Antarmuka Pengenalan

#### 4.4 Hasil dan Pembahasan

##### 4.4.1 Hasil Percobaan

###### a. Pengaruh Jumlah Neuron

Tahap terpenting dalam implementasi adalah mendapatkan struktur jaringan saraf yang efisien. Untuk memperoleh struktur jaringan syaraf tiruan yang terbaik untuk mendeteksi kanker payudara maka dilakukan pengujian terhadap sistem. Hal yang perlu ditentukan adalah menentukan nilai *learning rate*, jumlah neuron *hidden* dan target kesalahan. Penentuan jumlah *hidden neurons* pada sistem dilakukan melalui proses *trial and error*.

Dari beberapa percobaan yang dilakukan, didapatkan bahwa pada nilai *learning rate* yang sama maka akan menghasilkan kesalahan minimal yang sama. Sebagai data analisa, dilakukan percobaan pada *learning rate* 0.08. (sebelum didapatkan yang optimal, mengacu pada penelitian yang sudah ada sebelumnya). Tetapi setelah dilakukan percobaan dihasilkan nilai kesalahan minimal yang masih jauh dari rata-rata kesalahan yang ditargetkan. Dalam percobaan ini akan melakukan percobaan dengan *learning rate* 0.6 Kemudian jumlah neuron yang diujikan antara 5 hingga 50 dengan selang 5 neuron. Jumlah iterasi maksimal ditetapkan 5000 iterasi. Hasil percobaan tersebut dapat dilihat pada Tabel 4.1. Dapat dibuat grafik seperti pada Gambar 4.17

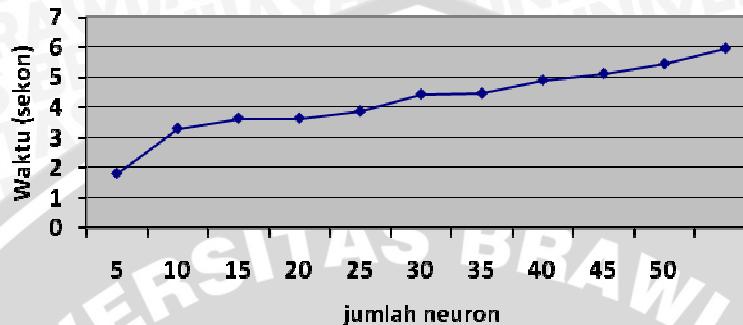
**Tabel 4.1** Pengaruh jumlah *neuron*

Jumlah neuron	Kesalahan minimal	Waktu
5	0.0000000	1s : 796ms
10	0.0000000	3s : 297ms
15	0.0000000	3s : 641ms
20	0.0000000	3s : 875ms
25	0.0000000	4s : 438ms
30	0.0000000	4s : 469ms
35	0.0000000	4s : 907ms
40	0.0000000	5s : 125ms
45	0.0000000	5s : 459ms
50	0.0000000	5s : 968ms

Berdasarkan Tabel 4.1 dan grafik pada Gambar 4.17, terlihat bahwa semakin banyak *neuron hidden* yang digunakan maka waktu

yang dibutuhkan untuk mencapai error minimal semakin lama. Hal ini disebabkan karena setiap unit input akan menerima sinyal dan menyalurkan ke lapisan *hidden*, jadi apabila menggunakan neuron *hidden* yang banyak akan membutuhkan waktu proses yang semakin lama. Pada percobaan pengujian selanjutnya akan dicobakan *neuron hidden* dengan jumlah 10, 20 dan 30.

**Pengaruh Jumlah Neuron Terhadap Waktu Pelatihan**



**Gambar 4.17** Grafik Pengaruh Jumlah Neuron

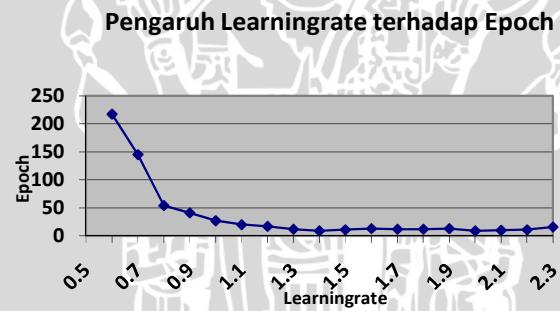
**b. Pengaruh Learning rate**

*Learning rate* pada pembelajaran menunjukkan seberapa cepat jaringan syaraf tiruan akan menyesuaikan diri dengan data pelatihan yang diterimanya. Dari hasil percobaan didapatkan, bahwa semakin kecil *learning rate* maka kemampuan pencapaian nilai kesalahan semakin lambat dan sulit mendekati kesalahan minimal. Sebaliknya semakin besar nilai *learning rate*, maka kecepatan mencapai bobot optimal semakin cepat. Pengaruh *learning rate* ini dapat dilihat pada grafik (Gambar 4.18).

Dari Tabel 4.2 terlihat nilai *learning rate* mencapai kesalahan minimal paling cepat pada *learning rate* 1.4 dan 2.0. Namun setelah tiap nilai *learning rate* dicobakan untuk seluruh pola masukan, tidak semua data tersebut baik untuk digunakan. Walaupun pada sebagian data terjadi keadaan seperti pada Tabel 4.2. Berikut ini digambarkan grafik pengaruh *learning rate* terhadap *epoch* atau iterasi pada Gambar 4.18.

**Tabel 4.2.** Pengaruh *Learning rate* terhadap epoch

Alpha	Epoch ke-
0.5	5000
0.6	217
0.7	145
0.8	54
0.9	41
1.0	27
1.1	20
1.2	17
1.3	12
1.4	8
1.5	11
1.6	13
1.7	12
1.8	12
1.9	13
2.0	9
2.1	10
2.2	11
2.3	16

**Gambar 4.18** Grafik Pengaruh *Learning Rate*

Setelah dilakukan percobaan *learning rate* dan jumlah *neuron hidden* dapat disimpulkan bahwa struktur jaringan yang relatif efisien untuk percobaan pengujian pengenalan ini dengan *inputan* sebanyak 600 *neuron input* yaitu jumlah *neuron hidden* antara 10-30 buah, sedangkan nilai *learning rate* yang dicobakan 1.4, dan 2.0 karena pada nilai tersebut menghasilkan *epoch* yang terkecil. Setelah dilakukan percobaan uji kemampuan struktur jaringan maka dilakukan percobaan pengujian pengenalan.

### c. Pengaruh ukuran Blok pada citra

Ukuran blok pada proses *preprocessing* gambar juga sangat mempengaruhi hasil pada pelatihan jaringan. Percobaan ini dilakukan dengan menggunakan *neuron hidden* sebanyak 30 dan *learning rate* sebesar 2.0. Pada percobaan ini akan digunakan gambar *mammogram* payudara kanan dengan jumlah 36 buah, dan untuk karakteristik data input digunakan adalah dari nilai *mean*.

**Tabel 4.3** Hasil Percobaan Pengaruh Ukuran Blok *mammogram*

Percobaan	Ukuran Blok (pixel)	Tes Pengenalan	
		Benar	Salah
1	5x 5	28	8
2	10x10	34	2
3	20 x 20	24	12

Dari percobaan tersebut menunjukkan bahwa ukuran blok 10 x 10 menghasilkan tingkat kesalahan yang kecil yaitu hanya menghasilkan 2 kesalahan pengenalan. Semakin kecil ukuran blok yang digunakan maka proses pelatihan akan semakin lama, begitu juga sebaliknya. Dengan hasil ini, percobaan berikutnya dilakukan dengan menggunakan data *input* dengan ukuran blok 10 x 10 piksel.

### 4.4.2 Analisa Hasil Percobaan

Dari hasil pelatihan akan didapat bobot yang akan digunakan untuk menguji kumpulan data uji dalam kemampuan pengenalan pola. Kumpulan pola data uji terdiri dari 30 data, yang terdiri dari 17 data payudara kiri dan 13 data payudara kanan. Table berikut menunjukkan hasil percobaan pengenalan dari data uji berdasarkan karakteristik data input gambar yang digunakan dan arsitektur JST yang berbeda-beda. Nilai *learning rate* yang digunakan berdasarkan Tabel 4.2 yang menghasilkan *epoch* terkecil.

**I. Pengujian *Mean Value* di payudara kiri****Tabel 4.4** Hasil pengenalan dengan *learning rate* 2.0

	DataUji	n=30	n=20	n=10
KANKER	1	Kanker	Kanker	Normal
	2	Kanker	Kanker	Normal
	3	Normal	Normal	Kanker
	4	Kanker	Normal	Normal
	5	Kanker	Kanker	Normal
NORMAL	1	Normal	Normal	Kanker
	2	Normal	Normal	Kanker
	3	Normal	Normal	Kanker
	4	Normal	Normal	Kanker
	5	Normal	Normal	Kanker

**Tabel 4.5** Hasil pengenalan dengan *learning rate* 1.4

	DataUji	n=30	n=20	n=10
KANKER	1	Normal	Normal	Normal
	2	Normal	Normal	Kanker
	3	Kanker	Normal	Normal
	4	Normal	Normal	Normal
	5	Kanker	Normal	Kanker
NORMAL	1	Normal	Normal	Normal
	2	Kanker	Normal	Normal
	3	Normal	Normal	Normal
	4	Kanker	Normal	Normal
	5	Normal	Normal	Normal

Berdasarkan Tabel 4.4 dan Tabel 4.5, pengenalan payudara kiri dengan menggunakan data input dari nilai *mean* dengan beberapa pengujian *learning rate* dan *neuron hidden*, arsitektur JST yang bisa menghasilkan pengenalan dengan akurasi pengenalan yang paling baik yaitu pada *learningrate* 2.0 dengan *neuron hidden* 30 yang hanya menghasilkan 1 kesalahan.

## II. Pengujian *Variance Value* di payudara Kiri

Tabel 4.6 Hasil pengenalan dengan *learning rate* 2.0

	DataUji	n=30	n=20	n=10
KANKER	1	Kanker	Normal	Normal
	2	Kanker	Normal	Normal
	3	Normal	Kanker	Normal
	4	Normal	Kanker	Kanker
	5	Kanker	Kanker	Normal
NORMAL	1	Normal	Normal	Kanker
	2	Normal	Normal	Kanker
	3	Kanker	Normal	Kanker
	4	Normal	Normal	Kanker
	5	Normal	Normal	Kanker

Tabel 4.7 Hasil pengenalan dengan *learning rate* 1.4

	DataUji	n=30	n=20	n=10
KANKER	1	Normal	Normal	Normal
	2	Kanker	Kanker	Normal
	3	Normal	Kanker	Normal
	4	Kanker	Normal	Normal
	5	Kanker	Kanker	Normal
NORMAL	1	Normal	Normal	Normal
	2	Normal	Kanker	Normal
	3	Normal	Normal	Normal
	4	Normal	Normal	Kanker
	5	Normal	Normal	Normal

Berdasarkan Tabel 4.6 dan Tabel 4.7, pengenalan payudara kiri dengan menggunakan data input *variance value* arsitektur JST yang bisa menghasilkan pengenalan dengan akurasi pengenalan yang paling baik yaitu pada *learningrate* 1.4 pada neuron hidden 30 yang menghasilkan 2 kesalahan pengenalan.

### III. Pengujian *Mean Value* di Payudara Kanan

Tabel 4.8 Hasil pengenalan dengan *learning rate* 2.0

	DataUji	n=30	n=20	n=10
KANKER	1	Kanker	Normal	Normal
	2	Kanker	Kanker	Normal
	3	Kanker	Kanker	Kanker
	4	Normal	Kanker	Normal
	5	Kanker	Kanker	Normal
NORMAL	1	Normal	Kanker	Normal
	2	Normal	Normal	Normal
	3	Normal	Kanker	Normal
	4	Normal	Kanker	Normal
	5	Normal	Kanker	Normal

Tabel 4.9 Hasil pengenalan dengan *learning rate* 1.4

	DataUji	n=30	n=20	n=10
KANKER	1	Normal	Normal	Normal
	2	Normal	Normal	Normal
	3	Normal	Kanker	Normal
	4	Normal	Normal	Normal
	5	Normal	Normal	Normal
NORMAL	1	Kanker	Normal	Normal
	2	Kanker	Normal	Normal
	3	Kanker	Normal	Normal
	4	Kanker	Normal	Normal
	5	Kanker	Normal	Normal

Berdasarkan Tabel 4.8 dan Tabel 4.9, pengenalan pada payudara kanan dengan karakteristik data input menggunakan *mean value*, arsitektur JST yang bisa menghasilkan pengenalan dengan tingkat akurasi paling baik adalah pada *learningrate* 2.0 dengan *neuron hidden* 30 yaitu hanya menghasilkan 1 kesalahan pengenalan.

**IV. Pengujian *Variance Value* di payudara Kanan****Tabel 4.10** Hasil pengenalan dengan *learning rate* 2.0

	DataUji	n=30	n=20	n=10
KANKER	1	Kanker	Normal	Kanker
	2	Normal	Kanker	Kanker
	3	Kanker	Kanker	Kanker
	4	Normal	Normal	Kanker
	5	Normal	Normal	Kanker
NORMAL	1	Normal	Normal	Kanker
	2	Normal	Normal	Kanker
	3	Normal	Normal	Kanker
	4	Normal	Normal	Kanker
	5	Normal	Normal	Kanker

**Tabel 4.11** Hasil pengenalan dengan *learning rate* 1.4

	DataUji	n=30	n=20	n=10
KANKER	1	Normal	Normal	Kanker
	2	Normal	Normal	Kanker
	3	Kanker	Normal	Kanker
	4	Normal	Normal	Normal
	5	Normal	Kanker	Normal
NORMAL	1	Normal	Normal	Normal
	2	Normal	Normal	Normal
	3	Normal	Normal	Normal
	4	Kanker	Normal	Normal
	5	Normal	Normal	Normal

Berdasarkan Tabel 4.10 dan Tabel 4.11, pengenalan pada payudara kanan menggunakan karakteristik data input *variance value*, arsitektur JST yang bisa menghasilkan pengenalan dengan tingkat akurasi paling baik yaitu pada *learningrate* 1.4 dengan *neuron hidden* 10 dengan 2 kesalahan pengenalan.

## V. Pengujian Perbandingan *Mean Variance Value* di Payudara Kiri

Tabel 4.12 Hasil pengenalan dengan *learning rate* 2.0

	DataUji	n=30	n=20	n=10
KANKER	1	Normal	Kanker	Kanker
	2	Kanker	Normal	Kanker
	3	Normal	Normal	Kanker
	4	Normal	Normal	Kanker
	5	Normal	Kanker	Kanker
	1	Normal	Kanker	Kanker
	2	Normal	Kanker	Kanker
	3	Normal	Kanker	Kanker
	4	Normal	Kanker	Kanker
	5	Normal	Kanker	Kanker

Tabel 4.13 Hasil pengenalan dengan *learning rate* 1.4

	DataUji	n=30	n=20	n=10
KANKER	1	Normal	Normal	Normal
	2	Normal	Kanker	Normal
	3	Kanker	Normal	Normal
	4	Normal	Normal	Normal
	5	Normal	Kanker	Normal
	1	Kanker	Kanker	Normal
	2	Kanker	Kanker	Normal
	3	Normal	Kanker	Normal
	4	Kanker	Kanker	Normal
	5	Kanker	Kanker	Normal

Berdasarkan Tabel 4.12 dan Tabel 4.13, pengenalan pada payudara kiri menggunakan karakteristik data input dari perbandingan *mean variance* dengan beberapa pengujian *learning rate* dan *neuron hidden*,arsitektur JST yang bisa menghasilkan pengenalan yang baik dibandingkan dengan yang lain yaitu pada *learning rate* 2.0 dengan *neuron hidden* 30.

**VI.Pengujian Perbandingan *Mean Variance Value* di Payudara Kanan****Tabel 4.14** Hasil pengenalan dengan *learning rate* 2.0

	DataUji	n=30	n=20	n=10
KANKER	1	Kanker	Kanker	Kanker
	2	Kanker	Normal	Kanker
	3	Kanker	Normal	Kanker
	4	Kanker	Normal	Kanker
	5	Kanker	Kanker	Kanker
NORMAL	1	Normal	Kanker	Kanker
	2	Normal	Kanker	Kanker
	3	Normal	Normal	Kanker
	4	Kanker	Kanker	Kanker
	5	Normal	Kanker	Kanker

**Tabel 4.15** Hasil pengenalan dengan *learning rate* 1.4

	DataUji	n=30	n=20	n=10
KANKER	1	Kanker	Normal	Kanker
	2	Kanker	Kanker	Kanker
	3	Kanker	Kanker	Kanker
	4	Normal	Kanker	Kanker
	5	Kanker	Kanker	Kanker
NORMAL	1	Normal	Kanker	Kanker
	2	Kanker	Normal	Kanker
	3	Normal	Kanker	Kanker
	4	Kanker	Kanker	Kanker
	5	Normal	Normal	Kanker

Berdasarkan Tabel 4.14 dan Tabel 4.15, pengenalan pada payudara kanan menggunakan karakteristik data input dari perbandingan *mean variance* dengan beberapa pengujian *learning rate* dan *neuron hidden*, arsitektur JST yang bisa menghasilkan pengenalan yang paling baik yaitu pada *learningrate* 2.0 dengan *neuron hidden* 30 dengan menghasilkan 1 kesalahan pengenalan.

#### 4.4.3 Penetapan Struktur Jaringan Syaraf Tiruan

Dari keseluruhan percobaan pengujian yang dilakukan dapat disimpulkan bahwa struktur jaringan yang relatif efisien untuk pendekripsi kanker payudara menggunakan *mammogram* dengan *inputan* sebanyak 600 *neuron* adalah jumlah *neuron hidden* 30 buah, nilai *learning rate* 2.0 dan nilai *error target* 0. Setelah diputuskan struktur jaringan tersebut, dilakukan percobaan uji kemampuan struktur jaringan. Dari 5 percobaan dapat dibuat tabel seperti pada Tabel 4.16.

**Tabel 4.16** Hasil percobaan uji kemampuan struktur jaringan

Epoch	Nilai kesalahan				
	p1	p2	p3	p4	p5
1	0.1161138	0.0571896	0.0957469	0.0885784	0.084625
2	0.0049302	0.004608	0.0049302	0.0048906	0.0046948
3	0.0045888	0.003604	0.0046272	0.0044462	0.0039516
4	0.000689	1.89E-03	0.0035955	0.0031601	0.0025205
5	0.0013313	0.0005249	0.001806	0.0014365	1.04E-03
6	0.0003001	0.0000769	5.02E-04	0.0003948	2.79E-04
7	0.000052	0.0000076	0.000884	7.81E-05	0.000053
8	0.0000097	1.30E-06	1.20E-05	1.51E-05	0.0000097
9	0.0000014	0.0000003	1.30E-06	2.00E-06	0.0000014
10	0.000001	0.0000001	0.0000007	0.0000011	0.0000007
11	0.0000007	0	0.0000004	5.00E-07	0.0000003
12	0.0000005		0.0000002	0.0000003	0.0000001
13	0.0000003		0.0000001	1.00E-07	0.0000001
14	0.0000002		0.0000001	0.0000001	0
15	0.0000002		0	0	
16	0.0000001				
17	0.0000001				
18	0				
19					

Pada Tabel 4.16 terlihat ada 5 percobaan, p1 hingga p5 dapat mencapai hasil yang konvergen. Seluruh percobaan dapat mencapai target yang ditetapkan, yaitu 0 (nol). Banyaknya iterasi atau epoch untuk setiap percobaan antara 12 hingga 19 iterasi. Ini berarti struktur jaringan dapat mencapai target dengan kesalahan yang

mencapai minimal, yaitu nilai keluaran mendekati dengan nilai target dengan kecepatan yang relative singkat. Sehingga struktur jaringan yang akan dipakai terdiri atas 600 neuron masukkan, 30 neuron hidden dan 1 neuron *output*, serta nilai *learning rate* sebesar 2.0. nilai target kesalahan pada proses pelatihan ditetapkan sebesar 0 (nol), karena struktur jaringan ini mampu mencapai kesalahan hingga 0 (nol).

#### 4.4.4 Hasil Pelatihan

Dari jaringan yang terbentuk dilakukan proses pelatihan dengan pemilihan data *input* yang tergolong baik. Masing-masing data *training* yang digunakan pada payudara kiri 25 buah dan payudara kanan 23 buah. Hasil pelatihan menunjukkan bahwa nilai *output* mendekati nilai target yang ditentukan, hal ini terlihat pada nilai kesalahan atau selisih target dan *output* adalah mendekati 0 (nol). Hasil pengenalan data pelatihan seperti pada Tabel berikut ini:

**Tabel 4.17** Hasil pelatihan dan tes pengenalan dengan *mean value* pada payudara kiri

Nama File	Tes
Kanker Kiri 1.bmp	Benar
Kanker Kiri 2.bmp	Benar
Kanker Kiri 4.bmp	Benar
Kanker Kiri 5.bmp	Benar
Kanker Kiri 7.bmp	Benar
Kanker Kiri 8.bmp	Benar
Kanker Kiri 9.bmp	Benar
Kanker Kiri 11.bmp	Benar
Kanker Kiri 12.bmp	Benar
Kanker Kiri 13.bmp	Benar
Kanker Kiri 14.bmp	Benar
Kanker Kiri 15.bmp	Benar
Kanker Kiri 16.bmp	Benar
Kanker Kiri 17.bmp	Benar
Kanker Kiri 18.bmp	Benar
Nama File	Tes
Normal Kiri 2.bmp	Benar
Normal Kiri 5.bmp	Benar
Normal Kiri 6.bmp	Benar
Normal Kiri 8.bmp	Benar
Normal Kiri 9.bmp	Benar
Normal Kiri 10.bmp	Benar
Normal Kiri 11.bmp	Benar
Normal Kiri 12.bmp	Benar
Normal Kiri 17.bmp	Benar
Normal Kiri 18.bmp	Benar

**Tabel 4.18** Hasil pelatihan dan tes pengenalan dengan *mean value* pada payudara kanan

Nama File	Tes
Kanker Kanan 1.bmp	Benar
Kanker Kanan 2.bmp	Benar
Kanker Kanan 3.bmp	Benar
Kanker Kanan 4.bmp	Benar
Kanker Kanan 6.bmp	Benar
Kanker Kanan 9.bmp	Benar
Kanker Kanan 10.bmp	Benar

Nama File	Tes
Normal Kanan 1.bmp	Benar
Normal Kanan 2.bmp	Benar
Normal Kanan 3.bmp	Benar
Normal Kanan 4.bmp	Benar
Normal Kanan 5.bmp	Benar
Normal Kanan 8.bmp	Benar
Normal Kanan 9.bmp	Benar
Normal Kanan 10.bmp	Benar
Normal Kanan 11.bmp	Benar
Normal Kanan 12.bmp	Benar
Normal Kanan 14.bmp	Benar
Normal Kanan 15.bmp	Benar
Normal Kanan 16.bmp	Benar
Normal Kanan 17.bmp	Benar
Normal Kanan 18.bmp	Benar

**Tabel 4.19** Hasil pelatihan dan tes pengenalan dengan *variance value* pada payudara kiri

Nama File	Tes
Kanker Kiri 1.bmp	Benar
Kanker Kiri 2.bmp	Benar
Kanker Kiri 9.bmp	Benar
Kanker Kiri 11.bmp	Benar
Kanker Kiri 12.bmp	Benar
Kanker Kiri 13.bmp	Benar
Kanker Kiri 14.bmp	<i>Salah</i>
Kanker Kiri 15.bmp	<i>Salah</i>
Kanker Kiri 16.bmp	Benar

Nama File	Tes
Normal Kiri 3 .bmp	Benar
Normal Kiri 4.bmp	<i>Salah</i>
Normal Kiri 5.bmp	Benar
Normal Kiri 6.bmp	Benar
Normal Kiri 7.bmp	Benar
Normal Kiri 8.bmp	<i>Salah</i>
Normal Kiri 9.bmp	Benar
Normal Kiri 10.bmp	Benar
Normal Kiri 11.bmp	Benar

Kanker Kiri 17.bmp	Benar
Kanker Kiri 18.bmp	Benar
Kanker Kiri 15.bmp	Benar
Kanker Kiri 16.bmp	Benar
Kanker Kiri 17.bmp	Benar
Kanker Kiri 18.bmp	Benar

Normal Kiri 12.bmp	Benar
Normal Kiri 13.bmp	Benar
Normal Kiri 15.bmp	Benar
Normal Kiri 16.bmp	Benar

**Tabel 4.20** Hasil pelatihan dan tes pengenalan dengan *variance value* pada payudara kanan

Nama File	Tes
Kanker Kanan 1.bmp	Benar
Kanker Kanan 2.bmp	Benar
Kanker Kanan 3.bmp	Benar
Kanker Kanan 4.bmp	Benar
Kanker Kanan 7.bmp	Benar
Kanker Kanan 9.bmp	<i>Salah</i>
Kanker Kanan 10.bmp	<i>Salah</i>

Nama File	Tes
Normal Kanan 1.bmp	Benar
Normal Kanan 2.bmp	Benar
Normal Kanan 3.bmp	Benar
Normal Kanan 4.bmp	Benar
Normal Kanan 5.bmp	Benar
Normal Kanan 6.bmp	Benar
Normal Kanan 7.bmp	Benar
Normal Kanan 8.bmp	Benar
Normal Kanan 10.bmp	<i>Salah</i>
Normal Kanan 11.bmp	Benar
Normal Kanan 12.bmp	Benar
Normal Kanan 13.bmp	Benar
Normal Kanan 14.bmp	<i>Salah</i>
Normal Kanan 15.bmp	Benar
Normal Kanan 17.bmp	Benar
Normal Kanan 18.bmp	Benar

**Tabel 4.21** Hasil pelatihan dan tes pengenalan dengan perbandingan *mean variance value* pada payudara kiri

Nama File	Tes
Kanker Kiri 2.bmp	Benar
Kanker Kiri 4.bmp	Benar
Kanker Kiri 5.bmp	Benar
Kanker Kiri 7.bmp	Benar
Kanker Kiri 8.bmp	Benar
Kanker Kiri 9.bmp	Benar
Kanker Kiri 10.bmp	Benar
Kanker Kiri 12.bmp	Benar
Kanker Kiri 14.bmp	Benar
Kanker Kiri 15.bmp	Benar
Kanker Kiri 18.bmp	Benar

Nama File	Tes
Normal Kiri 2.bmp	Benar
Normal Kiri 3.bmp	Benar
Normal Kiri 4.bmp	<i>Salah</i>
Normal Kiri 6.bmp	Benar
Normal Kiri 7.bmp	Benar
Normal Kiri 8.bmp	Benar
Normal Kiri 9.bmp	Benar
Normal Kiri 10.bmp	Benar
Normal Kiri 11.bmp	Benar
Normal Kiri 12.bmp	Benar
Normal Kiri 13.bmp	Benar
Normal Kiri 15.bmp	Benar
Normal Kiri 16.bmp	Benar
Normal Kiri 17.bmp	Benar

**Tabel 4.22** Hasil pelatihan dan tes pengenalan dengan perbandingan *mean variance value* pada payudara kanan

Nama File	Tes
Kanker Kanan 1.bmp	Benar
Kanker Kanan 2.bmp	Benar
Kanker Kanan 4.bmp	Benar
Kanker Kanan 6.bmp	Benar
Kanker Kanan 9.bmp	Benar
Kanker Kanan 10.bmp	Benar
Normal Kanan 1.bmp	Benar
Normal Kanan 2.bmp	Benar
Normal Kanan 3.bmp	Benar
Normal Kanan 4.bmp	Benar
Normal Kanan 5.bmp	Benar
Normal Kanan 6.bmp	Benar

Nama File	Tes
Normal Kanan 7.bmp	Benar
Normal Kanan 8.bmp	Benar
Normal Kanan 9.bmp	<i>Salah</i>
Normal Kanan 10.bmp	Benar
Normal Kanan 11.bmp	Benar
Normal Kanan 12.bmp	Benar
Normal Kanan 13.bmp	Benar
Normal Kanan 16.bmp	Benar
Normal Kanan 17.bmp	<i>Salah</i>
Normal Kanan 18.bmp	Benar

Kemudian hasil data pelatihan akan disimpan dalam dataset kemudian diujikan pada sampel data yang lain yang belum pernah dilatih. Dalam percobaan ini disediakan 30 data uji yang belum pernah dilatih.

#### 4.4.5 Hasil Pengenalan Kanker Payudara

Uji coba sistem dilakukan dengan mengujikan bobot masing data gambar dalam dataset, pola *output* akan dicocokkan satu persatu dengan tiap data gambar pada dataset. Tiap pencocokan pola dicari nilai kesalahan atau jaraknya. Sistem akan memutuskan hasil pengenalaannya pada data gambar dengan jarak target-*output* terkecil.

**Tabel 4.23** Hasil Pengujian data yang sudah dilatih

Payudara Kiri/kanan	Benar	Salah	Total	Keakuratan(%)
Mean Kiri	25	0	25	100
Variance Kiri	21	4	25	84
Mean/Variance Kiri	24	1	25	96
Mean Kanan	23	0	23	100
Variance kanan	19	4	23	82.60
Mean/Variance Kanan	21	2	23	91.30
Rata-Rata				92.32

Pertama akan diuji hasil pelatihan dengan data yang sudah dilatih. Hasil pengenalan dengan menggunakan *inputan* dari nilai *mean*, *variance* maupun perbandingan *mean variance* ada yang tidak dikenali benar, sehingga tingkat kebenaran untuk pengenalan *training dataset* mencapai 90.97%. Hasil pengujian data yang sudah dilatih dapat dilihat pada tabel 4.23.

**Tabel 4.24** Hasil Pengujian data yang belum di latih

Payudara Kiri/kanan	Benar	Salah	Total	Keakuratan(%)
Mean Kiri	15	2	17	88.24
Variance Kiri	13	4	17	76.47
Mean/Variance Kiri	14	3	17	82.35
Mean Kanan	11	2	13	84.62
Variance kanan	11	2	13	84.62
Mean/Variance Kanan	11	2	13	84.62

**Tabel 4.25** Hasil Pengujian data yang belum di latih secara keseluruhan berdasarkan data input yang digunakan

Data input	Benar	Salah	Total	Keakuratan(%)
Mean	26	4	30	86.67
Variance	24	6	30	80.00
mean/variance	25	5	30	83.33

Kemudian uji coba dilakukan pada data gambar yang belum dilatihkan atau *blind dataset*. Hasil pengujian dari data yang belum dilatihkan dapat dilihat pada Tabel 4.24. Dari 30 data yang belum dilatih, masing-masing untuk payudara kiri 17 data dan payudara kanan 13, dengan menggunakan *mean value* pada payudara kiri, tingkat akurasi pengenalan mencapai 88.24% sedangkan pada payudara kanan mencapai 84.62%. Jadi dari 30 data yang belum dilatih, dengan menggunakan *mean value* sebagai data input ada sebanyak 26 data gambar dikenali benar. Maka, kemampuan pengenalan *blind dataset* sebesar 86.67%.

Selanjutnya uji coba dilakukan menggunakan karakteristik data *input* dari nilai *variance*. Dari 30 data, masing-masing untuk payudara kiri 17 data dan payudara kanan 13, pada payudara kiri tingkat akurasi pengenalannya hanya mencapai 76.47%. Sedangkan pada payudara kanan mencapai 84.62%. Jadi dari 30 data yang belum dilatih, dengan menggunakan karakteristik data input *variance value* ada sebanyak 24 data gambar dikenali benar. Maka, kemampuan pengenalan *blind dataset* hanya sebesar 80%.

Uji coba yang terakhir dilakukan menggunakan karakteristik data *input* dengan perbandingan *mean variance*. Dari 30 data, masing-masing untuk payudara kiri 17 data dan payudara kanan 13, pada payudara kiri tingkat akurasi pengenalannya hanya mencapai 82.35%. Sedangkan pada payudara kanan mencapai 84.62%. Jadi dari 30 data yang belum dilatih, dengan menggunakan karakteristik data input *variance value* ada sebanyak 25 data gambar dikenali benar. Maka, kemampuan pengenalan *blind dataset* hanya sebesar 83.33%.

Hasil dari uji coba keseluruhan data gambar *mammogram* dilatih dan belum dilatih dengan menggunakan karakteristik data *inputan mean value*,(lihat Tabel 4.26) dari 78 data masing-masing untuk payudara kiri 42 data dan payudara kanan 36, dengan menggunakan *mean value* pada payudara kiri, tingkat akurasi pengenalamnya mencapai 95.24% sedangkan pada payudara kanan mencapai 94.44%. Jadi dari 78 data menghasilkan 74 data dikenali BENAR dan 4 data SALAH. Maka kemampuan pengenalan menggunakan data input dari nilai *mean* sebesar 94.87 %.

Sedangkan uji coba keseluruhan data gambar *mammogram* dilatih dan belum dilatih dengan menggunakan karakteristik data *inputan variance value*, (lihat Tabel 4.26) dari 78 data masing-masing untuk payudara kiri 42 data dan payudara kanan 36, dengan menggunakan *variance value* pada payudara kiri, tingkat akurasi pengenalamnya mencapai 80.95% sedangkan pada payudara kanan mencapai 83.33%. Jadi dari 78 data menghasilkan 64 data dikenali BENAR dan 14 data SALAH. Maka kemampuan pengenalan menggunakan data input dari nilai *variance* sebesar 82.05%.

Dan uji coba keseluruhan data gambar *mammogram* dilatih dan belum dilatih dengan menggunakan karakteristik data *input* dari perbandingan *mean variance value*, (lihat tabel 4.26) dari 78 data masing-masing untuk payudara kiri 42 data dan payudara kanan 36 data, dengan menggunakan perbandingan *mean variance value* pada payudara kiri, tingkat akurasi pengenalamnya mencapai 90.48% sedangkan pada payudara kanan mencapai 88.89%. Jadi dari 78 data menghasilkan 70 data dikenali BENAR dan 8 data SALAH. Maka kemampuan pengenalan menggunakan data input dari perbandingan *mean variance* sebesar 89.74 %.

**Tabel 4.26** Hasil Pengujian data keseluruhan

Payudara Kiri/kanan	Benar	Salah	Total	Keakuratan(%)
Mean Kiri	40	2	42	95.24
Variance Kiri	34	8	42	80.95
Mean/Variance Kiri	38	4	42	90.48
Mean Kanan	34	2	36	94.44
Variance kanan	30	6	36	83.33
Mean/Variance Kanan	32	4	36	88.89

**Tabel 4.27** Hasil pengujian data keseluruhan berdasarkan data input yang digunakan

Data input	Benar	Salah	Total	Keakuratan(%)
Mean	74	4	78	94.87
Variance	64	14	78	82.05
mean/variance	70	8	78	89.74
Rata-rata				89.03

Pengenalan dilakukan dengan memperhatikan jenis data yang dikenali yaitu kanker dan normal. Untuk memilih gambar *mammogram* yang baik yang digunakan sebagai data pelatihan yang nantinya di simpan di dataset sangat sulit. Perlu dilakukan beberapa percobaan untuk mendapatkan error yang minimal.

Dalam percobaan ini, sistem akan mencoba mengenali data *mammogram* yang kedua yaitu yang berasal dari RS.Onkologi. Dari data RS.onkologi terdapat 20 data *mammogram*. Data ini berbeda dengan data yang pertama. Maka dalam pengujian ini akan dilakukan beberapa kali percobaan. Pertama, data kedua akan diuji dengan menggunakan dataset yang ada pada data pertama. Ternyata data kedua dikenali kanker semua, ini dikarenakan nilai *output* yang dihasilkan dari 20 data kedua adalah sama sehingga menghasilkan pengenalan yang sama. Selain itu *threshold* yang digunakan tidak sesuai dengan data gambar yang kedua. Maka dilakukan percobaan pengujian dengan menggunakan *threshold* yang berbeda yaitu 50 dan hasil pelatihannya akan disimpan dalam data dataset sendiri yang kemudian digunakan untuk pengenalan. Ternyata dengan threshold yang lebih kecil menghasilkan pengenalan yang lebih baik.

**Tabel 4.28** Hasil Percobaan Data dari RS.Onkologi dengan Threshold=50

No.	Nama File	Jenis data	Tes Pengenalan
1	2 ny tansiok.bmp	DT Kanker	Benar
2	4 ny tansiok.bmp	DT Kanker	Salah
3	L MLO ny sri.bmp	DT Kanker	Benar
4	L MLO richwatun.bmp	DT Kanker	Benar
5	L MLO richwatun.bmp	DT Kanker	Benar

6	R CC richwatun.bmp	DT Kanker	Benar
7	R MLO richwatun.bmp	DT Kanker	Benar
8	Untitled-1 ny herawati.bmp	DT Kanker	Benar
9	Untitled-2 ny herawati.bmp	DT Kanker	Benar
10	Untitled-4 ny herawati.bmp	DT Kanker	Benar
11	L CC richwatun.bmp	DT Normal	Benar
12	R CC ny Emma.bmp	DT Normal	Benar
13	R CC ny sri.bmp	DT Normal	Benar
14	R MLO ny sri.bmp	DT Normal	Benar
15	L CC richwatun.bmp	DT Normal	Benar
16	1 ny tansiok.bmp	DU Kanker	Benar
17	L CC ny sri.bmp	DU Kanker	Benar
18	Untitled-3 ny herawati.bmp	DU Kanker	Benar
19	LCC ny Emma.bmp	DU Normal	Salah
20	R MLO ny Emma.bmp	DU Normal	Benar

DT: data training

DU: data uji

Dari percobaan pengujian diatas dapat disimpulkan bahwa data yang pertama dan data kedua memang berbeda. Apabila dalam sistem ini data pertama dan kedua diuji dalam dataset yang sama, maka pengenalan yang dihasilkan tidak sesuai. Seharusnya dataset yang digunakan dalam pengenalan harus sesuai dengan data yang digunakan untuk pelatihan dan pengenalan. Jadi untuk sistem ini data kedua tidak sesuai untuk mendeteksi kanker payudara.

#### 4.4.6 Uraian

Jaringan Syaraf Tiruan dengan lapisan tersembunyi tunggal yang dilengkapi propagasi balik dapat menyelesaikan pencapaian target yang sama dengan lapisan tersembunyi lebih dari satu. Tergantung pengaturan nilai parameter-paramernya. Diantaranya penentuan *learning rate*, banyak neuron, dan target kesalahan. Untuk mendapatkan arsitektur jaringan yang optimal harus disesuaikan

dengan banyak pola dan model pola masukan. Semakin banyak ragam pola, pencapaian target akan semakin sulit.

Banyaknya jumlah *neuron* tidak selalu menjamin optimalnya jaringan, jumlah *neuron* yang telalu besar dari jumlah pola masukan akan memperburuk proses pencapaian target *output*. Tiap *neuron* mempunyai fungsi penjumlahan, perkalian, perpangkatan, dan pembagian secara berulang. Jika jumlah *neuron* makin banyak maka waktu komputasi akan semakin lama. Selain itu tiap bobot optimal akan disimpan dalam dataset, semakin banyak *neuron*, maka bobot yang terbentuk akan semakin banyak. Maka, dalam penyimpanan diperlukan memori yang besar.

Hasil pengenalan untuk pendekstrian kanker payudara dengan menggunakan karakteristik data *inputan* dari *mean value* mencapai 94.87%, dengan *variance value* mencapai 82.05%, dan dengan perbandingan *mean variance value* mencapai 89.74%. Ini berarti kurang dari target yang diharapkan. Hal ini kemungkinan disebabkan karena hasil pengenalan pada data latih sendiri terdapat kesalahan. Sehingga mempengaruhi hasil pengenalan untuk data yang belum dilatih. Pengenalan data baru akan menghasilkan akurasi pengenalan yang baik apabila akurasi hasil uji pengenalan data latih mampu mencapai kebenaran 100%.

Dalam proses pendekstrian kanker payudara ini terdapat 3 tahap yang perlu diperhatikan, yaitu pemrosesan gambar data *input*, pelatihan, dan pencocokan. Walaupun dalam penelitian ini telah didapatkan struktur jaringan yang optimal, namun kegagalan dapat terjadi, kemungkinan disebabkan karena pemrosesan gambar *mammogram* masih kurang tepat.

Pada pemrosesan gambar *mammogram* hanya digunakan karakteristik data *input* dari nilai *mean*, *variance* dan perbandingan *mean variance*. Ternyata dari 3 karakteristik data *input* yang digunakan pada gambar *mammogram* ternyata data *input* menggunakan *mean* paling baik digunakan dalam pendekstrian kanker payudara.

# UNIVERSITAS BRAWIJAYA



94

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari hasil percobaan dan pembahasan dapat disimpulkan beberapa hal sebagai berikut:

1. Jaringan syaraf tiruan *backpropagation* yang sudah dilatih dengan data *mammogram*, dapat digunakan untuk mendeteksi kanker payudara.
2. Struktur Jaringan Saraf Tiruan yang berhasil disusun adalah Jaringan saraf *backpropagation* dengan satu lapisan tersembunyi. Terdiri atas lapisan input 600 neuron, lapisan tersembunyi 30 neuron dan lapisan output 1 neuron. Dengan nilai *learning rate* 2.0.
3. Hasil pengenalan dengan menggunakan tiga karakteristik data input yaitu dengan nilai *mean* menghasilkan 94.87%, dengan menggunakan *variance* 82.05% sedangkan menggunakan perbandingan *mean* dan *variance* menghasilkan 89.74%. Dari tiga karakteristik data *input* yang digunakan, karakteristik data *input* dengan menggunakan nilai *mean* menghasilkan pengenalan yang paling baik.

#### 5.2 Saran

Adapun saran penulis untuk pengembangan penelitian di bidang yang sama adalah:

1. Dalam skripsi ini *noise* diasumsikan statis atau tidak berpengaruh. Untuk percobaan berikutnya akan lebih baik jika gambar *mammogram* diolah sebelum dilakukan proses selanjutnya.
2. Perlu diperhatikan dalam proses prapengolahan gambar *mammogram*, dilakukan proses yang lebih kompleks pada gambar *mammogram* sehingga diperoleh ciri-ciri gambar *mammogram* yang yang lebih detail.
3. Untuk mendeteksian kanker payudara yang fleksibel, penelitian dapat dikembangkan dengan mendeteksian daerah kanker dan menggolongkan apakah termasuk kanker jinak atau ganas.

4. Aplikasi jaringan saraf tiruan sangat luas, memungkinkan penelitian pendekslan kanker payudara selanjutnya berdasar pada jaringan saraf tiruan dengan gabungan metode-metode yang ada. Seperti menggunakan metode PCA, algoritma genetik, *clustering* pola sebelum pembelajaran dan sebagainya. Kemudian dibandingkan agar memperoleh metode terbaik untuk mengenali kanker payudara.



## DAFTAR PUSTAKA

- A.B.Siemens-Elema dan S.P.S Division. 1999. *Mammography Applications*.Sweden.Siemens Medical Systems.  
<http://www.imaginis.com/breasthealth/mammography>.Tanggal akses: 24 Maret 2008.
- Achmad,B. dan Firdausy,K. 2005. *Teknik Pengolahan Citra Digital Menggunakan Delphi*. Ardi Publishing, Yogyakarta.
- AP.Nugraha dan AB.Mutiara. 2002. *Metode Ekstraksi Data Untuk Pengenalan Huruf dan Angka Tulisan tangan Dengan Menggunakan Jaringan Syaraf buatan Propagasi Balik*. Universitas Gunadarma.
- Bose, N.K and Liang, P. 1993. *Neural Network Fundamentals With Graphs, Algorithms, and Application*. McGraw Hill, Inc. Singapore.
- Dhaneswara,G. dan Moertini,SV.2004.*Jaringan Syaraf Tiruan Propagasi Balik Untuk Klasifikasi Data*.<http://www.unpac.ac.id>, tanggal akses :25 Maret 2008.
- Fu, Limin. 2002. *Neural Network in Computer Intelligence*. The MIT Press and McGraw Hill, Inc. Singapore.
- Gonzales, R, dan Paul W. 1987. *Digital Image Processing Second Edition*. Addison-Wesley Publishing Company.
- Gunawan. 2005. *Analisis Mammographic Microcalcifications Menggunakan Ciri-Ciri Struktur Citra Gray-Level*.  
<http://www.mipa.unej.ac.id/data/vol6no2/gunawan.pdf>.  
Tanggal akses : 12 maret 2008
- Hermawan,A. 2006. *Jaringan Saraf Tiruan: Teori dan Aplikasi*.Andi, Yogyakarta.
- Ibrahim Arif. *Penerapan Algoritma Kriptografi Kunci Publik Menggunakan Asitektur Jaringan Syaraf Tiruan*.ITB.

Kanker Payudara.<http://www.cancerhelps.com/kanker-payudara.htm>

Kiyan T. dan Yildirim T. 2004. *Breast Cancer Diagnosis Using Statistical neural Network*. University Turkey, Istambul.

Kristanto, A.2004. *Jaringan Syaraf Tiruan (Konsep dasar,Algoritma dan Aplikasi)*. Gaya Media, Yogyakarta.

L. Arbach, A.S dan J. M. Reinhardt, “*Classification of breast MRI lesions using a backpropagation neural network (BNN)*,” in IEEE, ISBI, (Washington DC., USA). Tanggal akses: 12 maret 2008.

Murni, A. 2004. *Diktat Kuliah PCD*. Universitas Indonesia, Jakarta.

Rassed E dan Awad G.M. *Neural Network Approach for mammography diagnosis using wavelet features*. Suez Canal Univ, Egypt.

Saputra A. 2002. *Penerapan Jaringan Syaraf Tiruan Untuk Pengenalan Pola Huruf*. Universitas Pancasila, Jakarta.

Siang,JJ. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab*. Andi, Yogyakarta.

Kusumadewi, S. 2003. *Artificial Intelligence (Teknik & Aplikasinya)*. Graha Ilmu, Yogyakarta.

Thangavel K.,Karnan M. *Ant Colony System for Segmentation and classification of Microcalcification in Mammograms*.College of Engineering&Technology, Tamil Nadu.

*University of South Florida Digital Mammography Home Page.*  
<http://marathon.csee.usf.edu/Mammography/Database.html>.Tanggal akses : 13 Februari 2008

<http://www.wartamedika.com/2008/02/mammografi.html>.Tanggal akses: 12 Maret 2008.