

**APLIKASI MODEL *RECURRENT NEURAL NETWORK* DAN  
MODEL *RECURRENT NEURO FUZZY* UNTUK KLASIFIKASI  
NODUL KANKER PARU DARI CITRA FOTO PARU**

**SKRIPSI**

Diajukan Kepada Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Negeri Yogyakarta  
untuk Memenuhi Sebagian Persyaratan guna Memperoleh Gelar Sarjana Sains



Oleh

Devi Nurdiyanti

NIM 10305141001

**PROGRAM STUDI MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI YOGYAKARTA**

**2014**

## PERSETUJUAN

Skripsi yang berjudul:

**APLIKASI MODEL *RECURRENT NEURAL NETWORK* DAN  
MODEL *RECURRENT NEURO FUZZY* UNTUK KLASIFIKASI NODUL  
KANKER PARU DARI CITRA FOTO PARU**

Oleh:

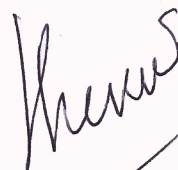
Devi Nurdiyanti

10305141001

Telah disetujui pada tanggal 12 Juni 2014  
untuk diujikan dihadapan dewan penguji skripsi

Prodi Matematika  
Jurusan Pendidikan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Negeri Yogyakarta

Menyetujui,  
Yogyakarta, 12 Juni 2014  
Dosen Pembimbing



Dr. Dhoriva Urwatul Wutsqa  
NIP. 19660331 199303 2 001

## HALAMAN PENGESAHAN

Skripsi yang berjudul:

**APLIKASI MODEL *RECURRENT NEURAL NETWORK* DAN  
MODEL *RECURRENT NEURO FUZZY* UNTUK KLASIFIKASI NODUL  
KANKER PARU DARI CITRA FOTO PARU**

Yang disusun oleh:

Nama : Devi Nurtiyasari  
NIM : 10305141001  
Prodi : Matematika

Skripsi ini telah diujikan di depan Dewan Penguji Skripsi pada tanggal 25 Juni 2014 dan dinyatakan LULUS.

DEWAN PENGUJI			
Nama	Jabatan	Tanda Tangan	Tanggal
<u>Dr. Dhoriva U.W</u> NIP. 19660331 199303 2 001	Ketua Penguji		30-06-2014
<u>Rosita K., M.Sc.</u> NIP. 19800707 200501 2 001	Sekretaris Penguji		30-06-2014
<u>Kuswari H., M.Kom.</u> NIP. 19760414 200501 2 002	Penguji Utama		27-06-2014
<u>Dr. Agus Maman Abadi</u> NIP. 19700828 199502 1 001	Penguji Pendamping		30-6-2014

Yogyakarta, 1 Juli 2014  
Fakultas Matematika dan  
Ilmu Pengetahuan Alam  
Universitas Negeri Yogyakarta  
Dekan,



Dr. Hartono  
NIP. 19620329 198702 1 001

## HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini, saya:

Nama : Devi Nurtiyasari

NIM : 10305141001

Program Studi : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

Judul Skripsi : APLIKASI MODEL *RECURRENT NEURAL NETWORK* DAN  
MODEL *RECURRENT NEURO FUZZY* UNTUK KLASIFIKASI  
NODUL KANKER PARU DARI CITRA FOTO PARU

Menyatakan bahwa skripsi ini benar-benar karya saya sendiri dan sepanjang pengetahuan saya, tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain, kecuali pada bagian-bagian tertentu yang diambil sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Apabila ternyata terbukti pernyataan saya ini tidak benar, maka sepenuhnya menjadi tanggung jawab saya, dan saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Yogyakarta, 12 Juni 2014  
Yang Menyatakan,



Devi Nurtiyasari  
NIM 10305141001



## **MOTTO**

*“Allah tempat meminta segala sesuatu”*

*(Qs. Al-Ikhlâs:2)*

*“Dan sungguh kelak Tuhanmu pasti memberikan karunia-Nya kepadamu, sehingga engkau menjadi puas”*

*(Qs. Ad –Duha:5)*

*“Karena sesungguhnya sesudah kesulitan itu ada kemudahan”*

*(Qs. Al-Insyirah : 5)*

*“Maka Nikmat Tuhanmu yang manakah yang engkau dustakan?”*

*(Qs.Ar-Rahman:13)*

## HALAMAN PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

Ibu dan Bapak serta adik-adikku. Terimakasih banyak atas doa, semangat, dan dukungan moril maupun materiil.

*Mas Yanuar, my best css and motivator ever.*

Kasih sayang, doa, nasehat, semangat dan kesabaranmu mendampingiku dalam menyelesaikan skripsi ini, *thank you so much.*

Reni dan Uswah, sahabat skripweet seperjuangan.

Banyak suka dan duka yang telah kita lewati bersama selama 8 bulan 12 hari proses penulisan skripsweet ini.

*You guys are amazing! Million thanks!!!*

Meita, Ratna, Nazil, Chandra, Rumi, Liha, Lina, Ahmad Arif, Aga, Teguh, Mas Anan, Mas Rizki, Marlin, Mbak Ayie, Mbak Rahma, Mas Iyos dan semua sahabat yang telah menjadi teman *refreshing* dan *sharing* selama penulisan skripsi ini. *Thank you guys so much!*

Om dan tante di JPC, sahabat Pikacu, dan Kaskuser (specially Kaskuser Regional Klaten dan Jogja) terimakasih telah memberikan informasi, saran dan bantuan dalam banyak hal selama proses penulisan skripsi ini.

Semua teman Matsub 2010, terimakasih atas kebersamaan dan bantuan kalian semua.

**APLIKASI MODEL *RECURRENT NEURAL NETWORK* DAN  
MODEL *RECURRENT NEURO FUZZY* UNTUK KLASIFIKASI NODUL  
KANKER PARU DARI CITRA FOTO PARU**

Oleh:  
Devi Nurtyasari  
NIM 10305141001

**ABSTRAK**

Model *Recurrent Neural Network* adalah model *neural network* yang mampu mengakomodasi *output* jaringan untuk menjadi *input* jaringan dalam rangka menghasilkan *output* jaringan berikutnya. Model *Recurrent Neuro Fuzzy* adalah model gabungan antara *Recurrent Neural Network* dan logika *fuzzy*. Penelitian ini bertujuan untuk menjelaskan prosedur, aplikasi serta tingkat keakuratan pemodelan *Recurrent Neural Network* dan pemodelan *Recurrent Neuro Fuzzy* untuk klasifikasi nodul kanker paru dari citra foto paru.

Pada dasarnya, langkah pemodelan *Recurrent Neural Network* dan *Recurrent Neuro Fuzzy* adalah mendefinisikan variabel *input* dan target, membagi data menjadi data *training* dan data *testing*, normalisasi data, perancangan model terbaik dan denormalisasi data. Variabel *input* yang digunakan adalah fitur hasil ekstraksi citra foto paru sedangkan target jaringan adalah keterangan kondisi dari citra foto paru yaitu berupa paru normal, tumor paru jinak, atau tumor paru ganas. Langkah ekstraksi citra diawali dengan melakukan transformasi citra yaitu dari citra foto paru asli (citra abu-abu) menjadi citra biner dan dilanjutkan dengan mengekstraksi citra hasil transformasi dengan metode *Gray Level Co-occurrence Matrix*. Langkah perancangan model *Recurrent Neural Network* terbaik adalah menentukan banyak neuron pada lapisan tersembunyi dan eliminasi *input* jaringan. Model *Recurrent Neural Network* menggunakan jaringan Elman dengan algoritma *backpropagation*. Adapun langkah perancangan model *Recurrent Neuro Fuzzy* terbaik diawali dengan langkah perancangan model *Recurrent Neural Network* terbaik dilanjutkan dengan langkah *clustering* data dengan metode *Fuzzy C-Means*, pembelajaran *Recurrent Neural Network* yang berhubungan dengan anteseden pada aturan inferensi *fuzzy*, pembelajaran *Recurrent Neural Network* yang berhubungan dengan konsekuen pada aturan inferensi *fuzzy*, dan penyederhanaan bagian konsekuen dengan melakukan eliminasi *input* serta mencari nilai koefisien konsekuen masing-masing *cluster* dengan metode *Least Square Estimator* (LSE). Aturan inferensi *fuzzy* menggunakan model Sugeno orde satu.

Klasifikasi nodul kanker paru menggunakan model *Recurrent Neural Network* memberikan hasil yang lebih baik daripada model *Recurrent Neuro Fuzzy*. Nilai sensitivitas, spesifisitas, dan akurasi model *Recurrent Neural Network* secara berurutan adalah 94%, 56%, dan 81,33% untuk data *training* serta 80%, 40%, dan 64% untuk data *testing*.

**Kata Kunci :** *recurrent neural network*, *recurrent neuro fuzzy*, klasifikasi, nodul kanker paru, citra foto paru, ekstraksi citra

## KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan nikmat serta hidayah-Nya terutama nikmat kesempatan dan kesehatan sehingga penulis dapat menyelesaikan tugas akhir skripsi ini. Skripsi yang berjudul “Aplikasi Model *Recurrent Neural Network* dan Model *Recurrent Neuro Fuzzy* untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru” disusun untuk memenuhi salah satu syarat kelulusan guna meraih gelar Sarjana Sains pada Program Studi Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta.

Penyelesaian skripsi ini tidak akan terselesaikan tanpa bantuan, bimbingan, saran dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Dr. Hartono, M.Si, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta yang telah memberikan kelancaran pelayanan dalam urusan akademik.
2. Dr. Sugiman, M.Si, selaku Ketua Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta yang telah memberikan kelancaran pelayanan dalam urusan akademik.
3. Dr. Agus Maman Abadi, M.Si, selaku Ketua Program Studi Matematika Universitas Negeri Yogyakarta serta Penasehat Akademik yang telah memberikan arahan, motivasi, serta dukungan akademik kepada penulis.



4. Dr. Dhoriva U. W selaku dosen pembimbing yang sangat sabar memberikan bimbingan, masukan serta arahan dalam penyusunan skripsi ini.
5. Seluruh dosen Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta yang telah memberikan ilmu kepada penulis.
6. Seluruh pihak yang telah memberikan dukungan, bantuan dan motivasi kepada penulis.

Penulis menyadari adanya ketidakteelitian, kekurangan dan kesalahan dalam penulisan tugas akhir skripsi ini. Oleh karena itu, penulis menerima kritik dan saran yang bersifat membangun. Semoga penulisan tugas akhir ini dapat bermanfaat bagi pembaca dan pihak yang terkait.

Yogyakarta, 12 Juni 2014  
Penulis



Devi Nurdiyanti

## DAFTAR ISI

PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN.....	iv
MOTTO.....	v
HALAMAN PERSEMBAHAN.....	vi
ABSTRAK.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	5
C. Tujuan Penelitian.....	5
D. Manfaat Penelitian.....	6
BAB II KAJIAN TEORI.....	7
A. Kanker Paru.....	7
1. Penyebab.....	7
2. Jenis.....	8
3. Tahapan.....	9
3. Deteksi.....	10
B. Pengolahan Citra.....	11
C. <i>Neural Network</i> (NN).....	20
1. Fungsi Aktivasi.....	20
2. Arsitektur.....	23
3. Algoritma Pembelajaran.....	25
D. Himpunan Klasik.....	27
E. Himpunan <i>Fuzzy</i> .....	28
1. Pengertian Himpunan <i>Fuzzy</i> .....	28
2. Fungsi Keanggotaan ( <i>Membership Function</i> ).....	30
3. Operator Himpunan <i>Fuzzy</i> .....	31
F. Logika <i>Fuzzy</i> .....	32
G. Model <i>Fuzzy</i> .....	33
H. Model <i>Neuro Fuzzy</i> .....	37
I. <i>Mean Square Error</i> (MSE).....	42
J. Sensitivitas, Spesifisitas, dan Akurasi.....	42
1. Sensitivitas.....	43
2. Spesifisitas.....	44
3. Akurasi.....	44
BAB III PEMBAHASAN.....	45
A. Prosedur Pemodelan <i>Recurrent Neural Network</i> (RNN) untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru.....	45
1. Algoritma Pembelajaran <i>Backpropagation</i> pada RNN Jaringan Elman.....	47

2. Membangun Jaringan Elman pada MatlabR2010a.....	55
3. Prosedur Pemodelan RNN.....	56
B. Prosedur Pemodelan <i>Recurrent Neuro Fuzzy</i> (RNF) untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru .....	59
C. Aplikasi Model <i>Recurrent Neural Network</i> (RNN) dan Model <i>Recurrent Neuro Fuzzy</i> (RNF) untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru .....	65
1. Aplikasi Model <i>Recurrent Neural Network</i> (RNN) Klasifikasi Nodul Kanker Paru .....	67
2. Aplikasi Model <i>Recurrent Neuro Fuzzy</i> (RNF) Klasifikasi Nodul Kanker Paru.....	73
3. Perbandingan Ketepatan Klasifikasi Nodul Kanker Paru Menggunakan Model RNN dan Model RNF.....	86
BAB IV PENUTUP.....	88
A. Kesimpulan .....	88
B. Saran .....	90
DAFTAR PUSTAKA.....	92
LAMPIRAN .....	96

## DAFTAR TABEL

<b>Tabel 2.1</b>	Hasil Uji Diagnosa Disajikan dalam Tabel 2x2 .....	43
<b>Tabel 3.1</b>	Nilai MSE Pemilihan Banyak Neuron .....	68
<b>Tabel 3.2</b>	Nilai MSE Eliminasi <i>Input</i> .....	69
<b>Tabel 3.3</b>	Hitungan Manual $y(1)_k$ .....	71
<b>Tabel 3.4</b>	Hitungan Manual $y_k$ .....	72
<b>Tabel 3.5</b>	Nilai Keanggotaan .....	75
<b>Tabel 3.6</b>	MSE Neuron Hasil Pembelajaran $RNN_1$ .....	76
<b>Tabel 3.7</b>	MSE Neuron Hasil Pembelajaran $RNN_2$ .....	77
<b>Tabel 3.8</b>	MSE Neuron Hasil Pembelajaran $RNN_3$ .....	78
<b>Tabel 3.9</b>	MSE Eliminasi <i>Input</i> $RNN_1$ .....	79
<b>Tabel 3.10</b>	Koefisien Konsekuen $RNN_1$ .....	80
<b>Tabel 3.11</b>	MSE Eliminasi <i>Input</i> $RNN_2$ .....	81
<b>Tabel 3.12</b>	Koefisien Konsekuen $RNN_2$ .....	82
<b>Tabel 3.13</b>	MSE Eliminasi <i>Input</i> $RNN_3$ .....	83
<b>Tabel 3.14</b>	Koefisien Konsekuen $RNN_3$ .....	84
<b>Tabel 3.15</b>	Nilai $\mu_{As}(t)$ dan $\mu_s(t)$ Data Pertama .....	85
<b>Tabel 3.16</b>	Nilai $\mu_{As}(t)$ dan $\mu_s(t)$ Data Kedua .....	86
<b>Tabel 3.17</b>	Sensitivitas, Spesifisitas, dan Akurasi .....	87



## DAFTAR GAMBAR

<b>Gambar 2.1(a)</b>	Citra Foto untuk Paru Normal .....	12
<b>Gambar 2.1(b)</b>	Citra Foto untuk Tumor Paru Jinak .....	12
<b>Gambar 2.1(c)</b>	Citra Foto untuk Tumor Paru Ganas .....	12
<b>Gambar 2.2(a)</b>	Citra Paru Abu-abu .....	13
<b>Gambar 2.2(b)</b>	Citra Paru Biner .....	13
<b>Gambar 2.3</b>	Tampilan GUI .....	14
<b>Gambar 2.4</b>	Fungsi Identitas .....	21
<b>Gambar 2.5</b>	Fungsi Undak Biner .....	21
<b>Gambar 2.6</b>	Fungsi Bipolar .....	22
<b>Gambar 2.7</b>	Struktur NN Lapisan Tunggal .....	23
<b>Gambar 2.8</b>	Struktur NN dengan Banyak Lapisan .....	25
<b>Gambar 2.9</b>	Kurva Segitiga .....	30
<b>Gambar 2.10</b>	Proses Inferensi <i>Fuzzy</i> .....	35
<b>Gambar 2.11</b>	Struktur <i>Neuro Fuzzy</i> .....	39
<b>Gambar 3.1</b>	Jaringan Elman .....	46

## DAFTAR LAMPIRAN

<b>Lampiran 3.1</b>	Citra Foto Toraks (Asli dan Biner) .....	96
<b>Lampiran 3.2</b>	Hasil Ekstraksi (GLCM) .....	102
<b>Lampiran 3.3</b>	M-File Ekstraksi Fitur Citra .....	106
<b>Lampiran 3.4</b>	Hasil Ekstraksi (GLCM) Data <i>Training</i> .....	113
<b>Lampiran 3.5</b>	Hasil Ekstraksi (GLCM) Data <i>Testing</i> .....	116
<b>Lampiran 3.6</b>	Hasil Normalisasi Data <i>Training</i> dan Data <i>Testing</i> .....	118
<b>Lampiran 3.7</b>	<i>Target, Output</i> RNN dan <i>Error</i> .....	122
<b>Lampiran 3.8</b>	Nilai $m_{it}^s$ dan Pembagian <i>Cluster</i> Data.....	125
<b>Lampiran 3.9</b>	Nilai Keanggotaan $\mu_{As}(t)$ Data pada Bagian Antesenden.....	128
<b>Lampiran 3.10</b>	Data <i>Cluster</i> 1 .....	131
<b>Lampiran 3.11</b>	Data <i>Cluster</i> 2 .....	131
<b>Lampiran 3.12</b>	Data <i>Cluster</i> 3 .....	132
<b>Lampiran 3.13</b>	Hasil Penyederhanaan <i>Cluster</i> 1 .....	133
<b>Lampiran 3.14</b>	Hasil Penyederhanaan <i>Cluster</i> 2 .....	134
<b>Lampiran 3.15</b>	Hasil Penyederhanaan <i>Cluster</i> 3 .....	135
<b>Lampiran 3.16</b>	Nilai $\mu_s(t)$ Data <i>Training</i> .....	136
<b>Lampiran 3.17</b>	Nilai $\mu_s(t)$ Data <i>Testing</i> .....	138
<b>Lampiran 3.18</b>	<i>Target, Output</i> dan <i>Error</i> Model RNF .....	139
<b>Lampiran 3.19</b>	<i>Performance Measure</i> Model RNF .....	142
<b>Lampiran 3.20</b>	<i>Performance Measure</i> Model RNN .....	144
<b>Lampiran 3.21</b>	M-File Proses RNN dan RNF .....	147
<b>Lampiran 3.22</b>	Hitungan Manual <i>Output</i> RNN .....	158
<b>Lampiran 3.23</b>	Percobaan Lainnya .....	163

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Kanker paru adalah jenis kanker yang tumbuh di jaringan paru-paru yang berperan penting dalam proses pernapasan. Kejadian kanker paru terus meningkat dan telah menjadi masalah kesehatan global. Jenis kanker yang menjadi penyebab kematian terbanyak di dunia adalah kanker paru yaitu mencapai 1,61 juta kematian pertahun (12,7%), kanker payudara yaitu mencapai 1,31 juta kematian pertahun (10,9%), dan kanker usus besar yaitu mencapai 1,23 juta kematian pertahun (9,7%). (Varalakshmi, 2013:1).

Penyebab utama kanker paru adalah asap rokok karena mengandung lebih dari 4.000 zat kimia, dimana 63 jenis diantaranya bersifat karsinogen dan beracun (Perhimpunan Dokter Paru Indonesia, 2003:2). *American Cancer Society* (2013) mengemukakan bahwa 80% kasus kanker paru disebabkan oleh rokok (perokok aktif) sedangkan perokok pasif beresiko 20% sampai 30% untuk terkena kanker paru. Penyebab kanker paru lainnya menurut *American Lung Association* (2010:55) adalah radiasi dan polusi udara.

Terdapat dua jenis kanker paru, yaitu: *Small Cell Lung Cancer* (SCLC) dan *Non Small Cell Lung Cancer* (NSCLC). (Varalakshmi, 2013:1). Tahapan perkembangan SCLC ada dua yaitu tahap terbatas dan tahap ekstensif, sedangkan tahapan perkembangan NSCLC ada 6 yaitu tahap tersembunyi, stadium 0, stadium I, stadium II, stadium III dan stadium IV (*Global Bioscience*, 2013).

Pemeriksaan awal yang dapat dilakukan untuk mendeteksi kanker paru adalah dengan pemeriksaan radiologi paru yaitu melalui foto paru. Foto paru atau

sering disebut *Chest X-Ray* (CXR) adalah suatu proyeksi radiografi dari paru (Perhimpunan Dokter Paru Indonesia, 2003:5). Citra foto paru sering digunakan untuk *screening* (deteksi dini) penyakit paru. Citra dari foto paru akan memberikan hasil yang berbeda antara paru-paru yang sehat dan yang tidak sehat. Adanya nodul di paru-paru pada citra foto paru menunjukkan bahwa paru-paru tidak sehat, akan tetapi nodul ini tidak serta merta menjadi indikasi kanker paru karena nodul dapat disebabkan oleh penyakit paru lain seperti pneumonia atau tuberculosis (Udeshani, *et al*, 2011:1). Nodul yang terdeteksi pada paru-paru dikategorikan menjadi dua yaitu *non cancerous nodule* (*benign* atau tumor jinak) dan *cancerous nodule* (*malignant* atau tumor ganas). (*Japanese Society of Radiology Technology*, 1997). Tumor jinak bukanlah kanker karena tumor jenis ini bisa diangkat (dihilangkan) sedangkan tumor ganas adalah kanker karena dapat menyebar dan membahayakan jaringan serta organ yang ada di dekat tumor tersebut (*Agency for Toxic Substances and Disease Registry*, 2013:1).

Pendeteksian nodul pada citra foto paru sangat diperlukan karena hasil deteksi menentukan langkah lanjutan yang harus diambil dalam menangani pasien. Deteksi dini yang tepat melalui citra foto paru dapat meningkatkan angka harapan hidup pasien secara signifikan (Udhesani, *et al*, 2011:1). Penelitian untuk mengklasifikasikan kanker paru biasanya dilakukan dengan mengekstraksi citra foto paru terlebih dahulu. Fitur hasil ekstraksi citra foto paru kemudian diklasifikasi dengan berbagai model seperti model *neural network* (NN), model *fuzzy*, dan model *neuro fuzzy* (NF).



NN adalah suatu sistem yang memiliki karakteristik tertentu mirip dengan jaringan syaraf biologi (Fausett, 1994:3). NN pertama kali diperkenalkan pada tahun 1943 oleh *neurophysiologist* Warren McCulloch dan *logician* Walter Pitts. NN merupakan jaringan untuk memodelkan cara kerja sistem syaraf manusia (otak). Pemodelannya didasari oleh kemampuan otak manusia dalam mengorganisasi sel-sel penyusunnya (*neuron*), sehingga mampu melakukan pemrosesan informasi. Klasifikasi kanker paru menggunakan model NN telah dilakukan diantaranya klasifikasi sel kanker paru menggunakan *artificial neural network* yang dilakukan oleh Zhou, *et al* (2002) dan Abdullah & Shaharum (2012), serta deteksi kanker paru menggunakan algoritma *backpropagation* yang dilakukan oleh Deviasari, *et al*.

Logika *fuzzy* (*fuzzy logic*) merupakan representasi pengetahuan yang dikonstruksikan dengan *IF-THEN rules*. Logika *fuzzy* pertama kali diperkenalkan oleh Lotfi Zadeh. Logika *fuzzy* merupakan perluasan dari logika konvensional Boolean yang telah diperluas untuk menangani konsep kebenaran parsial, yaitu nilai kebenaran yang terletak antara kebenaran absolut (direpresentasikan dengan nilai 1) dan kesalahan absolut (direpresentasikan dengan nilai 0). Penelitian mengenai klasifikasi kanker paru menggunakan model *fuzzy* telah dilakukan diantaranya penelitian oleh Jang, *et al* (2013) yang bertujuan untuk mendeteksi kanker paru menggunakan *Optimal Fuzzy Rule*, penelitian oleh Sivakumar & Chandrasekar (2013) yang bertujuan untuk mendeteksi kanker paru menggunakan *Fuzzy Clustering dan Support Vector Machines*, serta penelitian oleh Jaffar, *et al*

(2011) yang bertujuan untuk segmentasi paru dari gambar CT Scans menggunakan *Genetic Fuzzy*.

NF merupakan penggabungan antara NN dan logika *fuzzy*. NF adalah model yang sedang banyak digunakan dalam penelitian. NF mengadaptasi nilai parameter *fuzzy* dengan fungsi pembelajaran NN. NF dirancang untuk merealisasikan proses penalaran *fuzzy*, dimana bobot-bobot yang terhubung pada NN terhubung dengan parameter-parameter penalaran *fuzzy*. NF bertujuan untuk mengurangi kelemahan dari masing-masing sistem (NN dan *fuzzy*) serta menggabungkan kelebihan dari masing-masing sistem tersebut. Penelitian klasifikasi kanker paru menggunakan model NF telah dilakukan diantaranya penelitian oleh Malathi dan Santra yang bertujuan untuk mendiagnosis kanker paru menggunakan logika *neuro fuzzy* dan penelitian oleh Al-Daoud (2010) yang bertujuan untuk mendiagnosis kanker paru dengan menggunakan model *Adaptive Neuro Fuzzy Inference System (ANFIS)*.

Penelitian klasifikasi kanker paru dengan model NN dan NF biasanya menggunakan jaringan *feedforward*. Model NN dan NF dapat dikembangkan dengan jaringan *recurrent*. Jaringan *recurrent* adalah jaringan yang memiliki *feedback link* (umpan balik), sehingga jaringan ini mampu mengakomodasi *output* jaringan untuk menjadi *input* jaringan dalam rangka menghasilkan *output* jaringan berikutnya. NN dengan jaringan *recurrent* disebut sebagai *Recurrent Neural Network (RNN)* dan NF dengan jaringan *recurrent* disebut sebagai *Recurrent Neuro Fuzzy (RNF)*. Perhitungan yang lebih kompleks pada RNN dan RNF diharapkan dapat memberikan hasil klasifikasi yang baik.

Berdasarkan penelusuran terhadap penelitian terdahulu mengenai klasifikasi nodul kanker paru, penulis hendak melakukan kajian yang berjudul “Aplikasi Model *Recurrent Neural Network* dan Model *Recurrent Neuro Fuzzy* untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru”. Penelitian ini diharapkan dapat memberikan alternatif klasifikasi nodul kanker paru yang lebih akurat.

### **B. Rumusan Masalah**

Berdasarkan latar belakang tersebut maka permasalahan dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana prosedur pemodelan *Recurrent Neural Network* untuk klasifikasi nodul kanker paru dari citra foto paru?
2. Bagaimana prosedur pemodelan *Recurrent Neuro Fuzzy* untuk klasifikasi nodul kanker paru dari citra foto paru?
3. Bagaimana aplikasi model *Recurrent Neural Network* dan model *Recurrent Neuro Fuzzy* untuk klasifikasi nodul kanker paru dari citra foto paru?

### **C. Tujuan Penulisan**

Adapun tujuan dari penulisan ini adalah untuk:

1. Menjelaskan prosedur pemodelan *Recurrent Neural Network* untuk klasifikasi nodul kanker paru dari citra foto paru.
2. Menjelaskan prosedur pemodelan *Recurrent Neuro Fuzzy* untuk klasifikasi nodul kanker paru dari citra foto paru.
3. Mendeskripsikan aplikasi model *Recurrent Neural Network* dan model *Recurrent Neuro Fuzzy* untuk klasifikasi nodul kanker paru dari citra foto paru.

#### **D. Manfaat**

Hasil penelitian ini diharapkan memiliki manfaat sebagai berikut:

1. Menambah wawasan tentang model *Recurrent Neural Network* dan model *Recurrent Neuro Fuzzy* dalam aplikasi pengklasifikasian nodul kanker paru.
2. Sebagai alternatif klasifikasi nodul kanker paru dari citra foto paru.
3. Sebagai bahan informasi untuk penelitian tentang klasifikasi selanjutnya.
4. Menambah koleksi bahan pustaka yang bermanfaat bagi UNY pada umumnya, dan mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam pada khususnya.

## **BAB II KAJIAN TEORI**

### **A. Kanker Paru**

Kanker adalah pertumbuhan tak terkontrol dan penyebaran sel abnormal dalam tubuh (*Agency for Toxic Substances and Disease Registry*, 2013:1). Kanker paru adalah salah satu penyakit yang paling mematikan di dunia. Kanker paru memerlukan penanganan dan tindakan yang tepat dan terarah. Buruknya prognosis penyakit ini berkaitan dengan jaranganya penderita datang ke dokter ketika penyakitnya masih berada pada stadium awal (*Perhimpunan Dokter Paru Indonesia*, 2003:1). Zhao, *et al* (2003:2) menyatakan hanya 15% kasus kanker paru yang ditemukan sejak dini (pada stadium awal). Deteksi dini dan penanganan yang tepat pada pasien yang menderita kanker paru diharapkan mampu meningkatkan angka harapan hidup dan mengurangi angka kematian yang diakibatkan kanker paru.

#### **1. Penyebab**

Kejadian kanker paru sangat berkaitan dengan merokok. Asap rokok menjadi penyebab utama kanker paru karena mengandung lebih dari 4.000 zat kimia, dimana 63 jenis diantaranya bersifat karsinogen dan beracun (*Perhimpunan Dokter Paru Indonesia*, 2003:2). Semakin banyak seseorang terpapar asap rokok, semakin besar resiko terkena kanker paru. *American Cancer Society* (2013) mengemukakan bahwa 80% kasus kanker paru disebabkan oleh rokok (perokok aktif) sedangkan perokok pasif beresiko 20% sampai 30% untuk terkena kanker paru. Penyebab kanker paru lainnya adalah radiasi dan polusi udara (*American Lung Association*, 2010:55).

## 2. Jenis

Terdapat dua jenis kanker paru, yaitu (Varalakhshmi, 2013:1):

### a. *Small Cell Lung Cancer* (SCLC)

SCLC adalah jenis kanker paru yang tumbuh lebih cepat daripada jenis kanker yang bukan sel kecil (*non-small cell*), akan tetapi SCLC lebih bisa dikendalikan pertumbuhannya dengan kemoterapi. Sekitar 20% kasus kanker paru adalah SCLC.

### b. *Non Small Cell Lung Cancer* (NSCLC)

Sekitar 75%-80% kasus kanker paru adalah NSCLC. Ada 3 tipe NSCLC, yaitu :

#### 1) Karsinoma skuamosa

Kanker tipe ini adalah jenis kanker paru paling umum. Karsinoma skuamosa berkembang dalam sel yang menggarisi saluran udara. Jenis kanker ini seringkali disebabkan karena rokok.

#### 2) Adenokarsinoma

Kanker tipe ini berkembang dari sel-sel yang memproduksi lendir (dahak) pada permukaan saluran udara (*airways*).

#### 3) Karsinoma sel besar

Bentuk sel kanker ini dibawah mikroskop sesuai namanya yaitu sel-sel bundar besar. Tipe ini sering disebut juga karsinoma tidak berdiferensiasi.

Pasien SCLC yang mendapat terapi kemoterapi memiliki rata-rata harapan hidup sebesar 12 sampai 16 bulan dan sebesar 15% sampai 20% mengalami

harapan hidup dan bebas dari penyakit sedangkan pasien NSCLC tipe kanker sel skuamosa dapat bertahan hidup lebih lama dibandingkan dengan mereka yang mendertia adenokarsinoma dan karsinoma sel besar (Otto, 2005:256).

### **3. Tahapan**

Tahapan kanker paru berdasarkan *Global Bioscience* (2013) adalah sebagai berikut.

#### **a. Tahap perkembangan SCLC**

- 1) Tahap terbatas, yaitu kanker yang hanya ditemukan pada satu bagian paru-paru saja dan pada jaringan disekitarnya.
- 2) Tahap ekstensif, yaitu kanker yang ditemukan pada jaringan dada di luar paru-paru tempat asalnya atau kanker ditemukan pada organ-organ tubuh yang jauh.

#### **b. Tahap perkembangan NSCLC**

- 1) Tahap tersembunyi merupakan tahap ditemukannya sel kanker pada dahak (sputum) pasien di dalam sampel air saat bronkoskopi, tetapi tidak terlihat adanya tumor di paru-paru.
- 2) Stadium 0 merupakan tahap ditemukannya sel-sel kanker hanya pada lapisan terdalam paru-paru dan tidak bersifat invasif.
- 3) Stadium I merupakan tahap kanker yang hanya ditemukan pada paru-paru dan belum menyebar ke kelenjar getah bening sekitarnya.
- 4) Stadium II merupakan tahap kanker yang ditemukan pada paru-paru dan kelenjar getah bening di dekatnya.

- 5) Stadium III merupakan tahap kanker yang telah menyebar ke daerah di sekitarnya, seperti dinding dada, diafragma, pembuluh besar atau kelenjar getah bening di sisi yang sama atau pun sisi berlawanan dari tumor tersebut.
- 6) Stadium IV merupakan tahap kanker yang ditemukan lebih dari satu lobus paru. Sel-sel kanker telah menyebar juga ke organ tubuh lainnya, misalnya ke otak, kelenjar adrenalin, hati, dan tulang.

#### **4. Deteksi**

Deteksi kanker paru dapat dilakukan dengan beberapa cara, yaitu melalui biopsy, *computed tomography* (CT), dan *magnetic resonance imaging* (MRI). (Agency for Toxic Substances and Disease Registry, 2013:4). Biopsy adalah pemeriksaan mikroskopik menggunakan contoh jaringan tubuh. CT atau pemeriksaan radiologi adalah pemeriksaan menggunakan *X-rays* untuk menghasilkan citra bagian tubuh tertentu. MRI adalah pemeriksaan menggunakan medan magnet dan gelombang frekuensi radio tanpa *X-rays* atau radioaktif.

Secara umum, deteksi kanker paru dilakukan melalui CT atau pemeriksaan radiologi. Citra foto paru yang diperoleh dari hasil CT adalah suatu proyeksi radiografi dari paru. Citra ini sering digunakan untuk *screening* (deteksi dini) penyakit paru. Citra dari foto paru akan memberikan hasil yang berbeda antara paru-paru yang sehat dan yang tidak sehat. Adanya nodul di paru-paru pada citra foto paru menunjukkan bahwa paru-paru tidak sehat, akan tetapi nodul ini tidak serta merta menjadi indikasi kanker paru karena nodul dapat disebabkan oleh penyakit paru lain seperti pneumonia atau tuberculosis (Udeshani, *et al*, 2011:1).



Nodul yang terdeteksi pada paru-paru dikategorikan menjadi dua yaitu *non cancerous nodule* (*benign* atau tumor jinak) dan *cancerous nodule* (*malignant* atau tumor ganas). (*Japanese Society of Radiology Technology*, 1997).

Tumor jinak bukanlah kanker karena tumor jenis ini bisa diangkat (dihilangkan) dan tidak kembali lagi (*Agency for Toxic Substances and Disease Registry*, 2013:1). Sel dari tumor jinak hanya menyerang satu tempat dan tidak menyebar ke organ tubuh lainnya. Artinya, jika tumor jinak ada pada jaringan paru maka tumor ini tidak akan menyerang selain organ paru.

Tumor ganas adalah kanker (*Agency for Toxic Substances and Disease Registry*, 2013:1). Sel kanker menyebar dan membahayakan jaringan dan organ yang ada di sekitar tumor tersebut. Tumor ganas pada jaringan paru sangat berbahaya jika tidak dikendalikan pertumbuhannya karena sel kanker dapat berkembang dan menyebar keluar organ paru.

## **B. Pengolahan Citra**

Pengolahan citra merujuk pada transformasi dari suatu citra (*input*) menjadi citra lainnya (*output*). (Darma Putra, 2010:19). Citra foto adalah sebuah gambar yang dicetak dari hasil pemotretan kamera dengan perekaman secara fotografi (Felix Wicaksono, 2009:1). Citra foto paru dapat digolongkan menjadi tiga kelompok, yaitu citra foto paru normal (Gambar 2.1 (a)), citra foto tumor paru jinak (Gambar 2.1 (b)), dan citra foto tumor paru ganas (Gambar 2.1 (c)).



(a)



(b)



(c)

**Gambar 2.1** Citra Foto untuk Paru Normal (a), Tumor Paru Jinak (b), dan Tumor Paru Ganas (c) (*Japanese Society of Radiology Technology, 1997*)

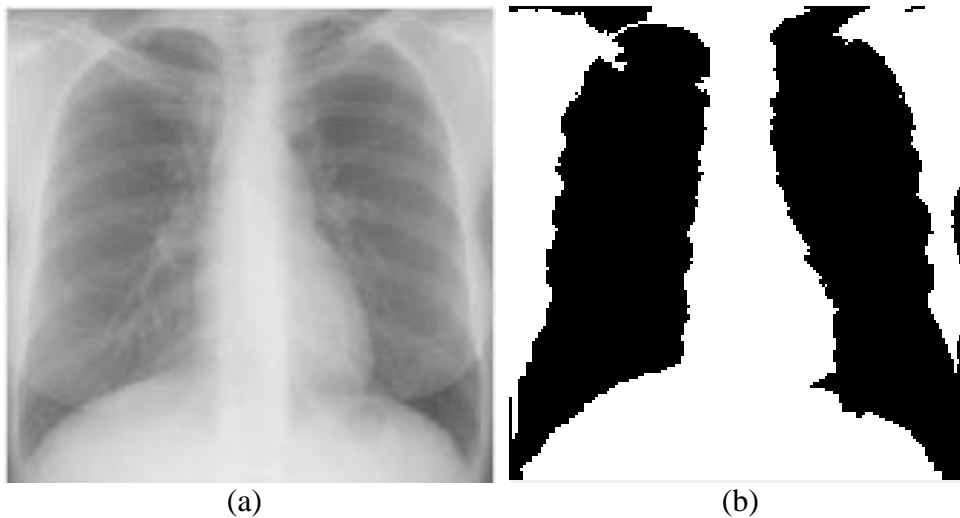
Citra foto paru tersebut dapat diekstraksi sehingga diperoleh fitur yang akan digunakan sebagai *input* dalam proses klasifikasi. Ekstraksi citra merupakan suatu teknik pengambilan sifat-sifat khusus (fitur) dari sebuah citra. Langkah pertama yang dilakukan sebelum proses ekstraksi citra adalah mengubah citra foto paru (citra abu-abu) menjadi citra biner. Setiap piksel dalam citra abu-abu merupakan bayangan abu-abu yang memiliki nilai intensitas 0 (hitam) sampai 255

(putih), sedangkan untuk citra biner, setiap pikselnya hitam atau putih saja (Sianipar, 2013:200).

Tranformasi dari citra abu-abu (Gambar 2.2 (a)) ke citra biner (Gambar 2.2 (b)) dilakukan menggunakan MATLABR2010a dengan perintah sebagai berikut:

```
a = imread('N1.jpg'); (membaca gambar)
```

```
b = im2bw(a); (tranformasi dari citra abu-abu ke citra biner)
```



**Gambar 2.2** Citra Paru Abu-abu (a) dan Citra Paru Biner (b)

Langkah selanjutnya setelah transformasi citra adalah ekstraksi citra menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Ekstraksi citra yang dilakukan dengan metode GLCM mampu menghasilkan 14 fitur ekstraksi. Fungsi bawaan MATLABR2010a hanya mampu memberikan 5 fitur saja yaitu *entropy*, *contrast*, *correlation*, *energy*, dan *homogeneity*. Fungsi bawaan MATLABR2010a tersebut adalah sebagai berikut:

```
a = imread('N1.jpg'); (membaca gambar)
```

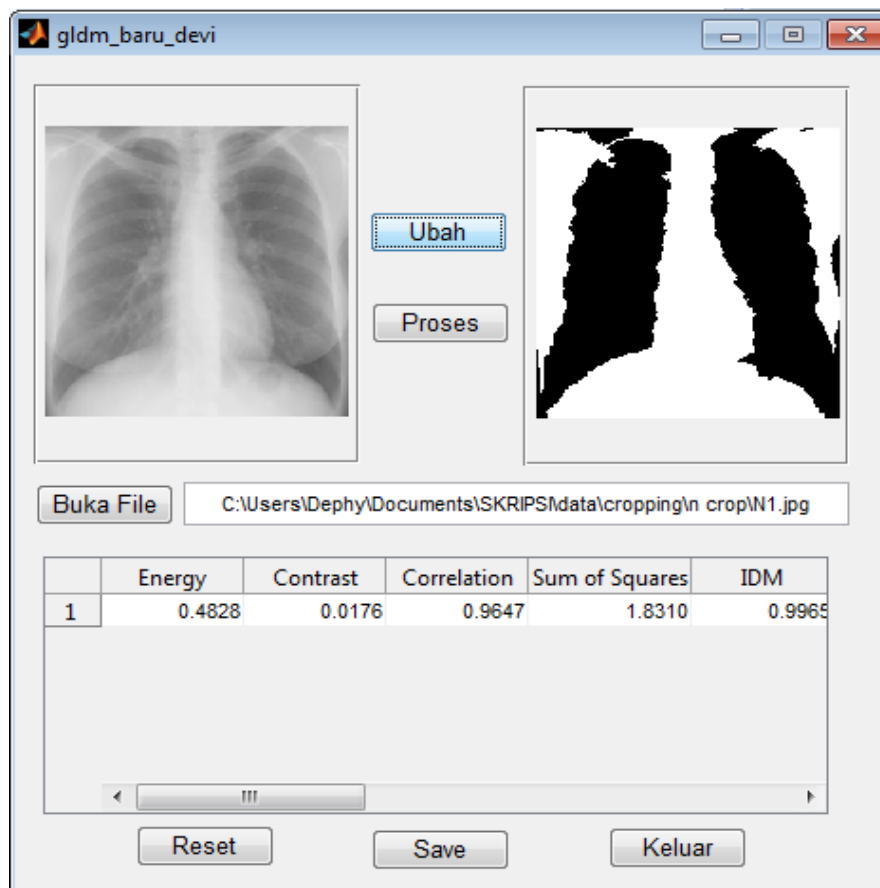
```
b = im2bw(a); (tranformasi dari citra abu-abu ke citra biner)
```

```
glcm = graycomatrix(b); (GLCM)
```

$stats = graycoprops(glcm, 'all');$  (mendapatkan nilai *contrast*, *correlation*, *energy*, dan *homogeneity*)

$c = entropy(b);$  (mendapatkan nilai entropi)

Fungsi bawaan MATLABR2010a tersebut dapat dikembangkan untuk mendapatkan 9 fitur lainnya dengan kode tertentu. *Graphical User Interface* (GUI) coba dibangun untuk dapat digunakan dalam proses ekstraksi. Tampilan GUI untuk ekstraksi dari citra abu-abu menjadi citra biner dan hasil ekstraksinya adalah sebagai berikut (Gambar 2.3).



**Gambar 2.3** Tampilan GUI

Fitur-fitur hasil ekstraksi yang diperoleh adalah *energy*, *contrast*, *correlation*, *sum of square variance*, *Inverse Difference Moment (IDM)*, *sum*

*average, sum entropy, sum variance, entropy, difference variance, difference entropy, maximum probability, homogeneity dan dissimilarity.*

### 1. *Energy*

*Energy* adalah jumlah kuadrat elemen di GLCM, jika nilainya 1 berarti merupakan citra konstan (Sharma & Mukharjee, 2013:331). Rumus *energy* (E) adalah sebagai berikut (Mohanaiah, *et al*, 2013:2):

$$E = \sum_i \sum_j \{p(i, j)\}^2 \quad (2.1)$$

dengan,

$p(i, j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

### 2. *Contrast*

Nilai *contrast* adalah ukuran intensitas kontras antara suatu piksel dan piksel lainnya dari keseluruhan citra (Sharma & Mukharjee, 2013:331). Rumus kontras (C) adalah sebagai berikut (Kalas, 2010:20):

$$C = \sum_i \sum_j p(i, j)(i - j)^2 \quad (2.2)$$

dengan,

$p(i, j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

### 3. *Correlation*

*Correlation* mengukur ketergantungan linear level abu-abu (*grey level*) antara piksel pada suatu posisi terhadap posisi lainnya dan dirumuskan sebagai berikut (Soh, 1999:781):

$$Cr = \sum_i \sum_j \frac{\{(ij)p(i, j)\} - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.3)$$

dengan,

$p(i, j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ ,

$$\begin{aligned}\mu_x &= \sum_i \sum_j \{(i)p(i, j)\}, \\ \mu_y &= \sum_i \sum_j \{(j)p(i, j)\}, \\ \sigma_x &= \sum_i \sum_j \{(i - \mu_x)^2 p(i, j)\}, \\ \sigma_y &= \sum_i \sum_j \{(i - \mu_y)^2 p(i, j)\}.\end{aligned}$$

#### 4. *Sum of square variance*

Rumus *sum of square variance* (SSV) adalah sebagai berikut (Anami & Burkpalli, 2009:11):

$$SSV = \sum_i \sum_j p(i, j)(i - \mu)^2 \quad (2.4)$$

dengan,

$p(i, j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ ,

$\mu$  = rata-rata  $\sum_i \sum_j p(i, j)$ .

#### 5. *Inverse Difference Moment (IDM)*

IDM adalah ukuran dari homogenitas lokal. Nilai IDM tinggi ketika level abu-abu (*grey level*) lokal seragam dan invers dari GLCM tinggi. (Mohanaiah, *et al*, 2013:2). Rumus IDM adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$IDM = \sum_i \sum_j \frac{p(i, j)}{1+(i-j)^2} \quad (2.5)$$

dengan,

$p(i, j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

#### 6. *Sum average*

Rumus *sum average* (SA) adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$SA = -\sum_k \{(k)(p_{x+y(k)})\} \quad (2.6)$$

dengan,

$$p_{x+y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j); i + j = k; k = 2, 3, \dots, 2N_g,$$

$N_g$  = banyak level abu-abu yang diperoleh dari citra.

#### 7. *Sum entropy*

Rumus *sum entropy* (SE) adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$SE = - \sum_k p_{x+y(k)} \log\{p_{x+y(k)}\} \quad (2.7)$$

dengan,

$$p_{x+y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j); i + j = k; k = 2, 3, \dots, 2N_g,$$

$N_g$  = banyak level abu-abu yang diperoleh dari citra.

#### 8. *Sum variance*

*Sum variance* menunjukkan seberapa banyak level keabu-abuan yang bervariasi dari nilai rata-rata (Sharma & Mukharjee, 2013:331). Rumus *Sum variance* (SV) adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$SV = \sum_k (i - SE)^2 p_{x+y(k)} \quad (2.8)$$

dengan,

$SE$  = jumlah entropi,

$$p_{x+y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j); i + j = k; k = 2, 3, \dots, 2N_g,$$

$N_g$  = banyak level abu-abu yang diperoleh dari citra.

#### 9. *Entropy*

*Entropy* adalah suatu ukuran keacakan. *Entropy* menunjukkan jumlah informasi dari sebuah citra yang dibutuhkan untuk mengkompres citra

(Mohanaiah, *et al*, 2013:2). Rumus *entropy* (EN) dari suatu citra adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$EN = - \sum_i \sum_j p(i, j) \log_2 \{p(i, j)\} \quad (2.9)$$

dengan,

$p(i, j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

#### 10. *Difference variance*

Rumus *difference variance* (DV) adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$DV = \text{varians dari } p_{x-y(k)} \quad (2.10)$$

dengan,

$$p_{x-y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) ; |i - j| = k ; k = 0, 1, 2, \dots, (N_g - 1),$$

$N_g$  = banyak level abu-abu yang diperoleh dari citra.

#### 11. *Difference entropy*

Rumus *difference entropy* (DE) adalah sebagai berikut (Harralick, *et al*, 1973:619):

$$DE = - \sum_k (p_{x-y(k)}) \{ \log(p_{x-y(k)}) \} \quad (2.11)$$

dengan,

$$p_{x-y(k)} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) ; |i - j| = k ; k = 0, 1, 2, \dots, (N_g - 1),$$

$N_g$  = banyak level abu-abu yang diperoleh dari citra.



## 12. *Maximum probability*

*Maximum probability* menunjukkan tingkat abu-abu (*grey level*) yang memenuhi relasi pada persamaan entropi dan dirumuskan sebagai berikut (Anami & Burkpalli, 2009:11):

$$MP = \max_{i,j} \{p(i,j)\} \quad (2.12)$$

dengan,

$p(i,j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

## 13. *Homogeneity*

*Homogeneity* memberikan nilai yang merupakan ukuran kedekatan dari distribusi elemen di GLCM ke diagonal GLCM dan dirumuskan sebagai berikut (Sharma & Mukharjee, 2013:331):

$$H = \sum_i \sum_j \frac{p(i,j)}{1+|i-j|} \quad (2.13)$$

dengan,

$p(i,j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

## 14. *Dissimilarity*

*Dissimilarity* (D) menunjukkan perbedaan tiap piksel dan dirumuskan sebagai berikut (Anami & Burkpalli, 2009:11):

$$D = \sum_i \sum_j p(i,j) |i - j| \quad (2.14)$$

dengan,

$p(i,j)$  = piksel di lokasi baris ke- $i$  dan kolom ke- $j$ .

### **C. Neural Network (NN)**

NN merupakan jaringan untuk memodelkan cara kerja otak manusia. NN adalah sebuah arsitektur yang terdiri dari banyak neuron yang bekerja bersama untuk memberikan respon pada *input* (Yeung, *et al*, 2010:3). Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian NN (Siang, 2005:23). Terdapat 3 lapisan penyusun NN yaitu lapisan *input* (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan *output* (*output layer*). Lapisan *input* menerima *input* dari luar yang berupa gambaran dari suatu permasalahan. Lapisan tersembunyi menghubungkan lapisan *input* dan lapisan *output*. Keluaran dari lapisan *output* merupakan hasil NN terhadap permasalahan yang diterima lapisan *input*.

Konsep dasar NN adalah menghubungkan antara variabel-variabel *input* dengan satu atau lebih variabel *output*. Lapisan *input* dan lapisan *output* dihubungkan melalui lapisan tersembunyi dan ditransformasi menggunakan fungsi aktivasi.

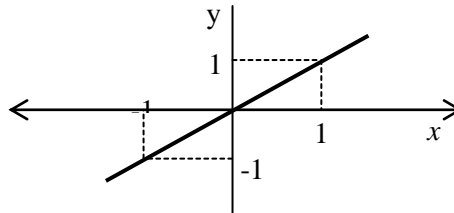
#### **1. Fungsi Aktivasi**

Fungsi aktivasi akan menentukan *output* suatu unit (mengubah sinyal *input* menjadi sinyal *output*) yang akan dikirim ke unit lain (Fausett,1994: 17). Fungsi aktivasi digunakan sebagai pengaktif NN, artinya fungsi aktivasi digunakan untuk mengaktifkan setiap neuron yang ada pada jaringan. Terdapat banyak fungsi aktivasi yang dapat digunakan sebagai pengaktif NN, yaitu sebagai berikut.

a. Fungsi Identitas (*Linier*)

Pada fungsi identitas, nilai *output* yang dihasilkan sama dengan nilai *inputnya* (Gambar 2.4). Fungsi linear dirumuskan sebagai berikut (Yeung, *et al*, 2010:2):

$$y = f(x) = x \quad (2.15)$$



**Gambar 2.4** Fungsi Identitas

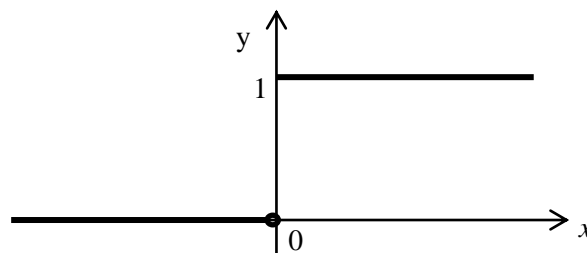
Pada MatlabR2010a, sintaks untuk fungsi aktivasi ini adalah *purelin*.

b. Fungsi Undak Biner (*Hard Limit*)

Fungsi undak biner digunakan untuk mengkonversikan *input* suatu variabel yang bernilai kontinu ke suatu *output* biner. Fungsi undak biner dirumuskan sebagai berikut (Yeung, *et al*, 2010:2):

$$y = f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{lainnya} \end{cases} \quad (2.16)$$

Grafik fungsi undak biner tampak pada Gambar 2.5 berikut:



**Gambar 2.5** Fungsi Undak Biner

Pada MatlabR2010a, sintaks untuk fungsi aktivasi ini adalah *hardlim*.

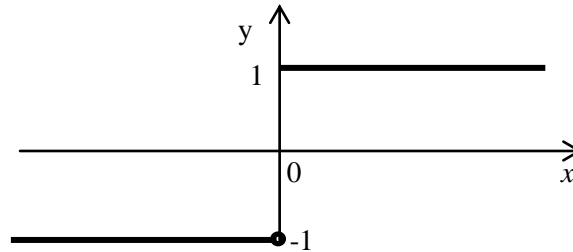
c. Fungsi Bipolar (*Symmetric Hard Limit*)

Fungsi bipolar mirip dengan fungsi undak biner, perbedaannya terletak pada nilai *output* yang dihasilkan. Nilai *output* fungsi bipolar

berupa 1 dan -1. Fungsi bipolar dirumuskan sebagai berikut (Khalil & Al-Kazzaz, 2009:17):

$$y = f(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (2.17)$$

Grafik fungsi bipolar dapat dilihat pada Gambar 2.6 berikut :



**Gambar 2.6** Fungsi Bipolar

Pada MatlabR2010a, sintaks untuk fungsi aktivasi ini adalah *hardlims*.

#### d. Fungsi Sigmoid Biner

Fungsi sigmoid biner memiliki sifat non-linier sehingga sangat baik untuk menyelesaikan permasalahan kompleks dan bersifat non-linier. Nilai fungsinya terletak antara 0 dan 1. Fungsi sigmoid biner dirumuskan sebagai berikut (Fausett, 1994:18):

$$y = f(x) = \frac{1}{1+e^{-x}} \quad (2.18)$$

Pada MatlabR2010a, sintaks untuk fungsi aktivasi ini adalah *logsig*.

#### e. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar mirip dengan fungsi sigmoid biner, perbedaannya terletak pada rentang nilai *outputnya*. Rentang nilai *output* fungsi sigmoid bipolar adalah -1 sampai 1. Fungsi sigmoid bipolar dirumuskan sebagai berikut (Fausett, 1994:19):

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad (2.19)$$

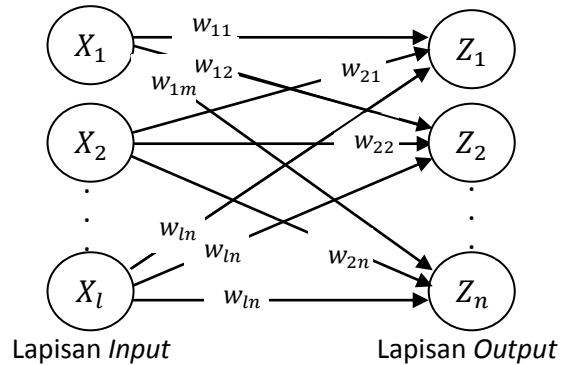
Pada MatlabR2010a, sintaks untuk fungsi aktivasi ini adalah *tansig*.

## 2. Arsitektur

Pengaturan neuron yang tersusun dalam lapisan NN disebut dengan arsitektur jaringan. Terdapat beberapa arsitektur NN, yaitu sebagai berikut.

### a. NN dengan Lapisan Tunggal (*Single Layered Network*)

NN dengan lapisan tunggal (Gambar 2.7) hanya terdiri dari *input layer* yang langsung terhubung ke *output layer* dan tidak ada lapisan yang tersembunyi. NN dengan lapisan tunggal adalah jaringan umpan maju atau *feedforward* (Haykin, 1998:21).



**Gambar 2.7** Struktur NN Lapisan Tunggal

Pada Gambar 2.7, terdapat sebanyak  $l$  neuron pada lapisan *input* dan sebanyak  $n$  neuron pada lapisan *output*, sedangkan  $w_{ik}$  adalah bobot yang menghubungkan lapisan *input* dengan lapisan *output* dengan  $i = 1, 2, \dots, l$  dan  $k = 1, 2, \dots, n$ . Sinyal *input* terbobot ( $z_{in_k}$ ) ke neuron  $Z_k$  adalah jumlah dari bobot sinyal neuron  $X_1, X_2, \dots, X_l$  sehingga

$$z_{in_k} = \sum_{i=1}^l \sum_{k=1}^n x_i w_{ik} \quad (2.20)$$

Aktivasi  $z_k$  dari neuron  $Z_k$  dilakukan dengan suatu fungsi aktivasi yaitu  $z_k = f(z_{in_k})$ .

NN dengan lapisan tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini membuat perkembangan NN terhenti

sekitar tahun 1970-an dan dapat ditanggulangi dengan menambahkan satu atau beberapa lapisan tersembunyi (*hidden layer*) di antara lapisan masukan (*input*) dan lapisan keluaran (*output*). (Siang, 2005:97).

b. NN dengan Banyak Lapisan (*Multi Layer Network*)

Berbeda dengan NN dengan lapisan tunggal, pada NN dengan banyak lapisan terdapat minimal satu lapisan tersembunyi (Gambar 2.8). Lapisan *input* pada NN dengan banyak lapisan akan memberikan sinyal pada lapisan tersembunyi dan kemudian lapisan tersembunyi akan meneruskan sinyal tersebut langsung pada lapisan *output* atau lapisan tersembunyi lainnya jika terdapat lebih dari satu lapisan tersembunyi pada jaringan tersebut.

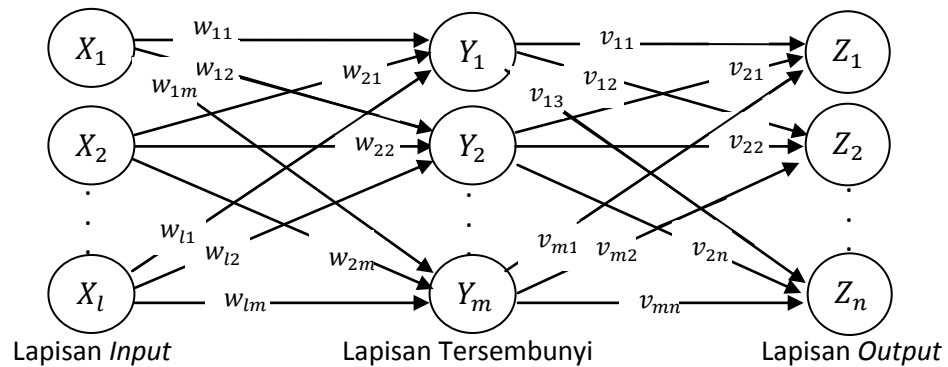
Pada Gambar 2.8, terdapat sebanyak  $l$  neuron pada lapisan *input*, sebanyak  $m$  neuron pada lapisan tersembunyi dan sebanyak  $n$  neuron pada lapisan *output*. Bobot yang menghubungkan lapisan *input* dengan lapisan tersembunyi adalah  $w_{ik}$  dimana  $i = 1, 2, \dots, l$  dan  $k = 1, 2, \dots, m$ . Adapun bobot yang menghubungkan lapisan tersembunyi dan lapisan *output* adalah  $v_{kr}$  dimana  $k = 1, 2, \dots, m$  dan  $r = 1, 2, \dots, n$ . Sinyal *input* terbobot ( $y_{in_k}$ ) ke neuron  $Y_k$  adalah jumlah dari bobot sinyal neuron  $X_1, X_2, \dots, X_l$  sehingga

$$y_{in_k} = \sum_{i=1}^l \sum_{k=1}^m x_i w_{ik} \quad (2.21)$$

Aktivasi  $y_k$  dari neuron  $Y_k$  dilakukan dengan suatu fungsi aktivasi yaitu  $y_k = f(y_{in_k})$ . Sinyal *input* terbobot ( $z_{in_r}$ ) ke neuron  $Z_r$  adalah jumlah dari bobot sinyal neuron  $Y_1, Y_2, \dots, Y_m$  sehingga

$$z_{in_r} = \sum_{k=1}^m \sum_{r=1}^n y_k v_{kr} \quad (2.22)$$

Aktivasi  $z_r$  dari neuron  $Z_r$  dilakukan dengan suatu fungsi aktivasi yaitu  $z_r = f(z_{in_r})$ .



**Gambar 2.8** Struktur NN dengan Banyak Lapisan

c. NN dengan Lapisan Kompetitif

NN dengan lapisan kompetitif memiliki bentuk yang berbeda dari NN lapisan tunggal dan banyak lapisan. Pada NN dengan lapisan kompetitif neuron-neuronnya dapat saling dihubungkan. Salah satu NN lapisan kompetitif adalah *Recurrent Neural Network* (RNN), yaitu NN yang memiliki koneksi umpan balik (*feedback link*). RNN adalah sebuah arsitektur NN dimana aliran proses yang dilalui *input* akan melalui banyak lapis dikarenakan aliran prosesnya melakukan timbal balik yang memungkinkan sebuah proses dilakukan berulang kali pada *layer* yang berbeda. RNN meliputi lapisan *input*, lapisan *output* dan lapisan tersembunyi seperti pada jaringan-jaringan yang lainnya, tetapi pada RNN paling sedikit terdapat 1 lapisan umpan balik.

**3. Algoritma Pembelajaran**

Pembelajaran dalam NN didefinisikan sebagai suatu proses dimana parameter-parameter bebas NN diadaptasi melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada (Suyanto, 2008:74). Proses

pembelajaran merupakan bagian penting dari konsep *neural network*. Proses pembelajaran bertujuan untuk melakukan pengaturan terhadap bobot yang ada pada *neural network*, sehingga diperoleh bobot akhir yang tepat sesuai dengan pola data yang dilatih (Sri Kusumadewi dan Sri Hartati, 2010:84). Pada proses pembelajaran akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu. Nilai bobot akan naik jika informasi yang diberikan ke suatu neuron mampu tersampaikan ke neuron yang lain. Sebaliknya, nilai bobot akan berkurang jika informasi yang diberikan ke suatu neuron tidak tersampaikan ke neuron lainnya.

Terdapat 2 metode pembelajaran *neural network*, yaitu pembelajaran terawasi (*supervised learning*) dan pembelajaran tak terawasi (*unsupervised learning*).

a. Pembelajaran Terawasi (*Supervised Learning*)

Metode pembelajaran pada *neural network* disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Tujuan pembelajaran terawasi adalah untuk memprediksi satu atau lebih variabel target dari satu atau lebih variabel *input* (Sarle, 1994:6). Pada proses pembelajaran, satu pola *input* akan diberikan ke suatu neuron pada lapisan *input*. Selanjutnya pola akan dirambatkan sepanjang *neural network* hingga sampai ke neuron pada lapisan *output*. Lapisan *output* akan membangkitkan pola *output* yang akan dicocokkan dengan pola *output* targetnya. *Error* muncul apabila terdapat perbedaan antara pola *output* hasil pembelajaran dengan pola target sehingga diperlukan pembelajaran lagi.



#### b. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

Pembelajaran tak terawasi tidak memerlukan target *output* dan jaringan dapat melakukan *training* sendiri untuk mengekstrak fitur dari variabel independen (Sarle, 1994:6). Pada metode ini, tidak dapat ditentukan hasil *outputnya*. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu sesuai dengan nilai *input* yang diberikan. Tujuan pembelajaran ini adalah untuk mengelompokkan unit-unit yang hampir sama ke dalam suatu area tertentu.

Adapun kelebihan NN adalah mampu melakukan pembelajaran, dapat melakukan generalisasi, dan model cenderung stabil, sedangkan kelemahan NN adalah ketidakmampuan menginterpretasi secara fungsional dan kesulitan untuk menentukan banyak neuron serta banyak layer pada lapisan tersembunyi (Vieira, *et al*, 2003:2).

#### **D. Himpunan Klasik**

Suatu himpunan didefinisikan sebagai kumpulan objek yang memiliki karakteristik tertentu, misalkan himpunan bilangan asli kurang dari sepuluh, himpunan warna pelangi, dan himpunan hewan menyusui. Himpunan klasik (*crisp set*) adalah himpunan yang membedakan anggota dan bukan anggota dengan batasan yang jelas (Ross, 2010:26). Misalkan pada suatu himpunan  $A$  hanya terdapat dua kemungkinan yaitu menjadi anggota  $A$  atau bukan anggota  $A$ . Nilai yang menunjukkan tingkat keanggotaan suatu elemen  $x$  dalam suatu himpunan  $A$  disebut nilai keanggotaan atau derajat keanggotaan, dinotasikan dengan  $\mu_A(x)$ . Nilai keanggotaan himpunan klasik menggunakan logika biner 0 atau 1 untuk

menyatakan keanggotaannya. Apabila  $x$  merupakan anggota  $A$  maka nilai  $\mu_A(x) = 1$  dan apabila  $x$  bukan anggota  $A$  maka nilai  $\mu_A(x) = 0$ .

**Contoh 2.1** Jika  $S = \{1,2,3,4,5\}$  adalah suatu himpunan semesta dan  $A = \{1,3,4\}$ , maka dapat dikatakan bahwa:

- a. Nilai Keanggotaan 1 pada himpunan  $A$ ,  $\mu_A(1) = 1$ , karena  $1 \in A$ ,
- b. Nilai Keanggotaan 2 pada himpunan  $A$ ,  $\mu_A(2) = 0$ , karena  $2 \notin A$ ,
- c. Nilai Keanggotaan 3 pada himpunan  $A$ ,  $\mu_A(3) = 1$ , karena  $3 \in A$ .
- d. Nilai Keanggotaan 4 pada himpunan  $A$ ,  $\mu_A(4) = 1$ , karena  $4 \in A$ .
- e. Nilai Keanggotaan 5 pada himpunan  $A$ ,  $\mu_A(5) = 0$ , karena  $5 \notin A$ .

### **E. Himpunan *Fuzzy***

Himpunan *fuzzy* merupakan perluasan dari himpunan klasik. Teori himpunan *fuzzy* diperkenalkan oleh Lotfi Zadeh A. pada tahun 1985. Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval  $[0,1]$ .

#### **1. Pengertian Himpunan *Fuzzy***

Misalkan  $U$  adalah himpunan semesta dan  $u$  adalah anggota  $U$ . Suatu himpunan *fuzzy*  $A$  dalam  $U$  didefinisikan sebagai suatu fungsi keanggotaan  $\mu_A[x]$ , yang memetakan setiap objek di  $U$  menjadi suatu nilai real dalam interval  $[0,1]$ .

**Contoh 2.2** Misalkan  $U$  menyatakan ketinggian permukaan air di suatu bendungan (dalam m) dan  $A$  adalah status ketinggian dari permukaan air di bendungan tersebut. Anggota  $U$  adalah sebagai berikut:

$$U = \{0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4\}$$

Status ketinggian permukaan air dapat di definisikan dengan fungsi keanggotaan  $\mu_A(u)$  sebagai berikut:

$\mu_A(u)$	0	0,1	0,3	0,5	0,6	0,7	0,8	0,9	1
$u$	0,8	1,0	1,2	1,4	1,6	1,8	2,0	2,2	2,4

Ketinggian air 1 m berarti normal (0), ketinggian air 1 sampai 2.2 m berarti siaga, dan ketinggian air 2.4 berarti waspada (1).

Himpunan *fuzzy* dapat direpresentasikan dengan 2 cara yaitu dalam bentuk Zadeh dan himpunan pasangan berurutan.

a. Himpunan *fuzzy* bentuk Zadeh dinotasikan sebagai:

$$\begin{aligned}
 A &= \mu_A(u_1)/u_1 + \mu_A(u_2)/u_2 + \dots + \mu_A(u_n)/u_n \\
 &= \sum_{i=1}^n \mu_A(u_i)/u_i
 \end{aligned}
 \tag{2.23}$$

untuk himpunan semesta diskret, dan

$$\begin{aligned}
 A &= \mu_A(u_1)/u_1 + \mu_A(u_2)/u_2 + \dots + \mu_A(u_n)/u_n \\
 &= \int_U \mu_A(u_i)/u_i
 \end{aligned}
 \tag{2.24}$$

untuk himpunan semesta kontinu. Simbol “ / ” bukan merupakan operasi pembagian, begitu juga simbol “+” bukan merupakan operasi penjumlahan. Menggunakan Contoh 2.2, himpunan *fuzzy* A direpresentasikan dengan bentuk Zadeh sebagai:

$$\begin{aligned}
 A &= 0/0.8 + 0.1/1.0 + 0.3/1.2 + 0.5/1.4 + 0.6/1.6 + 0.7/1.8 \\
 &\quad + 0.8/2.0 + 0.9/2.2 + 1/2.4
 \end{aligned}$$

b. Himpunan *fuzzy* direpresentasikan sebagai pasangan berurutan dengan elemen pertama nama elemen dan elemen kedua nilai keanggotaan, dan dinotasikan sebagai berikut (Zimmermann, 1991:11-12):

$$A = \{(u, \mu_A(u)) | u \in U\}
 \tag{2.25}$$

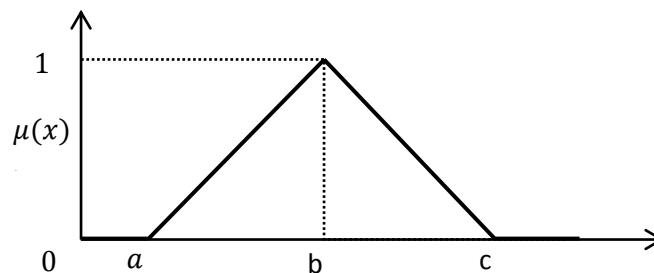
Menggunakan Contoh 2.2, himpunan *fuzzy* A direpresentasikan dengan himpunan pasangan berurutan sebagai berikut:

$$A = \{(0.8, 0), (1.0, 0.1), (1.2, 0.3), \dots, (2.0, 0.8), (2.2, 0.9), (2.4, 1)\}$$

## 2. Fungsi Keanggotaan (*Membership Function*)

Seperti halnya menentukan peluang dalam teori peluang, fungsi keanggotaan juga hanya dapat diperkirakan karena keterbatasan kognitif (Li, *et al*, 2001:38). Fungsi yang mengawankan setiap  $u \in U$  dengan bilangan real pada interval  $[0,1]$  pada himpunan *fuzzy* A dinamakan fungsi keanggotaan (Ibrahim, 2004:31-32). Terdapat beberapa fungsi keanggotaan yang dapat digunakan untuk merepresentasikan himpunan *fuzzy*, diantaranya adalah representasi linear (naik dan turun), representasi kurva segitiga, representasi kurva trapesium, representasi kurva bentuk lonceng (*Bell Curved*), dan representasi kurva-S.

Fungsi keanggotaan yang paling sering digunakan adalah fungsi keanggotaan representasi kurva segitiga (Barua, *et al*, 2014:1). Representasi kurva segitiga tidak lebih dari kumpulan 3 titik yang membentuk segitiga (Mandal, *et al*, 2012:294). Pada representasi kurva segitiga (Gambar 2.9) hanya terdapat satu nilai  $x$  yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika  $x = b$ .



Gambar 2.9 Kurva Segitiga

Fungsi keanggotaannya dirumuskan sebagai berikut:

$$\mu(x) = \begin{cases} 0; & x < a \text{ atau } x > c \\ (x-a)/(b-a); & a \leq x \leq b \\ (c-x)/(c-b); & b \leq x \leq c \end{cases} \quad (2.26)$$

### 3. Operator Himpunan Fuzzy

Seperti pada himpunan klasik, dalam himpunan *fuzzy* juga dikenal beberapa operator antara lain persamaan, ketermuatan, komplemen, gabungan dan irisan.

**Definisi 2.2** (Wang, 1997:29)

Persamaan, ketermuatan (*containment*), komplemen, gabungan, dan irisan dari dua himpunan *fuzzy* A dan B didefinisikan sebagai berikut:

- a. A dan B dikatakan sama jika hanya jika  $\mu_A(u) = \mu_B(u)$  untuk setiap  $u \in U$ ,
- b. B dikatakan memuat A, dinotasikan  $A \subset B$ , jika hanya jika  $\mu_A(u) \leq \mu_B(u)$  untuk setiap  $u \in U$ ,
- c. Komplemen A adalah himpunan *fuzzy*  $\bar{A}$  dalam  $U$  yang fungsi keanggotaannya didefinisikan sebagai berikut :

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \quad (2.27)$$

- d. Gabungan dari A dan B adalah himpunan *fuzzy* dalam  $U$ , dinotasikan  $A \cup B$ , dengan fungsi keanggotaan sebagai berikut:

$$\mu_{A \cup B}(u) = \max[\mu_A(u), \mu_B(u)] \quad (2.28)$$

- e. Irisan dari A dan B adalah himpunan *fuzzy* dalam  $U$ , dinotasikan  $A \cap B$ , dengan fungsi keanggotaan

$$\mu_{A \cap B}(u) = \min[\mu_A(u), \mu_B(u)] \quad (2.29)$$

Gabungan disebut juga sebagai operator AND dan irisan disebut juga sebagai operator OR, dan komplemen disebut juga sebagai operator NOT.

## **F. Logika Fuzzy**

Sebuah fakta dalam kehidupan yaitu bahwa banyak keputusan yang diambil seseorang yang didasarkan pada *fuzzy* (Zadeh, 1996:446). Logika *fuzzy* didasarkan pada bahasa alami, artinya dasar logika *fuzzy* adalah bahasa komunikasi manusia (MathWorks, 1999:5). Bahasa alami adalah bahasa yang digunakan dalam komunikasi sehari-hari. Logika *fuzzy* membangun struktur deskripsi kualitatif penggunaan bahasa sehari-hari tersebut sehingga logika *fuzzy* mudah digunakan.

Logika *fuzzy* adalah suatu cara untuk memetakan suatu ruang *input* ke dalam suatu ruang *output* (MathWorks, 1999:2). Logika *fuzzy* merupakan representasi ketidakpastian, ketidakjelasan, ketidaktepatan yang dikonstruksikan dengan *if-then rules*. Logika *fuzzy* merupakan perkembangan dari logika klasik. Pada logika *fuzzy* nilai kebenaran suatu *item* berada pada interval  $[0,1]$ .

**Contoh 2.3** (MathWorks, 1999:8) Misalkan angka 0 sampai 10 untuk merepresentasikan kualitas pelayanan di suatu restaurant (10 berarti bagus), berapa banyak persenan yang tepat untuk diberikan kepada pelayan?

Masalah ini dapat diselesaikan baik dengan pendekatan *non-fuzzy* maupun pendekatan *fuzzy*. Penyelesaian dengan pendekatan *non-fuzzy* adalah sebagai berikut. Misalkan uang persenan yang diberikan selalu 15% dari total *bill* maka tidak ada perbedaan persenan yang diberikan kepada pelayanan yang bagus maupun tidak. Padahal pada kenyataannya pelanggan akan memberikan persenan

sesuai pelayanan yang diberikan (tidak selalu sama), sehingga perlu ditambahkan aturan lainnya. Adapun penyelesaian dengan pendekatan *fuzzy* bisa bermacam-macam, diantaranya :

1. Jika pelayanan jelek, maka persenan sedikit
2. Jika pelayanan bagus, maka persenan rata-rata
3. Jika pelayanan sangat bagus, maka persenan tinggi

Penyelesaian dengan pendekatan *fuzzy* mendefinisikan aturan untuk sistem logika *fuzzy*, yaitu *if-then rules*. Penyelesaian masalah dengan persamaan sederhana (*linear*) memang membuat masalah lebih sederhana untuk diselesaikan tetapi terkadang tidak sesuai dengan kenyataan.

### **G. Model Fuzzy**

Model *fuzzy* terdiri dari empat komponen yaitu : aturan dasar *fuzzy* (*fuzzy rule base*), fuzzifikasi (*fuzzification*), defuzzifikasi (*defuzzification*), dan inferensi *fuzzy* (*fuzzy inference engine*).

#### 1. Aturan dasar Fuzzy (*Fuzzy Based Rule*)

Aturan *fuzzy* merupakan otak dari model *fuzzy*. Dalam kerangka model *fuzzy*, pengetahuan manusia direpresentasikan dalam bentuk *IF-THEN rules* (Wang, 1997:91). *IF-THEN rules* ditulis sebagai:

$$R^s: IF x_1 \text{ is } A_1^s \circ \dots \circ x_n \text{ is } A_n^s \text{ THEN } z \text{ is } B^s \quad (2.30)$$

dengan,

$s$  = banyak aturan inferensi,

$R^s$  = aturan ke- $s$ ,

$x_i$  = variable *input* ke- $i$ ,  $i = 1, 2, \dots, n$ ,

$A_i^s$  = himpunan *fuzzy input* pada variabel  $x_i$  aturan ke- $s$ ,  $i = 1, 2, \dots, n$ ,

$z$  = nilai *output*,

$B^s$  = himpunan *fuzzy output* pada aturan ke- $s$ ,

◦ = operator *fuzzy* seperti AND atau OR.

## 2. Fuzzifikasi

Fuzzifikasi adalah pemetaan dari himpunan *crisp* (sesuatu yang bernilai pasti) ke himpunan *fuzzy* (sesuatu yang bernilai samar) dengan suatu fungsi keanggotaan (Wang, 1997:105). Proses fuzzifikasi mengubah variabel *non fuzzy* (tegas) menjadi variabel *fuzzy* dengan suatu fungsi keanggotaan.

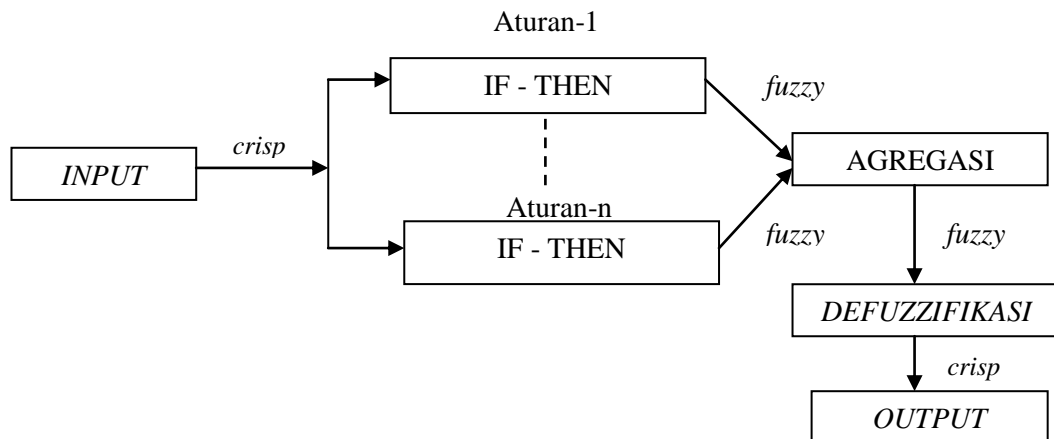
## 3. Defuzzifikasi

Defuzzifikasi menghasilkan *output* tegas untuk sistem logika *fuzzy* dari himpunan *fuzzy* yang merupakan hasil sistem inferensi *fuzzy* (Mendel, 1995:368). Defuzzifikasi merupakan pemetaan *output* yang diperoleh dari himpunan *fuzzy* yang merupakan hasil dari sistem inferensi *fuzzy* ke dalam himpunan *crisp*.

## 4. Inferensi Fuzzy

Inferensi *fuzzy* adalah kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF-THEN, dan logika *fuzzy*. Secara umum, proses inferensi *fuzzy* dapat dilihat pada Gambar 2.10. Pada Gambar 2.10, *input* model *fuzzy* adalah nilai tegas (*crisp*). *Input* ini kemudian dikirim ke basis pengetahuan yang berisi  $n$  aturan *fuzzy* berbentuk IF-THEN. Proses agregasi dilakukan apabila terdapat lebih dari 1 aturan *fuzzy*. Hasil agregasi akan didefuzzifikasi untuk mendapatkan nilai *output* yaitu berupa nilai tegas (*crisp*).





**Gambar 2.10** Proses Inferensi Fuzzy  
(Sri Kusumadewi dan Sri Hartati, 2010:40)

Terdapat 2 metode inferensi *fuzzy*, yaitu metode Tsukamoto dan metode Sugeno (Sri Kusumadewi dan Sri Hartati, 2010:41). Metode inferensi *fuzzy* Tsukamoto didasarkan pada konsep penalaran yang monoton. Setiap aturan pada metode ini, direpresentasikan menggunakan himpunan-himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Metode inferensi *fuzzy* Sugeno atau dikenal Takagi Sugeno Kang (TSK) merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dengan aturan *IF-THEN* dimana *outputnya* tidak berupa himpunan *fuzzy* melainkan berupa konstanta atau persamaan linear. Terdapat 2 model untuk metode inferensi *fuzzy* Sugeno ada 2, yaitu metode Sugeno orde-0 dan metode Sugeno orde-1 (Lin & Lee, 1996: 18-19).

a. Metode Sugeno orde-0

Metode inferensi *fuzzy* Sugeno orde-0 didefinisikan sebagai berikut:

$$IF (x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_n \text{ is } A_n) THEN z = k \quad (2.31)$$

dengan,

$x_i$  = variabel *input* ke- $i$ ,  $i = 1, 2, \dots, n$ ,

$A_i$  = himpunan *fuzzy input* pada variabel  $x_i$ ,  $i = 1, 2, \dots, n$ ,

$z$  = nilai *output*,

$k$  = konstanta (berupa *crisp*),

$\circ$  = operator *fuzzy* seperti AND atau OR.

b. Metode Sugeno orde-1

Metode inferensi *fuzzy* Sugeno orde-1 didefinisikan sebagai berikut:

$$IF (x_1 \text{ is } A_1) \circ \dots \circ (x_n \text{ is } A_n) THEN z = p_1x_1 + \dots + p_nx_n + q \quad (2.32)$$

dengan,

$x_i$  = variabel *input* ke- $i$ ,  $i = 1, 2, \dots, n$ ,

$A_i$  = himpunan *fuzzy input* pada variabel  $x_i$ ,  $i = 1, 2, \dots, n$ ,

$z$  = nilai *output*,

$p_i$  = konstanta *crisp* pada variable  $x_i$ ,

$q$  = konstanta untuk persamaan linier,

$\circ$  = operator *fuzzy* seperti AND atau OR.

Model *fuzzy* memiliki kelebihan yaitu mampu untuk merepresentasikan ketidakpastian dan mudah dalam menginterpretasikan hasil, sedangkan kelemahan model *fuzzy* adalah tidak mampu melakukan generalisasi dan memiliki ketergantungan pada keberadaan ahli untuk menentukan logika aturan inferensi (Vieira, *et al*, 2003:2).

## H. Model *Neuro Fuzzy* (NF)

Model NF merupakan penggabungan antara NN dan logika *fuzzy* ke dalam satu sistem kecerdasan buatan yang terintegrasi. Model NF menggabungkan kelebihan dari model NN dengan kemampuan melakukan pembelajaran dan logika *fuzzy* dengan kemampuan penalarannya. Model NF juga mengurangi kelemahan dari masing-masing model tersebut dengan cara mengadaptasi nilai parameter *fuzzy* dengan fungsi pembelajaran NN. NF dirancang untuk merealisasikan proses penalaran *fuzzy*, dimana bobot-bobot yang terhubung pada NN terhubung dengan parameter-parameter penalaran *fuzzy*. Model NF dalam penelitian ini menggunakan metode inferensi *fuzzy* Sugeno orde satu.

Konsep dasar pemakaian NN sebagai pengendali penalaran *fuzzy* adalah menggunakan NN untuk membangkitkan inferensi *fuzzy* baik pada anteseden (membangkitkan fungsi keanggotaan) maupun pada bagian konsekuen (melakukan inferensi). (Lin & Lee, 1996: 498). NN digunakan untuk menutupi kelemahan inferensi *fuzzy* dalam menentukan fungsi keanggotaan dan pembangkit fungsi pembelajaran pada aturan inferensi. Lin & Lee (1996:507) mendefinisikan aturan yang dipakai dalam model NF dengan metode inferensi *fuzzy* Sugeno sebagai berikut:

$$R^s: \text{IF } x = (x_1, x_2, \dots, x_n) \text{ is } A_s \text{ THEN } z^s = NN_s(x_1, x_2, \dots, x_n) \quad (2.33)$$

dengan,

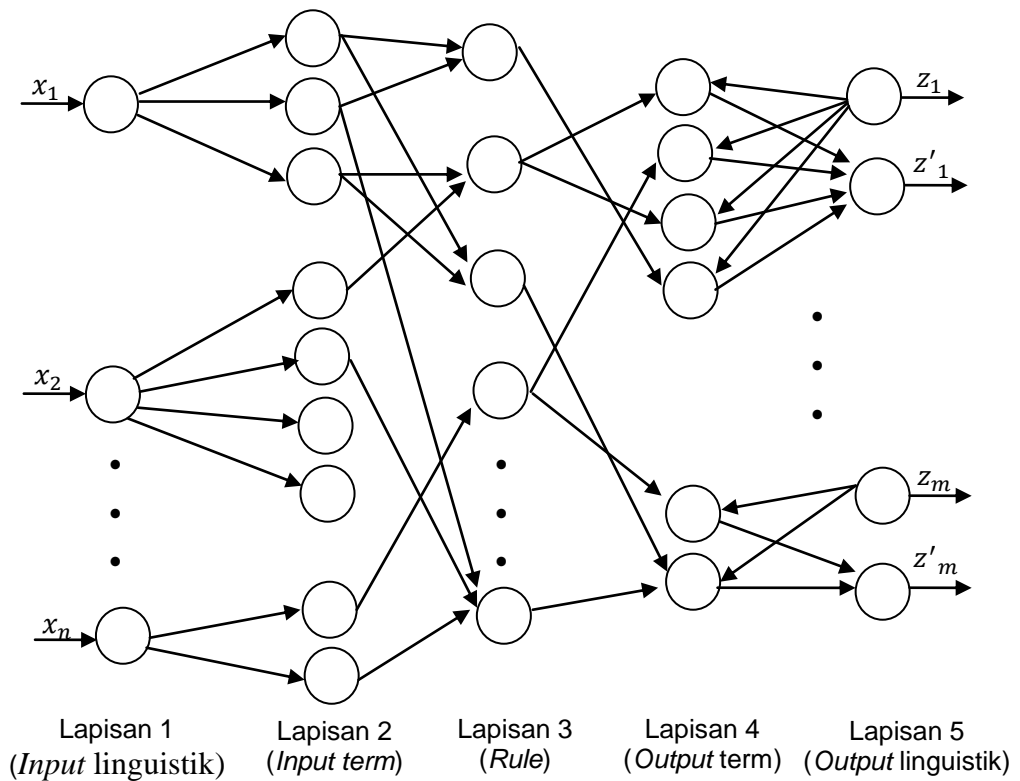
$s$  = banyak aturan inferensi,

$R^s$  = aturan ke- $s$ ,

$x$  = *input*,

- $x_i$  = variabel *input* ke- $i$ ,  $i = 1, 2, \dots, n$ ,  
 $A_s$  = himpunan *fuzzy* pada aturan ke- $s$  bagian anteseden,  
 $z^s$  = nilai *output* aturan ke- $s$ ,  
 $NN_s$  = NN dengan *input*  $(x_1, x_2, \dots, x_n)$  dan *output*  $z^s$ .

Struktur NF secara umum ditunjukkan pada Gambar 2.11. Setiap lapisan pada sistem *neuro fuzzy* dikaitkan dengan langkah-langkah tertentu dalam proses inferensinya. Neuron pada lapisan 1 adalah neuron *input* yang merepresentasikan variable neuron linguistik. Setiap neuron dalam lapisan ini mengirimkan sinyal langsung ke lapisan berikutnya. Lapisan 2 disebut juga sebagai lapisan fuzzifikasi. Neuron pada lapisan ini merupakan representasi dari himpunan *fuzzy* yang digunakan sebagai anteseden dalam aturan *fuzzy*. Neuron pada lapisan fuzzifikasi menerima *input* berupa *crisp* dan menentukan nilai keanggotaan setiap *input*. Lapisan 3 disebut juga sebagai lapisan dengan aturan *fuzzy*. Masing-masing neuron pada lapisan ini merepresentasikan aturan logika *fuzzy*, artinya setiap neuron pada lapisan ini memberentuk aturan dasar *fuzzy*. Neuron pada lapisan 4 adalah himpunan *fuzzy* yang digunakan sebagai konsekuen pada aturan *fuzzy*. Pada lapisan 5 terdapat 2 neuron linguistik untuk masing-masing variabel *output*. Satu neuron untuk mengembalikan ke jaringan dan satu neuron linguistik untuk sinyal keputusan (*output* sebenarnya) yang kemudian dipompa keluar jaringan. (Lin & Lee, 1996: 536-537). Dalam menerapkan aturan *fuzzy* dan melakukan proses fuzzifikasi untuk membangkitkan inferensi *fuzzy* baik pada anteseden (membangkitkan fungsi keanggotaan) maupun pada bagian konsekuen (melakukan inferensi) digunakan NN jaringan *backpropagation*.



**Gambar 2.11** Struktur *Neuro Fuzzy* (Lin & Lee, 1996: 536)

Pada Gambar 2.11, *input* yang diterima lapisan 3 merupakan himpunan *fuzzy* hasil fuzzifikasi (antesenden) dari lapisan 2. *Input* tersebut dimasukkan kedalam neuron-neuron yang bersesuaian (*cluster*). Teknik pengklusteran yang digunakan adalah *Fuzzy C-Means* (FCM), yaitu suatu teknik pengklusteran data berdasarkan derajat keanggotaan data tersebut. Konsep dasar FCM yang pertama adalah menentukan pusat *cluster* yang akan menandai lokasi rata-rata untuk tiap-tiap *cluster*. *Output* dari FCM berupa deretan beberapa pusat *cluster* dan derajat keanggotaan tiap data. Informasi ini dapat digunakan untuk membangun suatu sistem inferensi *fuzzy*.

Algoritma FCM adalah sebagai berikut (Zimmerman, 1991:234):

a. Menentukan:

- 1) Matriks  $X$  dengan ukuran  $n \times m$  dengan  $n$  adalah banyak data yang akan dicluster dan  $m$  banyak variabel. Selanjutnya cluster dinotasikan sebagai  $C$ .
- 2) Banyak cluster yang akan dibentuk. Tidak ada aturan tertentu dalam penentuan banyak cluster, artinya banyak cluster ditentukan secara bebas asalkan cluster yang dibentuk lebih dari 1 ( $C \geq 2$ ).
- 3) Pangkat atau pembobot yang dinotasikan dengan  $w$ . Banyak pembobot harus lebih dari 1 ( $w > 1$ ).
- 4) Maksimum iterasi
- 5) Kriteria penghentian ( $\xi$  atau nilai positif yang sangat kecil)
- 6) Iterasi awal ( $t = 1$  dan  $\Delta = 1$ )

b. Membentuk matriks partisi awal  $U$  sebagai berikut:

$$U = \begin{bmatrix} \mu_{11}(x_1) & \mu_{12}(x_2) & \cdots & \mu_{1n}(x_n) \\ \mu_{21}(x_1) & \mu_{22}(x_2) & \cdots & \mu_{2n}(x_n) \\ \vdots & & & \vdots \\ \mu_{c1}(x_1) & \mu_{c2}(x_2) & \cdots & \mu_{cn}(x_n) \end{bmatrix} \quad (2.34)$$

dengan  $U$  menyatakan matriks partisi awal dan  $\mu_{ik}$  = nilai keanggotaan data ke- $k$  pada cluster ke- $i$  ( $i = 1, 2, \dots, c$  dan  $k = 1, 2, \dots, n$ ).

c. Menghitung pusat cluster  $V$ , untuk setiap cluster:

Setiap cluster memiliki pusat cluster yang berbeda. Matriks pusat cluster adalah sebagai berikut:

$$V = \begin{bmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & & \vdots \\ v_{c1} & \cdots & v_{c2} \end{bmatrix} \quad (2.35)$$

$$v_{ij} = \frac{\sum_{k=1}^n (\mu_{ik})^w x_{kj}}{\sum_{k=1}^n (\mu_{ik})^w} \quad (2.36)$$

dengan,

- $v_{ij}$  = pusat *cluster* variabel ke- $j$  pada *cluster* ke- $i$ ,
- $\mu_{ik}$  = nilai keanggotaan data ke- $k$  pada *cluster* ke- $i$ ,
- $w$  = pangkat atau bobot,
- $x_{kj}$  = data ke- $k$  pada variabel ke- $j$ ,
- $i$  = 1,2, ...,  $c$ ,
- $j$  = 1,2, ...,  $m$ ,
- $k$  = 1,2, ...,  $n$ .

- d. Memperbaiki derajat keanggotaan setiap data pada setiap *cluster* dengan memperbaiki matriks partisi.

$$\mu_{ik} = \left[ \sum_{j=1}^c \left( \frac{d_{ik}}{d_{jk}} \right)^{2/(w-1)} \right]^{-1} \quad (2.37)$$

dengan,

$$d_{ik} = d(x_k - v_i) = \left[ \sum_{j=1}^m (x_{kj} - v_{ij}^*) \right]^{1/2} \quad (2.38)$$

dimana,

- $\mu_{ik}$  = nilai keanggotaan data ke- $k$  pada *cluster* ke- $i$ ,
- $d_{ik}$  = jarak antara pusat *cluster* ke- $i$  dengan data ke- $k$ ,
- $d_{jk}$  = jarak antara pusat *cluster* variabel ke- $j$  dengan data ke- $k$ ,
- $x_{kj}$  = data ke- $k$  pada variabel ke- $j$ ,
- $v_{ij}$  = pusat *cluster* variabel ke- $j$  pada *cluster* ke- $i$ ,
- $i$  = 1,2, ...,  $c$ ,

$$j = 1, 2, \dots, m,$$

$$k = 1, 2, \dots, n.$$

e. Menentukan kriteria berhenti, yaitu perubahan matriks partisi pada iterasi sekarang dengan iterasi sebelumnya.

$$\Delta = \|U^t - U^{t-1}\| \quad (2.391)$$

Apabila  $\Delta \leq \xi$ , maka iterasi dihentikan. Apabila  $\Delta > \xi$ , maka iterasi dinaikan ( $t = t + 1$ ) dan kembali ke langkah-3. Pencarian nilai  $\Delta$  dapat dilakukan dengan mengambil elemen terbesar dari nilai mutlak selisih antara  $\mu_{ik}(t)$  dengan  $\mu_{ik}(t-1)$ .

### I. Mean Square Error (MSE)

MSE adalah rata-rata jumlah kuadrat selisih target dan hasil *output* jaringan (rata-rata *error* klasifikasi). Semakin kecil nilai MSE dari suatu model yang digunakan untuk klasifikasi maka semakin baik klasifikasi yang dihasilkan. MSE dirumuskan sebagai berikut (Hanke & Wichern, 2005:79):

$$MSE = \frac{1}{n} \sum_{t=1}^n (g_t - z_t)^2 \quad (2.40)$$

dengan,

$g_t$  = target ke-  $t$ ,

$z_t$  = *output* jaringan ke-  $t$ ,

$t = 1, 2, \dots, n$ ,

$n$  = banyak data pengamatan.

### J. Sensitivitas, Spesifisitas, dan Akurasi

Keputusan praktisi medis mengenai tindakan medis yang harus dilakukan bergantung pada hasil uji diagnosa, yang mana ketelitian diagnosa adalah hal yang



sangat penting dalam dunia medis. Tingkat ketelitian diagnosa dapat diukur. Sensitivitas, spesifisitas, dan akurasi adalah statistik yang umum digunakan untuk mendeskripsikan hasil uji diagnosa (Zhu, *et al*, 2010:2). Ketiganya digunakan untuk mengetahui seberapa bagus dan terpercaya hasil uji yang telah dilakukan. Kemungkinan yang dapat terjadi dalam hasil uji diagnosa ditunjukkan dalam Tabel 2.1 berikut.

**Tabel 2.1** Hasil Uji Diagnosa Disajikan dalam Tabel 2x2 (Osborn, 2006:71)

<i>Test/Measure</i>	<i>True Situation / Event</i>		<i>Total</i>
	<i>Performance Indicator Present</i>	<i>Performance Indicator Absent</i>	
<i>Positive</i>	<i>True positive (a)</i>	<i>False positive (b)</i>	<i>a+b</i>
<i>Negative</i>	<i>False negative (c)</i>	<i>True negative (d)</i>	<i>c+d</i>
<i>Total</i>	<i>a+c</i>	<i>b+d</i>	<i>a+b+c+d</i>

Berdasarkan Tabel 2.1, terdapat 4 kemungkinan hasil uji diagnosa yaitu :

1. *true positive (a)* = pasien memang memiliki penyakit dan hasil uji tepat,
2. *false positive (b)* = hasil uji menyatakan ada penyakit tetapi tidak benar karena pasien tidak memiliki penyakit,
3. *false negative (c)* = pasien memang memiliki penyakit tetapi hasil uji tidak menyatakan begitu,
4. *true negative (d)* = pasien tidak memiliki penyakit dan hasil ujinya tepat.

### **1. Sensitivitas**

Sensitivitas adalah proporsi dari *true positive* teridentifikasi secara tepat dalam uji diagnosa (Zhu, *et al*, 2010:2). Spitalnic (2004:1) menyatakan bahwa sensitivitas adalah peluang hasil uji positif diberikan kepada pasien dengan kondisi memang berpenyakit. Contohnya, jika sensitivitas = 95%, artinya ketika dilakukan uji diagnosa pada pasien yang memiliki penyakit maka pasien tersebut

berpeluang 95% dinyatakan positif (berpenyakit). Rumus sensitivitas adalah sebagai berikut (Zhu, *et al*, 2010:2):

$$\text{Sensitivitas} = \frac{a}{a+c} \quad (2.41)$$

## 2. Spesifisitas

Zhu, *et al* (2010:2) menyatakan bahwa spesifisitas adalah proporsi dari *true negative* teridentifikasi secara tepat dalam uji diagnosa. Spesifisitas adalah peluang hasil uji negatif diberikan kepada pasien dengan kondisi memang tidak berpenyakit (Spitalnic, 2004:1). Contohnya, jika spesifitas = 95%, artinya ketika dilakukan uji diagnosa pada pasien yang tidak berpenyakit maka pasien berpeulang 95% dinyatakan negatif (tidak berpenyakit). Rumus spesifisitas adalah sebagai berikut (Zhu, *et al*, 2010:2):

$$\text{Spesifisitas} = \frac{d}{b+d} \quad (2.42)$$

## 3. Akurasi

Uji dinyatakan akurat jika merefleksikan nilai kebenaran atau mendeteksi adanya substansi yang diukur (Lord, 2008:11). Nilai akurasi merepresentasikan proporsri dari hasil *true positive* dan *true negative* dalam populasi terpilih dan dirumuskan sebagai berikut (Zhu, *et al*, 2010:2):

$$\text{Akurasi} = \frac{a+d}{a+b+c+d} \quad (2.43)$$

Contohnya, jika akurasi = 95% maka berarti hasil diagnosa akurat 95% baik untuk pasien yang dinyatakan positif maupun yang dinyatakan negatif.

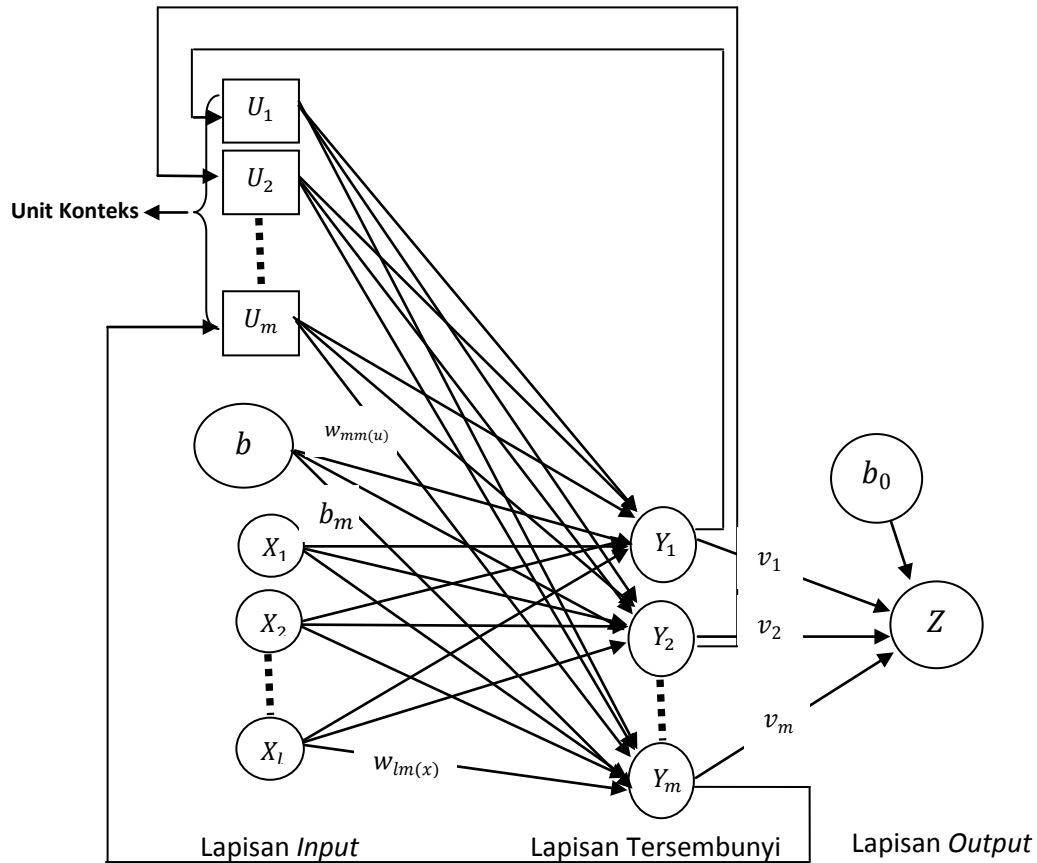
### BAB III PEMBAHASAN

#### **A. Prosedur Pemodelan *Recurrent Neural Network* (RNN) untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru**

RNN adalah jaringan yang mengakomodasikan *output* jaringan menjadi *input* pada jaringan tersebut untuk selanjutnya menghasilkan *output* jaringan berikutnya. Terdapat 2 jaringan RNN yaitu jaringan Hopfield dan jaringan Elman. Jaringan Hopfield adalah jaringan umpan balik satu lapis (*single layer feedback*) dengan bobot simetris yang dikenalkan oleh John Hopfield pada 1982 (Lin & Lee, 1996:6). Jaringan ini memiliki koneksi umpan balik (*feedback link*) pada seluruh neuron yang terbangun. Artinya, pada jaringan ini seluruh fungsi unit adalah sebagai *input* dan *output* sekaligus (Fausett, 1994:12). Jaringan Elman merupakan jaringan sederhana dari RNN karena pada jaringan ini koneksi umpan balik (*feedback link*) hanya terdapat pada lapisan tersembunyi ke lapisan *input*. Adapun jaringan RNN yang digunakan dalam tugas akhir ini adalah jaringan Elman. Arsitektur jaringan RNN Elman dapat dilihat pada Gambar 3.1.

Pada Gambar 3.1,  $X_1, X_2, \dots, X_l$  adalah neuron pada lapisan *input*,  $Y_1, Y_2, \dots, Y_m$  adalah neuron pada lapisan tersembunyi, dan  $Z$  adalah neuron pada lapisan *output*.  $U_1, U_2, \dots, U_m$  adalah neuron tambahan yang merupakan umpan balik dari lapisan tersembunyi.  $U_1$  merupakan umpan balik dari  $Y_1$ ,  $U_2$  merupakan umpan balik dari  $Y_2$ , dan seterusnya. Bobot dari sinyal *input* ke lapisan tersembunyi disimbolkan dengan  $w_{jk(x)}$ , bobot dari neuron tambahan ke lapisan tersembunyi

disimbolkan dengan  $w_{kk(u)}$  dan bobot dari lapisan tersembunyi ke lapisan *output* di simbolkan dengan  $v_k$  ( $j=1,2,\dots, l$  dan  $k=1,2,\dots, m$ ). Bobot bias pada neuron ke- $k$  pada lapisan tersembunyi disimbolkan dengan  $b_k$  dan bobot bias pada neuron lapisan *output* disimbolkan dengan  $b_0$ .



**Gambar 3.1** Jaringan Elman

Model RNN pada jaringan Elman secara matematis dapat dirumuskan sebagai

berikut:

$$z = \sum_{k=1}^m v_k \frac{1 - \exp\left(-\left(\sum_{i=1}^l x_i w_{ik(x)} + \sum_{k=1}^m u_k w_{kk(u)} + b_k\right)\right)}{1 + \exp\left(-\left(\sum_{i=1}^l x_i w_{ik(x)} + \sum_{k=1}^m u_k w_{kk(u)} + b_k\right)\right)} + b_0 + \varepsilon \quad (3.1)$$

dengan,

$z$  = nilai *output*,

$v_k$  = bobot dari neuron ke- $k$  pada lapisan tersembunyi yang menuju lapisan *output* dengan  $k = 1, 2, \dots, m$ ,

$x_i$  = variabel *input* dengan  $i = 1, 2, \dots, l$ ,

$w_{ik(x)}$  = bobot dari lapisan *input* ke- $i$  menuju neuron ke- $k$  pada lapisan tersembunyi,

$u_k$  = variabel *input* yang berada pada neuron tambahan (unit konteks) dengan  $k = 1, 2, \dots, m$ ,

$w_{kk(u)}$  = bobot dari lapisan *input* (neuron tambahan) ke- $k$  menuju neuron ke- $k$  pada lapisan tersembunyi,

$b_k$  = bobot bias pada neuron ke- $k$  pada lapisan tersembunyi dengan  $k = 1, 2, \dots, m$ ,

$b_0$  = bobot bias pada neuron lapisan *output*,

$\varepsilon$  = *error*.

### 1. Algoritma *Backpropagation* pada RNN Jaringan Elman

Algoritma pembelajaran yang digunakan dalam tugas akhir ini adalah algoritma pembelajaran *backpropagation* yang termasuk algoritma terawasi. Algoritma ini dirancang untuk operasi pada *multilayer neural network*. Pembelajaran *backpropagation* terdiri dari 3 fase yaitu perambatan maju (*feedforward*), perambatan mundur (*backward*) dan perubahan bobot (Fausett, 1994: 294-295). Fase pertama adalah fase perambatan maju, pola masukan dihitung maju dari lapisan *input* sampai lapisan *output* menggunakan suatu fungsi aktivasi. Fase selanjutnya adalah fase perambatan mundur, *error* yang merupakan selisih antara *output* jaringan dan target yang diinginkan terjadi di perambatan mundur. Fase perambatan mundur dimulai dari

sinyal yang berhubungan langsung dengan neuron pada lapisan *output* yang kembali menuju neuron pada lapisan *input*. Fase terakhir yaitu perubahan bobot yang dimaksudkan untuk menurunkan *error* yang terjadi dengan cara memodifikasi bobot. Karena jaringan yang digunakan dalam tugas akhir ini adalah RNN jaringan Elman maka fase-fase disesuaikan dengan RNN jaringan Elman yaitu dengan adanya umpan balik dari *output* lapisan tersembunyi ke lapisan *input*.

Algoritma *backpropagation* pada RNN jaringan Elman adalah sebagai berikut:

1. Menetapkan parameter pembelajaran, yaitu:

a. Maksimum epoch

Maksimum epoch adalah jumlah epoch maksimum yang boleh dilakukan selama proses pembelajaran. Iterasi akan dihentikan apabila nilai epoch melebihi maksimum epoch. Nilai default untuk maksimum epoch adalah 10. Sintaks pada MatlabR2010a adalah sebagai berikut:

```
net.trainParam.epochs = MaxEpoch.
```

b. Target *Error*

Target *error* adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dan atau sama dengan target *error*. Nilai default untuk kinerja tujuan adalah 0. Sintaks pada MatlabR2010a:

```
net.train.Param.goal = TargetError.
```

c. *Learning rate*

*Learning rate* adalah laju pembelajaran. Semakin besar nilai *learning rate*, semakin besar pula langkah pembelajaran. Semakin kecil *learning rate*, maka proses pembelajaran akan sangat lama. Sehingga perlu pemilihan nilai yang tepat untuk *learning rate*. Nilai default untuk *learning rate* adalah 0.01.

Sintaks pada MatlabR2010a: `net.trainParam.lr = LearningRate.`

d. Rasio untuk menaikkan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menaikkan *learning rate* apabila *learning rate* yang ada terlalu rendah untuk mencapai kekonvergenan. Nilai default untuk rasio kenaikan *learning rate* adalah 1.05.

Sintaks pada MatlabR2010a adalah sebagai berikut:

`net.trainParam.lr.inc=IncLearningRate.`

e. Rasio untuk menurunkan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menurunkan *learning rate* apabila *learning rate* yang ada terlalu tinggi untuk menuju ketidakstabilan. Nilai default untuk rasio penurunan *learning rate* adalah 0.7.

Sintaks pada MatlabR2010a: `net.trainParam.lr.dec=DecLearningRate.`

f. Jumlah epoch yang akan ditunjukkan kemajuannya

Menunjukkan jumlah epoch yang akan ditampilkan. Nilai default untuk jumlah epoch yang akan ditunjukkan adalah 25. Sintaks pada MatlabR2010a:

`net.trainParam.show = EpochShow.`

g. Maksimum kenaikan kinerja

Maksimum kenaikan kinerja adalah nilai maksimum kenaikan *error* yang diijinkan, antara *error* saat ini dan *error* sebelumnya. Nilai default untuk maksimum kenaikan kinerja adalah 1,04. Sintaks pada MatlabR2010a:  
`net.trainParam.max_perf_inc=MaxPerfInc.`

h. Momentum

Momentum adalah perubahan bobot yang didasarkan atas arah gradient pola terakhir dan pola sebelumnya. Besarnya momentum antara 0 sampai 1. Apabila nilai momentum = 0, maka perubahan bobot hanya akan dipengaruhi oleh gradiennya. Tetapi, apabila nilai momentum = 1, maka perubahan bobot akan sama dengan perubahan bobot sebelumnya. Sintaks pada MatlabR2010a:  
`net.trainParam.mc = Momentum.`

2. Kerjakan langkah berikut selama Epoch<Maksimum Epoch dan MSE>Target *Error* dengan target *error* 0,01.

a. Fase 1 (Perambatan Maju)

1) Tiap neuron *input* ( $x_i, i=1,2,3...l$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua neuron pada lapisan setelahnya, yaitu lapisan tersembunyi. Sinyal yang diterima lapisan tersembunyi dari lapisan *input* adalah sebagai berikut:

$$y_{in(1)_k} = b_k + \sum_{i=1}^l x_i w_{ik(x)} \quad (3.2)$$



dengan,

$b_k$  = bobot bias pada neuron ke- $k$  pada lapisan tersembunyi  
dengan  $k=1,2,\dots$ ,

$x_i$  = variabel *input*,  $i = 1,2,\dots,l$ ,

$w_{ik(x)}$  = bobot dari lapisan *input* ke- $i$  menuju neuron ke- $k$  pada  
lapisan tersembunyi.

Sinyal *output*  $y_{in(1)_k}$  dihitung menggunakan fungsi aktivasi sebagai  
berikut:

$$y(1)_k = f(y_{in(1)_k}) \quad (3.3)$$

2) Tiap neuron tambahan ( $u_k, k=1,2,3,\dots,m$ ) menerima sinyal dan meneruskan  
sinyal ke lapisan tersembunyi. Sinyal yang diterima neuron tambahan  
adalah hasil dari aktivasi sinyal *output* lapisan tersembunyi, yaitu sebagai  
berikut:

$$u_k = y(1)_k \quad (3.4)$$

Neuron tambahan kemudian mengirimkan kembali sinyal ke neuron lapisan  
tersembunyi sebagai berikut:

$$y_{in(2)_k} = \sum_{k=1}^m u_k w_{kk(u)} \quad (3.5)$$

dengan,

$u_k$  = sinyal *output* (telah diaktivasi) yang diterima lapisan tersembunyi dari neuron *input*  $k=1,2,\dots, m$ ,

$w_{kk(u)}$  = bobot dari lapisan *input* (neuron tambahan) ke- $k$  menuju neuron ke- $k$  pada lapisan tersembunyi.

- 3) Tiap neuron pada lapisan tersembunyi ( $y_k, k=1,2,\dots,m$ ) menjumlahkan sinyal-sinyal *input* terbobot yang diperoleh dari lapisan *input* dan neuron tambahan sebagai berikut:

$$y_{-in_k} = y_{in(1)_k} + y_{in(2)_k} = b_k + \sum_{i=1}^l \sum_{k=1}^m x_i w_{ik(x)} + \sum_{k=1}^m u_k w_{kk(u)} \quad (3.6)$$

dengan,

$y_{in(1)_k}$  = sinyal yang diterima lapisan tersembunyi dari neuron lapisan *input*,

$y_{in(2)_k}$  = sinyal yang diterima lapisan tersembunyi dari neuron tambahan,

$b_k$  = bobot bias pada neuron ke- $k$  pada lapisan tersembunyi dengan  $k=1,2,\dots, m$ ,

$x_i$  = variabel *input*,  $i = 1,2,\dots,l$ ,

$u_k$  = sinyal *output* (telah diaktivasi) yang diterima lapisan tersembunyi dari neuron *input*  $k=1,2,\dots, m$ ,

$w_{ik(x)}$  = bobot dari lapisan *input* ke- $i$  menuju neuron ke- $k$  pada lapisan tersembunyi,

$w_{kk(u)}$  = bobot dari lapisan *input* (neuron tambahan) ke- $k$  menuju neuron ke- $k$  pada lapisan tersembunyi.

Sinyal *output*  $y_{in_k}$  dihitung menggunakan fungsi aktivasi sebagai berikut:

$$y_k = f(y_{in_k}) \quad (3.7)$$

kemudian sinyal tersebut dikirim ke neuron pada lapisan *output*. Fungsi aktivasi yang digunakan dari lapisan *input* ke lapisan tersembunyi dalam tugas akhir ini adalah *logsig*, sehingga Persamaan (3.3) dan Persamaan (3.7) menjadi

$$y(1)_k = \frac{1}{1+e^{-(y_{in(1)})_k}} \quad (3.8)$$

$$\text{dan } y_k = \frac{1}{1+e^{-(y_{in_k})}} \quad (3.9)$$

4) Tiap neuron *output* ( $Z$ ) menjumlahkan sinyal-sinyal *input* terbobot dari lapisan tersembunyi menggunakan rumus berikut:

$$z_{in} = b_0 + \sum_{i=1}^l v_k y_k \quad (3.10)$$

dengan,

$b_0$  = bias pada neuron *output*,

$v_k$  = bobot dari neuron ke- $k$  pada lapisan tersembunyi yang menuju lapisan *output* dengan  $k = 1, 2, \dots, m$ ,

$y_k$  = sinyal *output* yang telah diaktivasi dari lapisan tersembunyi dengan  $k = 1, 2, \dots, m$ .

Sinyal *output*  $z_{in}$  dihitung menggunakan fungsi aktivasi sebagai berikut:

$$z = f(z_{in}) \quad (3.11)$$

Fungsi aktivasi yang digunakan dari lapisan *input* ke lapisan *output* dalam tugas akhir ini adalah *tansig*, sehingga Persamaan (3.11) menjadi

$$z = \frac{1 - e^{-(z\_in)}}{1 + e^{-(z\_in)}} \quad (3.12)$$

b. Fase 2 (Perambatan Mundur)

- 1) Neuron *output* ( $z$ ) menerima target pola yang berhubungan dengan pola *input* pembelajaran, kemudian dihitung informasi *error*nya:

$$\delta_0 = (t_k - z_k) f'(z\_in) \quad (3.13)$$

Kemudian dihitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai  $v_k$  dan koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai  $b_0$ , rumus-rumusnya adalah sebagai berikut:

$$\Delta v_k = \alpha \delta_0 z_k \quad (3.14)$$

$$\Delta b_0 = \alpha \delta_0 \quad (3.15)$$

dimana  $\alpha$  = laju percepatan.

Kemudian  $\delta_0$  dikirimkan ke neuron-neuron pada lapisan sebelumnya (lapisan tersembunyi). Langkah ini dilakukan sebanyak jumlah neuron pada lapisan tersembunyi.

- 2) Tiap-tiap neuron tersembunyi ( $y_k, k = 1, 2, \dots, m$ ) menjumlahkan delta *input*nya dari neuron-neuron yang berada pada lapisan di atasnya sebagai berikut:

$$\delta\_in_k = \sum_{k=1}^m \delta_0 v_k \quad (3.16)$$

Kalikan nilai  $\delta\_in_k$  dengan turunan dari fungsi aktivasinya:

$$\delta_k = \delta_{in_k} f' (y_{in_k}) \quad (3.17)$$

Kemudian hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai  $w_{ik(x)}$  dan  $w_{kk(u)}$  dengan rumus sebagai berikut:

$$\Delta w_{ik(x)} = \alpha \delta_k x_i \quad (3.18)$$

$$\Delta w_{kk(u)} = \alpha \delta_k u_k \quad (3.19)$$

selain itu, hitung koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai  $b_k$ :

$$\Delta b_k = \alpha \delta_k \quad (3.20)$$

### c. Fase Perubahan Bobot

1) Neuron *output*( $z$ ) memperbaiki bias dan bobotnya sebagai berikut:

$$v_k(\text{baru}) = v_k(\text{lama}) + \Delta v_k \quad (3.21)$$

$$b_0(\text{baru}) = b_0(\text{lama}) + \Delta b_0 \quad (3.22)$$

2) Tiap-tiap neuron lapisan tersembunyi ( $y_k$ ,  $k = 1, 2, \dots, m$ ) memperbaiki bias dan bobotnya sebagai berikut:

$$w_{ik(x)}(\text{baru}) = w_{ik(x)}(\text{lama}) + \Delta w_{ik(x)} \quad (3.23)$$

$$w_{kk(u)}(\text{baru}) = w_{kk(x)}(\text{lama}) + \Delta w_{kk(x)} \quad (3.24)$$

$$b_k(\text{baru}) = b_k(\text{lama}) + \Delta b_k \quad (3.25)$$

## 2. Membangun Jaringan Elman pada MatlabR2010a

Langkah pertama dalam membangun sebuah jaringan adalah menentukan *input* jaringan untuk selanjutnya diestimasi banyak neuron yang terletak pada lapisan tersembunyi. Pada Matlab R2010a, untuk membangun jaringan Elman digunakan

perintah *newelm*. Secara lebih lengkap, perintah pada toolbox Matlab R2010a adalah sebagai berikut:

$$\text{net} = \text{newelm}(\text{RP}, [S_1, S_2, \dots, S_q], \{\text{TF}_1, \text{TF}_2, \dots, \text{TF}_q\}, \text{BTF})$$

dengan,

PR : matriks berukuran  $R \times 2$  yang berisi nilai minimum dan maksimum, dengan  $R$  adalah banyak variabel *input*;

Si : banyak neuron pada lapisan ke- $p$ , dengan  $p = 1, 2, \dots, q$ ,

TF $_i$  : fungsi aktivasi pada lapisan ke- $p$ , dengan  $p = 1, 2, \dots, q$  (default: *tansig*),

BTF : fungsi pembelajaran untuk jaringan *backpropagation* (default: *traingdx*).

### 3. Prosedur Pemodelan RNN

Pemodelan RNN dilakukan dengan jaringan Elman algoritma *backpropagation*. Langkah-langkah pemodelan RNN untuk klasifikasi nodul kanker paru dari citra foto paru adalah sebagai berikut.

#### a. Mendefinisikan variabel *input* dan target

Pemodelan RNN diawali dengan mendefinisikan variabel *input* dan target yang akan digunakan sebagai *input* dan target jaringan. Variabel *input* yang digunakan adalah hasil ekstraksi citra foto paru. Proses ekstraksi citra dilakukan dengan metransformasi citra foto paru asli (berupa citra abu-abu) menjadi citra biner terlebih dahulu. Setelah citra ditransformasi, proses dilanjutkan dengan ekstraksi citra dengan metode *Gray Level Co-Occurrence Matrix* (GLCM). Proses ekstraksi citra dan hasil ekstraksi telah dibahas pada

Bab II. Hasil ekstraksi citra inilah yang digunakan sebagai variabel *input* RNN. Target RNN adalah keterangan kondisi dari masing-masing citra foto paru yaitu paru normal, tumor paru jinak, atau tumor paru ganas.

b. Membagi data

Data dibagi menjadi 2 yaitu data *training* dan data *testing*. Data *training* digunakan untuk pelatihan yang menghasilkan bobot-bobot jaringan sedangkan data *testing* digunakan untuk pengujian atau validasi data. Pilihan ukuran pembagian data *training* dan data *testing* yang dapat digunakan adalah 60%-40%, 75%-25%, dan 80%-20% (Hota, Shrivastava & Singhvi, 2013:165). Adapun ukuran pembagian data yang digunakan dalam tugas akhir ini adalah 75% untuk data *training* dan 25% untuk data *testing*.

c. Normalisasi data

Sebelum dilakukan pembelajaran, data perlu dinormalisasi terlebih dahulu. Rumus normalisasi data adalah sebagai berikut (Jayalakhsmi, 2011:91):

$$x_t^* = \frac{x_t - \bar{x}}{s} \quad (3.26)$$

dengan,

$x_t$  = nilai data ke-  $t$ ;  $t = 1, 2, \dots, n$ ,

$\bar{x}$  = rata-rata nilai data;  $\bar{x} = \frac{1}{n} \sum_{t=1}^n x_t$ ,

$s$  = standar deviasi data;  $s = \sqrt{\frac{1}{n-1} \sum_{t=1}^n (x_t - \bar{x})^2}$ ,

$n$  = banyak data.

Normalisasi dapat juga dilakukan dengan bantuan MatlabR2010a dengan perintah `prestd`. Sintaks lengkapnya adalah sebagai berikut:

$$[P_n, \text{mean}_p, \text{std}_p, T_n, \text{mean}_t, \text{std}_t] = \text{prestd}(P', T')$$

dengan,

$P'$  : transpose dari matriks *input*

$T'$  : transpose dari matriks target

Hasil yang diperoleh dari perintah ini adalah sebagai berikut:

$P_n$  : matriks *input* ternormalisasi

$\text{mean}_p$  : mean dari matriks *input* (P)

$\text{std}_p$  : standar deviasi pada matriks *input* (P)

$T_n$  : matriks target yang ternormalisasi

$\text{mean}_t$  : mean dari matriks target (T)

$\text{std}_t$  : standar deviasi pada matriks target (T)

#### d. Perancangan model terbaik

Perancangan model terbaik dimulai dengan menentukan banyak neuron pada lapisan tersembunyi dengan cara *trial and error*. Setelah banyak neuron pada lapisan tersembunyi ditentukan, langkah selanjutnya adalah melakukan eliminasi variable *input* sehingga diperoleh *input* yang optimal. Pembelajaran RNN baik untuk menentukan banyak neuron pada lapisan tersembunyi maupun menentukan *input* optimal dilakukan menggunakan algoritma *backpropagation*. Algoritma ini dilakukan setelah seluruh data ternormalisasi. Pemilihan model terbaik dengan melihat nilai *Mean Square*



*Error* (MSE) hasil pembelajaran. Prinsip parsimoni digunakan untuk membandingkan nilai MSE yang diperoleh dari hasil pembelajaran. Berdasarkan prinsip parsimoni, model yang terbaik adalah model yang mempunyai nilai MSE kecil dengan jaringan yang sederhana. Artinya, jika terdapat 2 atau lebih model yang menghasilkan nilai MSE yang hampir sama, model yang dipilih adalah model dengan arsitektur yang paling sederhana.

e. Denormalisasi data

Denormalisasi dimaksudkan untuk mengembalikan data yang dinormalisasikan selama proses pembelajaran. Denormalisasi dilakukan dengan perintah `posttd`. Sintaks pada MatlabR2010a untuk proses denormalisasi adalah sebagai berikut:

```
[P, T]=posttd(Pn, meanp, stdp, Tn, meant, stdt)
```

## **B. Prosedur Pemodelan *Recurrent Neuro Fuzzy* (RNF) untuk Klasifikasi Nodul**

### **Kanker Paru dari Citra Foto Paru**

Model RNF adalah salah satu bentuk pengembangan dari model *Neuro Fuzzy* (NF). Model NF dirancang untuk merealisasikan logika *fuzzy* menggunakan suatu algoritma *Neural Network* (NN). Pada tugas akhir ini akan dibahas mengenai RNF yang merupakan realisasi logika *fuzzy* menggunakan RNN jaringan Elman dengan algoritma *backpropagation*. Pembelajaran RNN digunakan untuk membangkitkan fungsi keanggotaan pada bagian antesenden (IF) dan sebagai fungsi pembelajaran pada inferensi *fuzzy*. Metode inferensi *fuzzy* yang digunakan adalah metode inferensi

*fuzzy* Sugeno orde satu. Dasar aturan *fuzzy* dalam model RNF adalah pengembangan dari aturan yang dipakai dalam model NF (Persamaan 2.33) yaitu sebagai berikut:

$$R^s: IF \left( (x_{t1}^s, x_{2t}^s \dots x_{lt}^s), g_t^s \right) \text{ is } A_s \text{ THEN } z^s = RNN_s \left( (x_{t1}^s, x_{2t}^s \dots x_{lt}^s), g_t^s \right) \quad (3.27)$$

dengan

$R^s$  = aturan inferensi ke- $s$  ( $s$  adalah banyak aturan inferensi,  $s = 1, 2, \dots, r$ ),

$x_{it}^s$  = variabel *input* ke- $i$  data ke- $t$  pada aturan inferensi ke- $s$ ,

$g_t^s$  = target jaringan data ke- $t$  pada aturan inferensi ke- $s$ ,

$A_s$  = himpunan *fuzzy input* bagian anteseden (IF) pada aturan inferensi ke- $s$ ,

$z^s$  = nilai *output* pada aturan inferensi ke- $s$ ,

$RNN_s$  = RNN pada aturan inferensi ke- $s$ ,

$i$  = 1, 2, ...  $l$  ( $l$  adalah banyak variabel *input*).

Himpunan *fuzzy* bagian anteseden (bagian IF) menggunakan fungsi keanggotaan sebagai berikut:

$$\mu_{A_s}(x_{t1}^s, x_{2t}^s, \dots, x_{lt}^s) = \frac{1}{1 + e^{-(b_0 + \sum_{i=1}^l v_k y_k)}} \quad (3.28)$$

dengan,

$b_0$  = bias pada neuron *output*,

$v_k$  = bobot dari neuron ke- $k$  pada lapisan tersembunyi yang menuju lapisan *output*,

$y_k$  = sinyal *output*  $y_{in_k}$ ;  $y_k = \frac{1}{1 + e^{-(y_{in_k})}}$ ,

$y_{in_k} = y_{in(1)_k} + y_{in(2)_k} = b_k + \sum_{i=1}^l \sum_{k=1}^m x_i w_{ik}(x) + \sum_{k=1}^m u_k w_{kk}(u)$ ,

$y_{in(1)_k}$  = sinyal yang diterima lapisan tersembunyi dari neuron lapisan

$$input; y_{in(1)_k} = b_k + \sum_{i=1}^l x_i w_{ik(x)},$$

$y_{in(2)_k}$  = sinyal yang diterima lapisan tersembunyi dari neuron tambahan;  $y_{in(2)_k} =$

$$\sum_{k=1}^m u_k w_{kk(u)},$$

$b_k$  = bobot bias pada neuron ke- $k$  pada lapisan tersembunyi,

$x_i$  = variabel *input* ke- $i$ ,

$u_k$  = sinyal *output* ke- $k$  (telah diaktivasi) yang diterima lapisan tersembunyi dari neuron *input*,

$w_{ik(x)}$  = bobot dari lapisan *input* ke- $i$  menuju neuron ke- $k$  pada lapisan tersembunyi,

$w_{kk(u)}$  = bobot dari lapisan *input* (neuron tambahan) ke- $k$  menuju neuron ke- $k$  pada lapisan tersembunyi,

$k$  = 1, 2, ...,  $m$ ,

$i$  = 1, 2, ...,  $l$ .

Langkah pertama pemodelan RNF dengan metode inferensi *fuzzy* Sugeno orde satu dan menggunakan RNN jaringan Elman dengan algoritma *backpropagation* adalah penentuan *input* dan target jaringan RNF. Langkah penentuan *input* dan target jaringan RNF adalah sama dengan langkah pemodelan RNN, sehingga *input* dan target jaringan RNF adalah hasil dari pembelajaran RNN. Langkah pemodelan RNF selanjutnya adalah sebagai berikut (Lin & Lee, 1996:509-510):

### 1. *Clustering* data

Pada langkah ini, hasil pembelajaran RNN akan dibagi menjadi  $r$  *cluster* dengan menggunakan metode *Fuzzy C-Means* (FCM) sehingga akan diperoleh  $r$  buah aturan  $R^s, s = 1, 2, \dots, r$ . Tidak ada aturan tertentu dalam penentuan banyak *cluster*, artinya banyak *cluster* ditentukan secara bebas asalkan *cluster* yang dibentuk lebih dari 1. Prosedur *clustering* menggunakan FCM telah dibahas pada Bab II. Pasangan *input* dan target pada aturan ke- $s$  akan direpresentasikan sebagai  $((x_{1t}^s, x_{2t}^s, \dots, x_{lt}^s), g_t^s)$ .

### 2. Pembelajaran RNN yang berhubungan dengan bagian anteseden (bagian IF) pada aturan-aturan inferensi *fuzzy*

Dilakukan pembelajaran RNN yang berhubungan dengan bagian anteseden (bagian IF) pada aturan-aturan inferensi *fuzzy*. Pembelajaran dilakukan menggunakan RNN jaringan Elman dengan algoritma *backpropagation*. Untuk masing-masing *input* data *training*,  $(x_{1t}, x_{2t}, \dots, x_{lt}) \in R^s$ , didefinisikan matriks partisi  $m_{it} = (m_{1t}^1, m_{2t}^2, \dots, m_{it}^r)$ , dengan ketentuan

$$m_{it}^s = \begin{cases} 1; & k=s \\ 0; & k \neq s \end{cases} \quad (3.29)$$

Selanjutnya, pasangan  $((x_{1t}, x_{2t}, \dots, x_{lt}), m_{it}^s)$  dijadikan *input* dan target pembelajaran RNN. Hasil pembelajaran adalah nilai keanggotaan bagian anteseden yang didefinisikan sebagai nilai  $m_{it}^s$  terinferensi ( $\hat{m}_{it}^s$ ) yang merupakan *output* dari pembelajaran RNN pada setiap aturan inferensi *fuzzy*  $R^s$  pada bagian anteseden, yaitu sebagai berikut:

$$(\mu_{A_s}(t)) = \hat{m}_{it}^s \quad (3.30)$$

3. Pembelajaran RNN yang berhubungan dengan bagian konsekuen (bagian *THEN*) pada aturan-aturan inferensi *fuzzy*

Dilakukan pembelajaran RNN pada masing-masing data *cluster* ke-*s* bagian *THEN* (pembelajaran RNN<sub>s</sub>). Data pada aturan ke-*s* yang direpresentasikan sebagai  $((x_{1t}^s, x_{2t}^s, \dots, x_{lt}^s), g_t^s)$  dijadikan *input* dan target dari RNN<sub>s</sub>. Pembelajaran RNN<sub>s</sub> dilakukan untuk menentukan banyak neuron lapisan tersembunyi dari masing-masing *cluster*.

4. Penyederhanaan bagian konsekuen (bagian *THEN*)

Dilakukan pembelajaran yang bertujuan untuk menyederhanakan hasil pembelajaran RNN<sub>s</sub> pada langkah 3. *Input* dan target pada langkah ini sama dengan *input* dan target pada langkah 3 yaitu berupa  $((x_{1t}^s, x_{2t}^s, \dots, x_{lt}^s), g_t^s)$ . RNN<sub>s</sub> pada langkah ini dibangun dengan neuron lapisan tersembunyi sebanyak yang telah ditentukan pada pembelajaran RNN<sub>s</sub> langkah 3. Pembelajaran pada langkah ini dilakukan dengan melakukan eliminasi *input* sehingga diperoleh *input* optimal dari masing-masing *cluster*. Variabel  $x_{it}^s$  ( $i = 1, 2, \dots, l; t = 1, 2, \dots, N_s$ ) dapat dieliminasi jika MSE *input* tanpa  $x_{it}^s$  lebih kecil daripada MSE *input* tanpa eliminasi variabel.

Langkah selanjutnya setelah menentukan *input* optimal dari masing-masing *cluster* adalah menentukan model RNF. Penentuan model RNF dilakukan secara terpisah untuk masing-masing *cluster*. Model RNF yang akan diaplikasikan untuk klasifikasi nodul kanker paru pada tugas akhir ini adalah *Fuzzy Modeling Network*

tipe III dengan konsekuen berupa persamaan linear orde pertama. Aturan untuk FMN tipe III adalah sebagai berikut:

$$R^s: IF x_t^s = (x_{1t}^s, x_{2t}^s, \dots, x_{lt}^s) \text{ is } A_s \text{ THEN } g_t^{s*} = f_s(x_{1t}^s, x_{2t}^s, \dots, x_{lt}^s) \quad (3.31)$$

dengan,

$R^s$  = aturan inferensi ke- $s$  ( $s$  adalah banyak aturan inferensi;  $s = 1, 2, \dots, r$ ),

$x_t^s$  = *input* jaringan data ke- $t$  pada aturan inferensi ke- $s$ ,

$x_{it}^s$  = variabel *input* ke- $i$  data ke- $t$  pada aturan inferensi ke- $s$ ;

$g_t^{s*}$  = *target* jaringan data ke- $t$  pada aturan inferensi ke- $s$ ,

$A_s$  = himpunan *fuzzy* bagian anteseden pada aturan inferensi ke- $s$ ,

$f_s(x_1, x_2, \dots, x_l) = p_{s0} + p_{s1}x_1 + p_{s2}x_2 + \dots + p_{sl}x_l$ ;  $p_{si}$  adalah parameter konsekuen;

$i = 1, 2, \dots, l$ .

Parameter konsekuen untuk tiap  $R^s$  ditentukan dengan menggunakan metode *Least Square Estimator* (LSE). Koefisien konsekuen yang diperoleh akan digunakan untuk menyederhanakan bagian anteseden pada pembelajaran selanjutnya. Hasil penyederhanaan yaitu  $(x_{1t}^s, x_{2t}^s, \dots, x_{lt}^s, g_t^{s*})$  kemudian digunakan sebagai *input* dan *target* pembelajaran RNN<sub>s</sub> untuk memperoleh nilai  $\mu_s(t)$  yang merupakan *output* jaringan hasil pembelajaran RNN pada setiap aturan inferensi *fuzzy*  $R^s$  pada bagian konsekuen.

##### 5. Penentuan *output* akhir

Nilai  $z_t$  akhir dapat diperoleh dengan rumus sebagai berikut:

$$z_t = \frac{\sum_{s=1}^r (\mu_{A_s}(t)) \mu_s(t)}{\sum_{s=1}^r \mu_{A_s}(t)} \quad (3.32)$$

dengan,

$z_t$  = *output* akhir data ke- $t$ ,

$(\mu_{A_s}(t))$  = *output* jaringan hasil pembelajaran RNN pada setiap aturan inferensi *fuzzy*  $R^s$  pada bagian anteseden,

$\mu_s(t)$  = *output* jaringan hasil pembelajaran pada setiap aturan inferensi *fuzzy*  $R^s$  pada bagian konsekuen,

$t$  = 1, 2, ...,  $N_s$ ;  $N_s$  adalah banyak aturan *fuzzy* ke- $s$ .

### **C. Aplikasi Model *Recurrent Neural Network* (RNN) dan Model *Recurrent Neuro Fuzzy* (RNF) untuk Klasifikasi Nodul Kanker Paru dari Citra Foto Paru**

Prosedur pemodelan baik model RNN maupun model RNF diawali dengan mendefinisikan variabel *input* dan target yang digunakan sebagai *input* dan target jaringan. Penelitian ini menggunakan 100 citra foto paru (abu-abu), yang terdiri dari 35 citra paru normal, 33 citra tumor paru jinak, dan 32 citra tumor paru ganas. Data citra foto paru diperoleh dari *digital image database* yang disediakan oleh JSRT (*Japanese Society of Radiology Technology*). Citra yang disediakan berupa citra abu-abu (*grayscale*). Citra abu-abu ini kemudian ditransformasi menjadi citra biner untuk mendapatkan citra yang lebih jelas. Hasil transformasi citra abu-abu menjadi citra biner dapat dilihat pada Lampiran 3.1. Setelah dilakukan transformasi citra, proses berikutnya adalah ekstraksi citra dengan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk mendapatkan 14 fitur ekstraksi. Fitur-fitur hasil ekstraksi citra dengan metode GLCM kemudian digunakan sebagai variable *input*. Fitur-fitur tersebut

adalah *energy* ( $X_1$ ), *contrast* ( $X_2$ ), *correlation* ( $X_3$ ), *sum of square* ( $X_4$ ), *IDM* ( $X_5$ ), *sum average* ( $X_6$ ), *sum variance* ( $X_7$ ), *sum entropy* ( $X_8$ ), *entropy* ( $X_9$ ), *difference variance* ( $X_{10}$ ), *difference entropy* ( $X_{11}$ ), *maximum probability* ( $X_{12}$ ), *homogeneity* ( $X_{13}$ ), dan *dissimilarity* ( $X_{14}$ ). Hasil ekstraksi citra yang dapat dilihat pada Lampiran 3.2 menunjukkan bahwa nilai ekstraksi  $X_2$ ,  $X_{10}$  dan  $X_{14}$  sama. Hal ini menyebabkan variabel  $X_{10}$  dan  $X_{14}$  dapat dihilangkan sehingga data yang digunakan sebagai variabel *input* adalah 12 fitur hasil ekstraksi citra tanpa  $X_{10}$  dan  $X_{14}$ . Data yang digunakan sebagai variabel target adalah stadium dari masing-masing citra foto paru. Target dan *output* jaringan berupa data numerik, yaitu angka 1 untuk paru normal, 2 untuk tumor paru jinak, dan 3 untuk tumor paru ganas. Dalam mengklasifikasi nodul kanker paru, terdapat beberapa kriteria pengelompokkan, yaitu:

1. jika *output* ( $z$ ) bernilai  $1 < y \leq 1,5$  maka dianggap 1, sehingga hasil klasifikasi normal,
2. jika *output* ( $z$ ) bernilai  $1,5 < y \leq 2,5$  maka dianggap 2, sehingga hasil klasifikasi tumor paru jinak,
3. jika *output* ( $z$ ) bernilai  $2,5 < y < 3$  maka dianggap 3, sehingga hasil klasifikasi tumor paru ganas.

Langkah berikutnya setelah mendefinisikan variabel *input* dan target adalah pembagian data. Data hasil ekstraksi citra dibagi menjadi 75 data sebagai data *training* (Lampiran 3.4) dan 25 data sebagai data *testing* (Lampiran 3.5). Jaringan dibangun menggunakan RNN algoritma *backpropagation* dengan 1 lapisan



tersembunyi. Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner (Persamaan 2.18) antara lapisan *input* dengan lapisan tersembunyi serta fungsi sigmoid bipolar (Persamaan 2.19) antara lapisan tersembunyi dengan lapisan *output*.

## **1. Aplikasi model *Recurrent Neural Network* (RNN) untuk Klasifikasi Nodul Kanker Paru**

Prosedur pemodelan RNN selanjutnya setelah mendefinisikan variabel *input* dan target serta membagi data adalah perancangan model terbaik yaitu melalui proses *trial and error* terhadap beberapa macam arsitektur model. Sebelum perancangan model, data perlu dinormalisasi agar data berdistribusi normal. Hasil normalisasi seluruh data dapat dilihat pada Lampiran 3.6. Setelah data berdistribusi normal, dilakukan perancangan model terbaik. Adapun prosedur perancangan model terbaik adalah sebagai berikut.

### a. Pemilihan banyak neuron terbaik pada lapisan tersembunyi

Pemilihan banyak *neuron* pada lapisan tersembunyi dilakukan dengan perhitungan terhadap nilai MSE hasil pembelajaran pada data *training* dan data *testing*. Parameter yang digunakan: maksimum epoh = 5000, laju pembelajaran = 1, dan toleransi eror =  $10^{-5}$ . Tabel 3.1 merupakan nilai MSE untuk setiap percobaan pada banyak *neuron* pada lapisan tersembunyi dari 1 *neuron* hingga 22 *neuron*.

**Tabel 3.1** Nilai MSE Pemilihan Banyak Neuron

Banyak Neuron	MSE <i>Training</i>	MSE <i>Testing</i>	Banyak Neuron	MSE <i>Training</i>	MSE <i>Testing</i>
1	0.5647	0.73907	12	0.34969	0.99601
2	0.49855	0.70875	13	0.35999	0.92307
3	0.46855	0.78906	14	0.31724	0.97967
4	0.44023	0.72675	15	0.34203	0.97597
5	0.38647	1.0256	16	0.34308	0.94393
6	0.36491	0.98755	17	0.37545	0.92388
7	0.38515	0.77562	18	0.33673	0.93086
8	0.34052	0.98237	19	0.32677	1.0547
9	0.36581	0.8718	20	0.34492	0.99167
10	0.34816*)	0.89514*)	21	0.34492	0.99167
11	0.32639	0.98617	22	0.30847	0.97838

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.1, MSE terkecil diperoleh pada lapisan tersembunyi dengan 10 *neuron*. Oleh karena itu, jaringan RNN akan dibangun menggunakan 10 *neuron* pada lapisan tersembunyi.

b. Eliminasi *input*

Setelah ditentukan menggunakan 10 *neuron* pada lapisan tersembunyi, dilakukan pembelajaran untuk menentukan *input* optimal. Parameter yang digunakan: maksimum epoh = 5000, laju pembelajaran = 1, dan toleransi eror =  $10^{-5}$ . Tabel 3.2 merupakan nilai MSE data *training* dan data *testing* untuk hasil percobaan terbaik dari setiap pemilihan banyak fitur *input* yang digunakan sebagai *input* jaringan dalam proses eliminasi *input*, adapun proses eliminasi secara lengkap dapat dilihat pada Lampiran 3.23 (3).

**Tabel 3.2** Nilia MSE Eliminasi *Input*

Eliminasi	<i>Input</i>	MSE <i>Training</i>	MSE <i>Testing</i>
-	12 Fitur	0.34816	0.89514
$X_8, X_{10}, X_{14}$	11 Fitur	0.34973	0.99191
$X_4, X_8, X_{10}, X_{14}$	10 Fitur	0.35946	0.92078
$X_4, X_6, X_8, X_{10}, X_{14}$	9 Fitur	0.36176	1.1866
$X_4, X_6, X_8, X_{10}, X_{11}, X_{14}$	8 Fitur	0.33742*)	0.74137*)

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.2, MSE yang diperoleh pada saat 8 fitur saja yang digunakan sebagai variabel *input* lebih sederhana daripada MSE model dengan variasi variabel *input* lainnya, sehingga eliminasi 6 fitur akan memberikan model yang lebih baik. Fitur *input* yang digunakan sebagai variable *input* adalah *energy* ( $X_1$ ), *contrast* ( $X_2$ ), *correlation* ( $X_3$ ), IDM ( $X_5$ ), *sum variance* ( $X_7$ ), *entropy* ( $X_9$ ), *maximum probability* ( $X_{12}$ ) dan *homogeneity* ( $X_{13}$ ). *Output* RNN dapat dilihat pada Lampiran 3.7. Hasil percobaan pemodelan RNN dengan menggunakan citra abu-abu sebagai *input* (tanpa ditransformasi ke citra biner) dapat dilihat pada Lampiran 3.23 (1), sedangkan nilai MSE pemodelan RNN lainnya dengan percobaan pemilihan fungsi aktivasi dan ukuran pembagian data dapat dilihat pada Lampiran 3.23 (2).

Perhitungan *output* RNN secara manual adalah sebagai berikut. Misalkan akan dihitung *output* RNN data *testing* pertama. Perhitungan *output* RNN menggunakan Persamaan (3.1). Berdasarkan pelatihan menggunakan

struktur jaringan Elman RNN dengan algoritma *backpropagation* dengan menggunakan program MatlabR2010a, diperoleh bobot bobot sebagai berikut:

$$b_k = [3.7563; 3.0789; \dots ; 5.9570; -2.211]$$

$$b_0 = 0.118248$$

$$W_{ik(x)} =$$

$$\begin{bmatrix} -0.48027 & 0.755582 & \dots & -0.43936 & 0.771386 \\ -0.31745 & 0.552705 & \dots & -0.38709 & -0.72416 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0.147367 & -0.57026 & \dots & 0.165837 & -0.04811 \\ -0.57152 & -0.44983 & \dots & -0.15277 & -0.64822 \end{bmatrix}$$

$$W_{kk(u)} =$$

$$\begin{bmatrix} -2.2216 & -1.5828 & \dots & -1.043 & 1.61991 \\ 1.18990 & 1.42461 & \dots & -1.8780 & -0.5649 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -1.5966 & -0.4611 & \dots & 1.16603 & -1.426 \\ 2.1544 & 0.74549 & \dots & -0.7265 & -1.5635 \end{bmatrix}$$

$$v =$$

$$[1.1537; -0.2513; \dots ; -0.4396; -1.018]$$

Selanjutnya bobot-bobot yang diperoleh disubstitusikan kedalam

persamaan model yaitu Persamaan (3.2) menjadi:

$$\begin{aligned} y_{in(1)}_1 &= X_1 w_{11(x)} + X_2 w_{21(x)} + X_3 w_{31(x)} + X_5 w_{41(x)} + X_7 w_{51(x)} \\ &\quad + X_9 w_{61(x)} + X_{12} w_{71(x)} + X_{13} w_{81(x)} + b_1 \\ &= (-1.06115)(-0.48027) + \dots + (-0.89916)(0.771386) + (3.756371) \\ &= 4.337235 \end{aligned}$$

Sinyal terbobot  $y_{in}(1)_1$  kemudian diaktivasi dengan fungsi aktivasi *logsig* (Persamaan 3.8), yaitu sebagai berikut:

$$y(1)_1 = f(y_{in}(1)_1) = \frac{1}{1 + e^{-(4.337235)}} = 0.987096$$

Tabel 3.3 berikut adalah perhitungan untuk  $k = 1, 2, \dots, 10$ .

**Tabel 3.3** Hitungan Manual  $y(1)_k$

$k$	$y_{in}(1)_k$	$y(1)_k = u_k$
2	6.671022	0.998735
3	4.915414	0.992721
4	4.249478	0.985929
5	4.050908	0.982891
6	2.205736	0.900763
7	2.791012	0.942188
8	4.728066	0.991234
9	3.279071	0.963704
10	3.522216	0.971313

Operasi keluaran dari lapisan tersembunyi ke- $k$  (dengan neuron tambahan konteks) menuju ke lapisan *output* yang dihitung dengan Persamaan 3.6 adalah sebagai berikut:

$$y_{in_1} = X_1w_{11(x)} + X_2w_{21(x)} + \dots + X_{13}w_{81(x)} + b_1 + u_1w_{11(u)} + u_2w_{21(u)} \\ + \dots + u_{10}w_{10_1(u)} + u_{11}w_{11_1(u)}$$

yang dapat disederhanakan menjadi

$$y_{in_1} = y_{in}(1)_1 + u_1w_{11(u)} + u_2w_{21(u)} + \dots + u_{10}w_{10_1(u)} + u_{11}w_{11_1(u)} \\ = 0,268618 + (0.987096)(-2.22161) + (0.998735)(-1.58289) + \dots \\ + (0.963704)(-1.0438) + (0.971313)(1.619913) \\ = 3,403863$$

Sinyal terbobot  $y_{in_1}$  kemudian diaktivasi dengan *logsig* (Persamaan 3.9), yaitu sebagai berikut:

$$y_1 = f(y_{in_1}) = \frac{1}{1 + e^{-(3,403863)}} = -0,93565$$

Tabel 3.4 menunjukkan perhitungan untuk  $k = 1, 2, \dots, 10$ .

**Tabel 3.4** Hitungan Manual  $y_k$

$k$	$y_{in_k}$	$y_k = f(y_{in_k})$
2	7.225536	-0.99855
3	9.566466	-0.99986
4	3.991619	-0.96373
5	-1.20701	0.539539
6	4.104963	-0.96755
7	2.932943	-0.8989
8	7.437926	-0.99882
9	-2.47397	0.844593
10	1.520731	-0.64129

Selanjutnya, berdasarkan persamaan (3.10) diperoleh operasi pada lapisan *output*:

$$\begin{aligned} z_{in} &= \sum_{k=1}^{10} v_k y_k + b_0 \\ &= v_1 y_1 + v_2 y_2 + \dots + v_{10} y_{10} + b_0 \\ &= (1.15374)(-0,93565) + (-0.25131)(-0.998546) + \dots + \\ &\quad (-1.0184)(-0.641292) + 0.118248 \\ &= -5.39 \end{aligned}$$

Sinyal terbobot  $z_{in}$  kemudian diaktivasi dengan *tansig* (Persamaan 3.12), yaitu sebagai berikut:

$$z = f(z_{in}) = \frac{1 - e^{5.39}}{1 + e^{5.39}} = -0,99092 \approx -1$$

Diperoleh nilai  $z = -1$ . Nilai tersebut masih dalam bentuk normal. Untuk mengembalikan dalam bentuk data semula, digunakan fungsi *poststd* pada

MatlabR2010a. *Output* klasifikasi data *testing* pertama adalah 2,706. Bobot-bobot hasil pembelajaran dan hitungan manual *output* RNN data *testing* secara lengkap dapat dilihat pada Lampiran 3.22.

## **2. Aplikasi Model *Recurrent Neuro Fuzzy* (RNF) untuk Klasifikasi Nodul Kanker Paru**

Langkah klasifikasi menggunakan model RNF setelah mendefinisikan variable *input* dan target serta membagi data adalah penentuan variabel *input*. Karena langkah penentuan *input* RNF sama dengan langkah perancangan model terbaik pada RNN, maka *input* RNF adalah hasil perancangan model terbaik model RNN.

Perancangan model terbaik model RNN diawali dengan menentukan neuron pada lapisan tersembunyi terlebih dahulu. Tabel 3.1 menunjukkan nilai MSE percobaan neuron 1 sampai 22. Hasil pembelajaran menunjukkan bahwa 10 neuron lapisan tersembunyi memberikan MSE paling sederhana sehingga jaringan akan dibangun menggunakan 10 neuron pada lapisan tersembunyi. Setelah ditentukan banyak neuron pada lapisan tersembunyi, langkah berikutnya adalah eliminasi variabel *input* untuk menentukan *input* optiman. Nilai MSE model yang dihasilkan selama proses percobaan eliminasi *input* dapat dilihat pada Tabel 3.2. Berdasarkan Tabel 3.2 nilai MSE paling sederhana diperoleh dengan menggunakan 8 fitur sebagai variable *input*. Fitur *input* yang berpartisipasi pada bagian anteseden (bagian *IF*) model RNF adalah *energy* ( $X_1$ ), *contrast* ( $X_2$ ), *correlation* ( $X_3$ ), *IDM* ( $X_5$ ), *sum variance* ( $X_7$ ), *entropy* ( $X_9$ ), *maximum probability* ( $X_{12}$ ), dan *homogeneity* ( $FX_{13}$ ).

Hasil dari pemodelan RNN kemudian di bagi menjadi  $r$  cluster. Dalam tugas akhir ini, hasil pemodelan RNN akan dibagi menjadi 3 cluster dengan menggunakan metode *clustering* FCM sehingga diperoleh 3 buah aturan  $R^s, s = 1,2,3$ . Berdasarkan hasil *clustering*, diperoleh kecenderungan suatu data masuk ke suatu cluster yang dapat dilihat pada Lampiran 3.8. Matriks pusat cluster pada data *training* (V) dan data *testing* ( $V_c$ ) adalah sebagai berikut:

$$V = \begin{bmatrix} 0.535799 & 0.020698 & 0.95343 & 0.99586 & 0.734434 & 0.644503 & 0.989638 & 2.699011 \\ 0.553508 & 0.019527 & 0.954229 & 0.996093 & 0.71241 & 0.674507 & 0.990233 & 1.180718 \\ 0.779835 & 0.007214 & 0.966524 & 0.998557 & 0.405818 & 0.873411 & 0.996394 & 2.181861 \end{bmatrix}$$

dan

$$V_c = \begin{bmatrix} 0.840804 & 0.004998 & 0.967656 & 0.999005 & 0.316487 & 0.912807 & 0.997511 & 2.168228 \\ 0.625931 & 0.017338 & 0.951529 & 0.996545 & 0.625875 & 0.754177 & 0.991345 & 2.228061 \\ 0.51664 & 0.020507 & 0.955713 & 0.995888 & 0.754839 & 0.615959 & 0.989736 & 1.615892 \end{bmatrix}$$

Matriks pusat cluster data *training* dan data *testing* berukuran  $3 \times 9$  dengan banyaknya baris sesuai dengan banyaknya cluster dan banyaknya kolom sesuai dengan banyaknya variabel *input* dan target *output*.

Bersamaan dengan proses *clustering* data, dilakukan pembelajaran RNN yang berhubungan dengan bagian anteseden (bagian *IF*) pada aturan-aturan inferensi *fuzzy*. *Input* pembelajaran adalah hasil pembelajaran RNN sebelumnya yaitu  $(X_{1t}, X_{2t}, X_{3t}, X_{5t}, X_{7t}, X_{9t}, X_{12t}, X_{13t})$  dan target pembelajaran adalah nilai  $m_{it}^s$  yang dapat dilihat di Lampiran 3.8. Pembelajaran dilakukan menggunakan RNN jaringan Elman dengan algoritma *backpropagation*. Hasil pembelajaran langkah ini adalah nilai keanggotaan data bagian anteseden pada setiap aturan inferensi. Nilai keanggotaan data *training* dapat dilihat pada Tabel 3.5 berikut.



**Tabel 3.5** Nilai Keanggotaan

$t$	Nilai Keanggotaan (Himpunan Fuzzy)		
	$\mu_{A_1}(t)$	$\mu_{A_2}(t)$	$\mu_{A_3}(t)$
1	0.430365	0.566431	9.81E-06
2	0.114903	0.883061	1.65E-05
3	0.003534	0.996416	4.76E-05
4	0.002311	3.91E-09	0.999696
5	0.264051	0.742263	5.82E-06
6	0.000318	0.999708	4.14E-05
.	.	.	.
.	.	.	.
.	.	.	.
74	0.966459	0.025011	0.000152
75	0.787839	0.208837	6.39E-06

Nilai keanggotaan seluruh data bagian antesenden dapat dilihat pada Lampiran 3.9. Langkah pemodelan RNF selanjutnya adalah sebagai berikut:

- Pembelajaran RNN yang berhubungan dengan bagian konsekuen (bagian *THEN*) pada aturan-aturan inferensi fuzzy.

Pembelajaran RNN<sub>s</sub> pada bagian konsekuen menggunakan *input* dan target  $((X_{1t}^s, X_{2t}^s, X_{3t}^s, X_{5t}^s, X_{7t}^s, X_{9t}^s, X_{12t}^s, X_{13t}^s), g_t^s)$  dengan  $t = 1, 2, \dots, N_s$  dan  $N_s$  adalah jumlah data yang masuk pada *cluster* ke- $s$ ). Proses pembelajaran dibagi menjadi 3 proses pembelajaran sesuai dengan banyak *cluster*, yaitu  $R^1(\text{RNN}_1)$ ,  $R^2(\text{RNN}_2)$  dan  $R^3(\text{RNN}_3)$ . Pembelajaran  $R^1(\text{RNN}_1)$  menggunakan 21 data *training* dan 6 data *testing* yang masuk *cluster* 1, pembelajaran  $R^2(\text{RNN}_2)$  menggunakan 27 data *training* dan 6 data *testing* yang masuk *cluster* 2, dan pembelajaran  $R^3(\text{RNN}_3)$  menggunakan 27 data *training* dan 13 data *testing* yang masuk *cluster* 3.

Proses pembelajaran tiap  $RNN_s$  menggunakan RNN jaringan elman dengan algoritma *backpropagation*. Parameter yang digunakan: maksimum epoch = 5000, Kinerja tujuan =  $e^{-5}$ , *learning rate* = 0,1, momentum = 0,8, dan jumlah *epoch* = 500. Pembelajaran pada bagian ini dilakukan untuk menentukan banyak neuron pada lapisan tersembunyi dari masing-masing *cluster*.

1) Pembelajaran pada  $RNN_1$

Data *training* dan data *testing* yang akan digunakan sebagai *input* dan target untuk  $R^1(RNN_1)$  dapat dilihat pada Lampiran 3.10. Tabel 3.6 adalah tabel nilai MSE hasil pembelajaran  $RNN_1$ .

**Tabel 3.7** MSE Neuron Hasil Pembelajaran  $RNN_1$

Banyak Neuron	MSE <i>Training</i>	MSE <i>Testing</i>
1	0.084584	1.4386
2	0.082235	0.91804
3	0.013455	1.4368
4	0.080304	0.87808
5	0.043104	0.58508
6	0.043411*)	0.47862*)
7	0.01348	1.4088
8	0.043093	0.63583
9	0.043114	1.177
10	0.043147	0.95223

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.6, model dengan MSE sederhana adalah model dengan 6 neuron pada lapisan tersembunyi, maka digunakan 6 neuron pada lapisan tersembunyi untuk pembelajaran pada  $RNN_1$ .

## 2) Pembelajaran pada $RNN_2$

Data *training* dan *testing* yang akan digunakan sebagai *input* dan target untuk  $R^2(RNN_2)$  dapat dilihat pada Lampiran 3.11. Tabel 3.7 menunjukkan nilai MSE dan hasil pembelajaran  $RNN_2$ .

**Tabel 3.7** MSE Neuron Hasil Pembelajaran  $RNN_2$

Banyak Neuron	MSE <i>Training</i>	MSE <i>Testing</i>
1	0.08934	1.3446
2	0.083409	1.2267
3	0.029641	1.1822
4	0.028219	1.8314
5	0.029573*)	1.2253*)
6	0.028167	1.9024
7	0.02797	1.574
8	0.027986	1.5393
9	0.027953	1.5967

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.7, model dengan MSE sederhana adalah model dengan 5 neuron pada lapisan tersembunyi, maka digunakan 5 neuron pada lapisan tersembunyi untuk pembelajaran pada  $RNN_2$ .

## 3) Pembelajaran pada $RNN_3$

Data *training* dan *testing* yang akan digunakan sebagai *input* dan target untuk  $R^3(RNN_3)$  dapat dilihat pada Lampiran 3.12. Tabel 3.8 menunjukkan nilai MSE hasil pembelajaran  $RNN_3$ .

**Tabel 3.8** MSE Neuron Hasil Pembelajaran  $RNN_3$

Banyak Neuron	MSE <i>Training</i>	MSE <i>Testing</i>
1	0.33875	2.1823
2	0.31931	2.1823
3	0.27917	2.1823
4	0.27138	2.1777
5	0.16413	1.2417
6	0.14167	0.77581
7	0.14402*)	0.74503*)
8	0.1094	2.0918
9	0.12544	1.7759
10	0.13247	2.1823

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.8, model dengan MSE sederhana adalah model dengan 7 neuron pada lapisan tersembunyi, maka digunakan 7 neuron pada lapisan tersembunyi untuk pembelajaran pada  $RNN_3$ .

b. Penyederhanaan bagian konsekuen (bagian *THEN*)

Setelah dilakukan pembelajaran pada masing-masing  $RNN_5$  untuk menentukan banyak neuron pada lapisan tersembunyi, pada bagian ini akan dilakukan penyederhanaan bagian konsekuen. Penyederhanaan dilakukan melakukan eliminasi *input* pada masing-masing *cluster*. Parameter yang digunakan antara lain maksimum epoh = 5000, Kinerja tujuan =  $e^{-5}$ , *learning rate* = 0,1, momentum = 0,8, dan jumlah *epoch* = 500.

Langkah selanjutnya setelah menentukan *input* optimal adalah menentukan model RNF. Langkah ini dilakukan secara terpisah untuk masing-masing *cluster*. Model yang digunakan adalah FMN Tipe III dengan aturan inferensi sebagai berikut:

$$IF x = (X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13}) \text{ is } A_s \text{ THEN } g_t^{s*} = f_s; s = 1, 2, 3 \quad (3.32)$$

Penyederhanaan pada tiap jaringan syaraf  $RNN_s$  adalah sebagai berikut.

1) Penyederhanaan pada  $RNN_1$

Hasil pembelajaran  $RNN_1$  adalah ditetapkan menggunakan 6 neuron pada lapisan tersembunyi pada  $RNN_1$ . Langkah selanjutnya adalah eliminasi *input* jaringan. *Input* jaringan akan dikeluarkan satu demi satu secara bergantian kemudian dipilih model yang menghasilkan MSE terkecil dengan prinsip parsimoni. Tabel 3.9 menunjukkan nilai MSE hasil eliminasi *input*.

**Tabel 3.9** MSE Eliminasi *Input*  $RNN_1$

Eliminasi	<i>Input</i>	MSE <i>Training</i>	MSE <i>Testing</i>
-	8 Fitur ( $X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, X_{13}$ )	0.043411*)	0.47862*)
$X_1$	7 Fitur	0.043395	1.4329
$X_2$	7 Fitur	0.043197	1.3933
$X_3$	7 Fitur	0.012679	1.4386
$X_5$	7 Fitur	0.043035	0.51799
$X_7$	7 Fitur	0.043058	1.2827
$X_9$	7 Fitur	0.013343	1.4386
$X_{12}$	7 Fitur	0.043056	1.1054
$X_{13}$	7 Fitur	0.04352	1.421

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.9, model dengan MSE sederhana adalah model dengan *input* jaringan tanpa eliminasi *input*. Oleh karena itu, jaringan pada  $RNN_1$  menggunakan 6 neuron pada lapisan tersembunyi dengan *input*  $X_1, X_2, X_3, X_5, X_7, X_9, X_{12}$ , dan  $X_{13}$ .

Format untuk aturan inferensi pertama ( $R^1$ ) adalah:

$$R^1: IF x = X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13} \text{ is } A_1 \text{ THEN } g^{1*} = f_1 \quad (3.33)$$

dengan  $f_1 = p_1X_1 + p_2X_2 + p_3X_3 + p_4X_5 + p_5X_7 + p_6X_9 + p_7X_{12} + p_8X_{13} + p_0$  dan  $p_0, p_1, \dots, p_8$  adalah koefisien konsekuensi dari persamaan linear. Koefisien konsekuensi dari persamaan linear dicari dengan menggunakan metode LSE. Tabel 3.10 adalah koefisien konsekuensi yang diperoleh dengan metode LSE.

**Tabel 3.10** Koefisien Konsekuensi  $RNN_1$

$p_0$	$p_1$	$p_2$	$p_3$	$p_4$
-0.0086	240.3776	-633.779	162.6688	1639.037
$p_5$	$p_6$	$p_7$	$p_8$	
-2.45426	166.5667	8.886659	-2030.76	

Berdasarkan koefisien konsekuensi yang diperoleh untuk  $RNN_1$  (Tabel 3.10), maka aturan untuk  $R^1$  menjadi

$$R^1: IF x = X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13} \text{ is } A_1 \text{ THEN } g^{1*} = f_1 \quad (3.34)$$

dengan

$$f_1 = 240.3776X_1 - 633.779X_2 + 162.6688X_3 + 1639.037X_5 - 2.45426X_7 + 166.5667X_9 + 8.886659X_{12} - 2030.76X_{13} - 0.0086 \quad (3.35)$$

Hasil penyederhanaan bagian konsekuen pada pembelajaran data *training* dan data *testing* pada  $RNN_1$  dapat dilihat pada Lampiran 3.13.

## 2) Penyederhanaan pada $RNN_2$

Hasil pembelajaran  $RNN_2$  adalah ditetapkan menggunakan 5 neuron pada lapisan tersembunyi pada  $RNN_2$ . Langkah selanjutnya adalah eliminasi *input* jaringan. *Input* jaringan akan dikeluarkan satu demi satu secara bergantian kemudian dipilih model yang menghasilkan MSE terkecil dengan prinsip parsimoni. Tabel 3.11 menunjukkan nilai MSE hasil eliminasi *input*.

**Tabel 3.11** MSE Eliminasi *Input*  $RNN_2$

Eliminasi	<i>Input</i>	MSE <i>Training</i>	MSE <i>Testing</i>
-	8 Fitur ( $X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, X_{13}$ )	0.029573*)	1.2253*)
$X_1$	7 Fitur	0.027978	1.8815
$X_2$	7 Fitur	0.028034	2.001
$X_3$	7 Fitur	0.028663	1.4868
$X_5$	7 Fitur	0.029385	1.7118
$X_7$	7 Fitur	0.034989	2.0404
$X_9$	7 Fitur	0.058831	1.4976
$X_{12}$	7 Fitur	0.028078	1.6598
$X_{13}$	7 Fitur	0.028051	1.8791

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.11, model dengan MSE sederhana adalah adalah model dengan *input* jaringan tanpa eliminasi *input*. Oleh karena itu, jaringan pada  $RNN_2$  menggunakan 5 neuron pada lapisan tersembunyi dengan *input*  $X_1, X_2, X_3, X_5, X_7, X_9, X_{12}$ , dan  $X_{13}$ .

Format untuk aturan inferensi pertama ( $R^2$ ) adalah:

$$R^2: IF x = X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13} \text{ is } A_2 \text{ THEN } g^{2*} = f_2 \quad (3.36)$$

dengan  $f_2 = p_1X_1 + p_2X_2 + p_3X_3 + p_4X_5 + p_5X_7 + p_6X_9 + p_7X_{12} + p_8X_{13} + p_0$  dan  $p_0, p_1, \dots, p_8$  adalah koefisien konsekuen dari persamaan linear. Koefisien konsekuen dari persamaan linear dicari dengan menggunakan metode LSE. Tabel 3.12 adalah koefisien konsekuen yang diperoleh dengan metode LSE.

**Tabel 3.12** Koefisien Konsekuen  $RNN_2$

$p_0$	$p_1$	$p_2$	$p_3$	$p_4$
-0.00095	9.639417	-264.298	-65.3939	-305.625
$p_5$	$p_6$	$p_7$	$p_8$	
16.01332	112.9598	-133.88	252.3052	

Berdasarkan koefisien konsekuen yang diperoleh untuk  $RNN_2$

(Tabel 3.12), maka aturan untuk  $R^2$  menjadi

$$R^2: IF x = X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13} \text{ is } A_2 \text{ THEN } g^{2*} = f_2 \quad (3.37)$$

dengan,

$$f_2 = 9.639417X_1 - 264.298X_2 + -65.3939X_3 - 305.625X_5 + 16.01332X_7 - 112.9598X_9 - 133.88X_{12} + 252.3052X_{13} - 0.00095 \quad (3.38)$$



Hasil penyederhanaan bagian konsekuen pada pembelajaran data *training* dan data *testing* pada  $RNN_2$  dapat dilihat pada Lampiran 3.14.

### 3) Penyederhanaan pada $RNN_3$

Hasil pembelajaran  $RNN_3$  adalah ditetapkan menggunakan 7 neuron pada lapisan tersembunyi pada  $RNN_3$ . Langkah selanjutnya adalah eliminasi *input* jaringan. *Input* jaringan akan dikeluarkan satu demi satu secara bergantian kemudian dipilih model yang menghasilkan MSE terkecil dengan prinsip parsimoni. Tabel 3.13 menunjukkan nilai MSE hasil eliminasi *input*.

**Tabel 3.13** MSE Eliminasi *Input*  $RNN_3$

Eliminasi	<i>Input</i>	MSE <i>Training</i>	MSE <i>Testing</i>
-	8 Fitur ( $X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, X_{13}$ )	0.14402*)	0.74503*)
$X_1$	7 Fitur	0.1739	0.99478
$X_2$	7 Fitur	0.17708	2.1751
$X_3$	7 Fitur	0.18903	2.1823
$X_5$	7 Fitur	0.15792	0.98149
$X_7$	7 Fitur	0.1446	2.1823
$X_9$	7 Fitur	0.2086	1.8859
$X_{12}$	7 Fitur	0.16286	1.3178
$X_{13}$	7 Fitur	0.2442	1.4357

Keterangan: \*) Model Terbaik

Berdasarkan Tabel 3.13, model dengan MSE sederhana adalah adalah model dengan *input* jaringan tanpa eliminasi *input*. Oleh karena itu, jaringan pada  $NN_3$  menggunakan 7 neuron pada lapisan tersembunyi dengan *input*  $X_1, X_2, X_3, X_5, X_7, X_9, X_{12}$  dan  $X_{13}$ .

Format untuk aturan inferensi pertama ( $R^3$ ) adalah:

$$R^3: IF x = X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13} \text{ is } A_3 \text{ THEN } g^{3*} = f_3 \quad (3.39)$$

dengan  $f_2 = p_1X_1 + p_2X_2 + p_3X_3 + p_4X_5 + p_5X_7 + p_6X_9 + p_7X_{12} + p_8X_{13} + p_0$  dan  $p_0, p_1, \dots, p_8$  adalah koefisien konsekuen dari persamaan linear. Koefisien konsekuen dari persamaan linear dicari dengan menggunakan metode LSE. Tabel 3.14 adalah koefisien konsekuen yang diperoleh dengan metode LSE.

**Tabel 3.14** Koefisien Konsekuen  $RNN_3$

$p_0$	$p_1$	$p_2$	$p_3$	$p_4$
8.696836	19203.33	-114527	-1420.12	-78941.6
$p_5$	$p_6$	$p_7$	$p_8$	
4305.868	48427.08	-34967.3	26710.07	

Berdasarkan koefisien konsekuen yang diperoleh untuk  $RNN_3$  (Tabel 3.14), maka aturan untuk  $R^3$  menjadi

$$R^3: IF x = X_1, X_2, X_3, X_5, X_7, X_9, X_{12}, \text{ dan } X_{13} \text{ is } A_2 \text{ THEN } g^{3*} = f_3 \quad (3.40)$$

dengan,

$$f_3 = 19203.33X_1 - 114527X_2 - 1420.12 X_3 - 78941.6X_5 + 4305.868X_7 + 48427.08X_9 - 34967.3X_{12} + 26710.07X_{13} + 8.696836 \quad (3.39)$$

Hasil penyederhanaan bagian konsekuen pada pembelajaran data *training* dan data *testing* pada  $RNN_2$  dapat dilihat pada Lampiran 3.15.

Setelah diperoleh hasil penyederhanaan dari masing-masing *cluster*, langkah selanjutnya adalah melakukan pembelajaran  $RNN_s$  dengan menggunakan *input* dan target hasil penyederhanaan masing-masing *cluster* sebagai *input* dan target jaringan. Pembelajaran dilakukan untuk memperoleh nilai  $\mu_s(t)$ . Nilai  $\mu_s(t)$  data *training* terlampir pada Lampiran 3.16 dan nilai  $\mu_s(t)$  data *testing* terlampir pada Lampiran 3.17.

c. Penentuan *output* akhir

*Output* akhir dihitung dengan Persamaan (3.31). *Output* RNF selengkapnya dapat dilihat pada Lampiran 3.18. Hasil percobaan dengan pemilihan *cluster*, kategori data dan metode inferensi yang berbeda dapat dilihat pada Lampiran 3.23 (4 dan 5). Perhitungan *output* RNF secara manual adalah sebagai berikut. Misalkan akan dihitung *output* RNF data *training* pertama dan kedua. Nilai  $\mu_{As}(t)$  dan  $\mu_s(t)$  data pertama dapat dilihat pada Tabel 3.15.

**Tabel 3.15** Nilai  $\mu_{As}(t)$  dan  $\mu_s(t)$  Data Pertama

	<i>Cluster 1</i>	<i>Cluster 2</i>	<i>Cluster 3</i>
$\mu_{As}(1)$	0.430365	0.566431	9.81E-06
$\mu_s(1)$	2.384274	1.714728	1.865298

Nilai *output* data pertama diperoleh dengan mensubstitusikan  $\mu_{As}(1)$  dan  $\mu_s(1)$  ke dalam Persamaan (3.32). Hasil dari substitusi tersebut adalah sebagai berikut.

$$\begin{aligned}
z_1 &= \frac{\sum_{s=1}^r \mu_{A_s}(1) \mu_s(1)}{\sum_{s=1}^r \mu_{A_s}(1)} \\
&= \frac{(0.430365 \times 2.384274) + (0.566431 \times 1.714728) + (9.81E - 06 \times 1.865298)}{(0.430365) + (0.566431) + (9.81E - 06)} \\
&= \frac{1.02610808 + 0.971275 + 1.83E - 05}{0.996806} = \frac{1.997401}{0.996806} = 2.003802
\end{aligned}$$

Nilai  $\mu_{A_s}(t)$  dan  $\mu_s(1)_s(t)$  data kedua dapat dilihat pada Tabel 3.16.

**Tabel 3.16** Nilai  $\mu_{A_s}(t)$  dan  $\mu_s(t)$  Data Kedua

	Cluster 1	Cluster 2	Cluster 3
$\mu_{A_s}(2)$	0.114903	0.883061	1.65E-05
$\mu_s(2)$	1.840453	1.698375	1.383215

Nilai *output* data ke-2 diperoleh dengan mensubstitusikan  $\mu_{A_s}(2)$  dan  $\mu_s(2)$  ke dalam Persamaan (3.32). Hasil dari substitusi tersebut adalah sebagai berikut.

$$\begin{aligned}
z_2 &= \frac{\sum_{s=1}^r \mu_{A_s}(2) \mu_s(2)}{\sum_{s=1}^r \mu_{A_s}(2)} \\
&= \frac{(0.114903 \times 1.840453) + (0.883061 \times 1.698375) + (1.65E - 05 \times 1.383215)}{(0.114903) + (0.883061) + (1.65E - 05)} \\
&= \frac{0.21147357 + 1.499769 + 2.28E - 05}{0.997981} = \frac{1.711265}{0.997981} = 1.714728
\end{aligned}$$

### 3. Perbandingan Ketepatan Klasifikasi Nodul Kanker Paru Menggunakan Model RNN dan Model RNF

Hasil klasifikasi pasti memiliki kesalahan dan tidak jarang dapat salah mengidentifikasi paru normal menjadi tumor paru atau sebaliknya. Oleh karena itu, perlu dilakukan tes akurasi dimana harus dideskripsikan terlebih dahulu *true positive* (TP), *false positive* (FP), *true negative* (TN) dan *false negative* (FN). Tabel 3.17

adalah perhitungan sensitivitas (Persamaan 2.41), spesifisitas (Persamaan 2.42), dan akurasi (Persamaan 2.43) berdasarkan *performance measure* model RNN (Lampiran 3.20) dan model RNF (Lampiran 3.19) adalah sebagai berikut.

**Tabel 3.17** Sensitivitas, Spesifisitas, dan Akurasi

	Data Training			Data Testing		
	Sensi-tivitas	Spesi-Fisitas	Akurasi	Sensiti-Vitas	Spesi-Fisitas	Akurasi
<b>RNN</b>	94%	56%	81,33%	80%	40%	64%
<b>RNF</b>	42%	68%	50,67%	40%	0%	24%

Berdasarkan Tabel 3.17 terlihat bahwa model RNN memberikan nilai sensitivitas, spesifisitas, dan akurasi yang lebih baik daripada model RNF baik untuk data *training* maupun data *testing* dengan menghasilkan nilai sensitivitas, spesifisitas, dan akurasi masing masing adalah 94%, 56%, dan 81,33% untuk data *training* serta 80%, 40%, dan 64% untuk *data testing*. Sensitivitas data *testing* model RNN adalah 80% artinya bahwa pasien yang memang menderita tumor paru berpeluang 80% untuk dinyatakan positif tumor dari hasil foto paru. Spesifisitas data *testing* model RNN adalah 40% artinya bahwa pasien yang tidak menderita tumor paru berpeluang 40% untuk dinyatakan negatif tumor dari hasil foto paru. Akurasi data *testing* model RNN adalah 64% artinya hasil diagnosa hanya akurat 64% baik untuk pasien yang dinyatakan positif maupun yang dinyatakan negatif. Meskipun model RNN memberikan nilai sensitivitas, spesifisitas, dan akurasi yang lebih baik daripada model RNF, tetapi model ini belum maksimal untuk digunakan sebagai indikator saat hasil negatif (paru normal) maupun hasil positif (tumor paru) karena keakuratan yang hanya mencapai 64%.

## **BAB IV PENUTUP**

### **A. Kesimpulan**

Berdasarkan hasil klasifikasi nodul kanker paru menggunakan model *Recurrent Neural Network* (RNN) dan *Recurrent Neuro Fuzzy* (RNF) dapat disimpulkan hal-hal sebagai berikut:

1. Prosedur pemodelan RNN adalah sebagai berikut:

a. Mendefinisikan variabel *input* dan target

Pemodelan RNN diawali dengan mendefinisikan variabel *input* dan target yang akan digunakan sebagai *input* dan target jaringan. Variabel *input* yang digunakan adalah hasil ekstraksi citra foto paru. Target RNN adalah keterangan kondisi dari masing-masing citra foto paru yaitu paru normal, tumor paru jinak, atau tumor paru ganas.

b. Membagi data

Data dibagi menjadi 2 yaitu data *training* dan data *testing*. Data *training* digunakan untuk pelatihan yang menghasilkan bobot-bobot jaringan sedangkan data *testing* digunakan untuk pengujian atau validasi data. Ukuran pembagian data yang digunakan adalah 75% untuk data *training* dan 25% untuk data *testing*.

c. Normalisasi data

Data dibawa dalam bentuk normal. Normalisasi data menggunakan *mean* dan standart *deviasi* yaitu dengan cara mengubah data asli menjadi data yang memiliki  $mean=0$  dan standart *deviasi*=1.

d. Perancangan model terbaik

Perancangan model terbaik dilakukan dengan menentukan banyak neuron pada lapisan tersembunyi dan menentukan *input* optimal. Penentuan banyak neuron pada lapisan tersembunyi dan penentuan *input* optimal dilakukan dengan melihat nilai MSE yang dihasilkan dari model.

e. Denormalisasi data

Data pembelajaran adalah data yang memiliki mean=0 dan standart deviasi=1. Setelah pembelajaran, data dikembalikan menjadi data asli.

2. Prosedur pemodelan RNF adalah sebagai berikut:

a. Pemodelan RNN

Dilakukan pembelajaran RNN dengan menentukan banyak neuron lapisan tersembunyi dan menentukan *input* optimal, selanjutnya hasil pemodelan RNN digunakan sebagai *input* pemodelan RNF.

b. Membagi data ke dalam *cluster*

Banyak *cluster* yang digunakan adalah 3 *cluster*. Metode yang digunakan adalah *Fuzzy C-Means* dimana proses *clustering* didasarkan nilai keanggotaan dari data.

c. Melakukan pembelajaran jaringan syaraf yang berhubungan dengan anteseden (bagian *IF*) pada aturan-aturan inferensi *fuzzy* untuk mendapatkan nilai keanggotaan setiap data pada anteseden. Pembelajaran menggunakan jaringan RNN algoritma *backpropagation*.

d. Melakukan pembelajaran jaringan syaraf yang berhubungan dengan konsekuen (bagian *THEN*) pada aturan-aturan inferensi *fuzzy*.

Pembelajaran dilakukan untuk menentukan banyak neuron pada lapisan tersembunyi dari masing-masing *cluster*.

e. Menyederhanakan bagian konsekuen dengan melakukan eliminasi *input* pada masing-masing *cluster* menggunakan RNN algoritma *backpropagation* yang bertujuan untuk memperbaiki bagian anteseden. Langkah setelah eliminasi input adalah pemodelan RNF dengan *Fuzzy Modeling Network* (FMN) Tipe III dengan konsekuen berupa persamaan linear orde pertama. Parameter konsekuen ditentukan dengan menggunakan metode *Least Square Estimator* (LSE).

3. Dihasilkan 2 model klasifikasi nodul kanker paru yaitu model RNN dan model RNF. Model RNN menghasilkan nilai sensitivitas, spesifisitas, dan akurasi masing masing adalah 94%,56%, dan 81,33% untuk data *training* serta 80%, 40%, dan 64% untuk *data testing*. Model RNF menghasilkan nilai sensitivitas, spesifisitas, dan akurasi masing masing adalah 42%, 68%, dan 50,67% untuk data *training* serta 40%, 0%, dan 24% untuk *data testing*. Model yang lebih baik adalah model RNN karena menghasilkan nilai sensitivitas, spesifisitas, dan akurasi yang lebih baik daripada model RNF. Meskipun model RNN lebih baik daripada model RNF, tetapi model ini belum maksimal untuk digunakan sebagai indikator saat hasil negatif (paru normal) maupun hasil positif (tumor paru) karena akurasi data *testing* yang hanya mencaai 64%.

## **B. Saran**

Dalam penulisan tugas akhir ini citra foto paru di ekstraksi untuk mendapatkan 14 fitur, selanjutnya citra foto paru diklasifikasi menggunakan



model RNN dan RNF. Metode ekstraksi citra yang digunakan pada tugas akhir ini adalah metode *Gray Level Co-occurrence Matrix (GLCM)*, untuk penulisan tugas akhir yang lain disarankan agar menggunakan metode ekstraksi yang lain seperti metode *BW Labelling* atau *tresholding*. Model jaringan RNN pada tugas akhir ini menggunakan jaringan Elman, pada tugas akhir yang lain dapat menggunakan jaringan Hopfield. Sistem inferensi RNF pada tugas akhir ini menggunakan model Soegen, pada tugas akhir yang lain dapat menggunakan sistem inferensi yang berbeda seperti sistem inferensi Tsukamoto.

## DAFTAR PUSTAKA








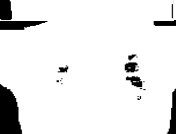


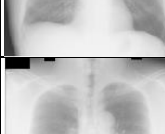

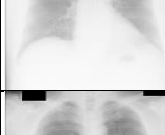





- Abdullah, Azian A. & Shaharum, Syamimi Mardiah. Lung Cancer Cell Clasification Using Artificial Neural Network. *Information Engineering Letters*, ISSN: 2160-4114, Vol. 2, No. 1.
- Agency for Toxic Substances and Disease Registry. (2013). *What is Cancer?.* 22 Maret 2014. <<http://www.atsdr.cdc.gov/COM/cancer.pdf>>.
- Al-Daoud, Essam. (2010). Cancer Diagnosis Using Modified Fuzzy Network. *Universal Journal of Computer Science and Engineering Technology*, Vol.1, No.2. Computer Science Department, Zarka University.
- American Cancer Society. (2013). *What are the risk factors for non-small cell lung cancer?.* 19 Desember 2013. <<http://www.cancer.org>>.
- American Lung Association. (2010). Lung Cancer. 19 Maret 2014. <<http://www.lungusa.org>>.
- Anami, Basavaraj S. & Burkpalli, Vishwanath C. (2009). Texture based Identification and Classification of Bulk Sugary Food Object. *ICGST-GVIP Journal*, ISSN: 1687-398X, Vol. 9, Issue 4.
- Barua, A., Mudunuri, L.S., & Kosheleva, Olga. (2014). Why Trapezoidal and Triangular Membership Function Work So Well: Towards Theoretical Explanation. *Journal of Uncertain Systems*, Vol. 8.
- Darma Putra. (2010). *Pengolahan Citra Digital*. Yogyakarta : Andi Offset.
- Deviasari, Purwati, & Yasin M. *Deteksi Kanker Paru-Paru Dari Citra Foto Rontgen Menggunakan Jaringan Saraf Tiruan Backpropagation*. UNAIR.
- Fausett, L. (1994). *Fundamentals of Neural Network (Archetectures, Algoruthms, and Applications)*. New-Jersey: Prentice-Hall.
- Felix Wicaksono (2009). *Apa itu Foto udara?.* Yogyakarta : Badan Perpustakaan dan Arsip Daerah Provinsi DIY.
- Global Bioscience. (2013). *Bagaimana Kanker Paru Ditemukan?.* 18 Desember 2013. <<http://www.cancerhelps.com>>
- Hanke, J.E. & Wichern, D.W. (2005). *Bussiness Forecasting, 8th Edition*. Upper Saddle River. New Jersey:Prentice Hall.
- Haralick, R.M., K. Shanmugam, & Its'hak Dinstein. 1973. Textural Feature for Image Classification. *IEEE Transaction on System, Man and Cybernetics*. Vol. 3. No. 6.




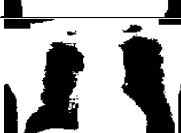
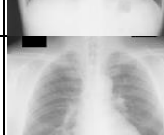

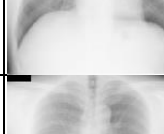

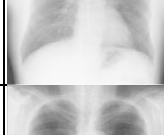



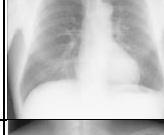





- Haykin, S. (1999). *Neural Networks & Comprehensive foundation*. New York: Prentice Hall.
- Hota H.S., Shrivasa A.K, & Singhai S.K. (2013). Artificial Neural Network, decision Tree and Statistical Techniques Applied for Designing and developing E-mail Classifier. *International Journal of Recent technology and Engineering*, Vol. 1, Issue 6.
- Ibrahim, Ahmad M. (2004). *Fuzzy Logic for Embedded Systems Applications*. Elsevier Science.
- Jaffar, et al. (2011). Genetic Fuzzy Based Automatic Lungs Segmentation from CT Scans Images. *International Journal of Innovative Computing, Information and Control (ICIC)*, Vol. 7, No. 4.
- Jang, Hoon-seok, Jin Choi, Wook, & Choi, tae-Sun (2013). Optimal Fuzzy Rule based Pulmonary Nodule Detection. *Advanced Science and Technology Letters*, Vol. 29, pp.75-78.
- Japanese Society of Radiology Technology. 1997. *Digital Image Database*. 19 Maret 2014. < <http://www.jsrt.or.jp>>.
- Jayalakshmi T. & Santhakumaran A. (2011). Statistical Normalization and Backpropagation for Classification. *International Journal of Computer Theory and Engineering*, Vol.3, No.1.
- Kalas, Mamata S. (2010). An Artificial Neural Network for Detection of Biological Early Brain Cancer. *International Journal of computer Applications*, Vol.1, No.6.
- Khalil, Rafid Ahmed & Al-Kazzaz, Sa'ad Ahme. (2009). Digital Hardware Implementation of Artificial Neurons Models using FPGA. *Al-Rafidain Enggineering*, Vol. 17, No. 2.
- Li, H., Chen, Philip C.L., & Huang, Han-Pang. (2001). *Fuzzy Neural Intelligent Systems*. New York: CRC Press.
- Lin & Lee, C.-T., & Lee, G. (1996). *Neuro Fuzzy Systems*. New Jersey: Prentice-Hall.
- Lord, Richard S. (2008). *Laboratory Evaluations for Integrative and Functional Medicine (2<sup>nd</sup> Edition)*. Duluth: Metamatrix Institute.
- Malathi A. dan Santra A.K. Diagnosis of Lung Cancer Disease using Neuro-Fuzzy Logic. *CARE Journal of Applied Research (ISSN 2321-4090)*. Department of CSE, M.S. Ramaiah College, Bangalore.



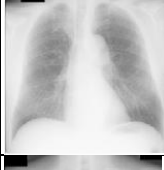











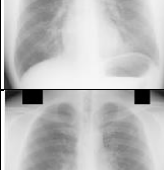


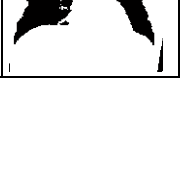
- Mandal, S., Coudhury, J.P., & Chauduri, S.R.B. (2010). In Search of Suitable Fuzzy Membership Function in Prediction of Time Series Data. *IJCSI International Journal of Computer Science Issues*, Vol 9, No.3.
- MathWorks. (1999). *Fuzzy Logic Toolbox for Use with Matlab (User's Guide Version 2)*. Natick: The MathWorks, Inc.
- Mendel, Jerry M. (1995). *Fuzzy Logic Systems for Engineering: A Tutorial. Proceeding of the IEEE*, Vol. 83, No.3.
- Mohanaiah,P., Sathyanarayana,P., & GuruKumar, L. (2013). Image Texture Feature Extraction Using GLCM Approach. *International Journal of Scientific and Research Publications*, Vol. 3, Issue 5.
- Osborn, Carol E. (2006). *Statistical Applications for Health Information Management (Second Edition)*. Missisauga: Jones and Bartlett Publisher, Inc.
- Otto, Shirley E. (2005). *Pocket Guide to Oncology Nursing*. (Alih bahasa: Jane Freyena Budi). Jakarta : EGC.
- Perhimpunan Dokter Paru Indonesia. (2003). *Kanker Paru : Pedoman Diagnosis dan Penatalaksanaan di Indonesia*. Jakarta: Perhimpunan Dokter Paru Indonesia.
- Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications (3 ed.)*. Chichester: John Wiley & Sons Inc.
- Sarle, Warren S. (1994). Neural Networks and Statistical Models. *Proceedings of the Nineteenth Annual SAS Users Group International Conference*.
- Sharma, Minakshi & Mukharjee, Sourabh. (2013). Artificial Neural Network Fuzzy Inference System (ANFIS) for Brain Tumor Detection. *Advances in Intelligent Systems and Computing*. Vol. 177. Hal. 329-339.
- Siang. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta : Penerbit Andi.
- Sianipar H.S. (2013). *Matlab untuk Pemrosesan Citra Digital* . Informatika: Bandung.
- Sivakumar, S. & Chandrasekar, C. (2013) Lung Nodule Detection Using Fuzzy Clustering and Support Vector Machines. *International Journal of Engineering and Technology (IJET)*, Vol. 5, No.1, pp. 179-185.
- Soh, Leen-Kiat. 1999. Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-Occurrence Matrices. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37, No.2.












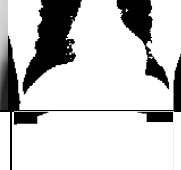
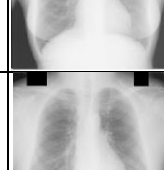
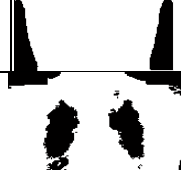





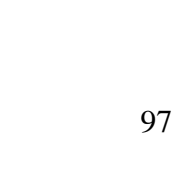
- Spitalnic, Stuart. (2004). *Test properties I: Sensitivity, Specificity, and Predictive Values*. Wayne: Turner White Communications Inc.
- Sri Kusumadewi dan Sri Hartati. (2010). *Neuro-Fuzzy: Integrasi Sistem Fuzzy & Jaringan Syaraf*. Penerbit Graha Ilmu : Yogyakarta
- Suyanto, S. M. (2008). *Soft Computing Membangun Mesin Ber-IQ Tinggi*. Bandung: Informatika.
- Udeshani, K.A.G., Meegans, R.G.N., & Fernando, T.G.I. (2011). Statistical Feature-based Neural Network Approach for the Detection of Lung Cancer in Chest X-Ray Images. *International Journal of Image Processing (IJIP)*, Vol. 5.
- Varalakshmi K. (2013). Classification of Lung Cancer Nodules using a Hybrid Approach. *Journal of Emerging Trends in Computing and Information Sciences*. Vol. 4, No. 1. Department of Computer Science and Engineering SRM University.
- Vieira, J.A.B., *et al.* (2003). Neuro-Fuzzy Systems : A Survey. *Intelligent Components and Instruments for Control Applications*. Elsevier Science.
- Wang, Li-Xing. (1997). *A Course in Fuzzy Systems and Control*. New Jersey: Prentice Hall International, Inc.
- Yeung, Daniel S. *et al.* (2010). *Sensitivity Analysis for Neural Network*. Berlin: Springer.
- Zadeh, Lotfi A. (1996). *Fuzzy Sets and Information Granularity, from Advances in fuzzy Systems-Applications Theory Vol.16. Fuzzy Sets, Fuzzy Logic and Fuzzy Systems. Selected Papers by Lotfi A. Zadeh*. Amsterdam: World Scientific.
- Zhao, Bhinsheng. *et al.* (2003). Automatic Detection of Small Lung Nodules on CT Utilizing A Local Density Maximum Algorithm. *Journal of Applied Clinical Medical Physics*, Vol. 4, No. 3.
- Zhou, Zhi-Hua, *et al.* (2002). Lung Cancer Cell Classification Based on Artificial Neural Network Ensembles. *Artificial Intelligence in Medicine*, Vol. 24. No.1, pp. 25-36.
- Zhu, W., Zeng, N., & Wang, N. (2010). Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS Implementation. *Health Care and Life Science, NESUG (SAS Global Forum)*.
- Zimmermann. (1991). *Fuzzy Sets Theory and Its Applications (2 ed.)*. Massachusetts: Kluwer Academic Publishers.




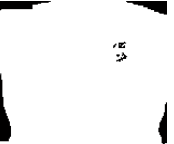












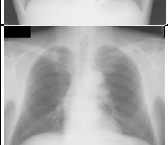

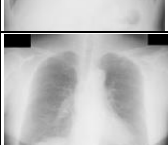

**Lampiran 3.1** Citra Foto Toraks (Asli dan Biner)





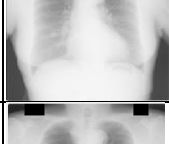

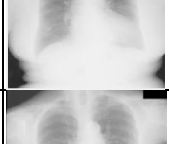


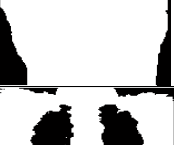




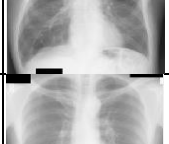
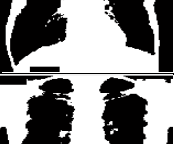
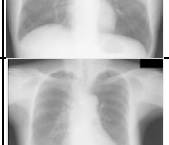

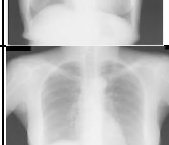

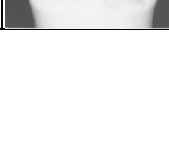

No	Klasifikasi	Citra Asli	Citra Biner
1	Normal		
2	Normal		
3	Normal		
4	Normal		
5	Normal		
6	Normal		
7	Normal		
8	Normal		
9	Normal		

10	Normal		
11	Normal		
12	Normal		
13	Normal		
14	Normal		
15	Normal		
16	Normal		
17	Normal		
18	Normal		




















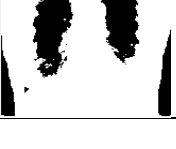
19	Normal		
20	Normal		
21	Normal		
22	Normal		
23	Normal		
24	Normal		
25	Normal		
26	<i>Benign</i>		
27	<i>Benign</i>		











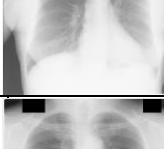



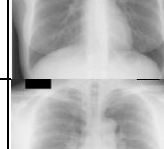

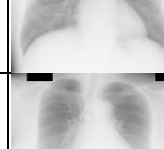
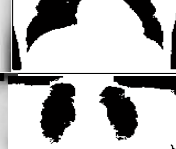

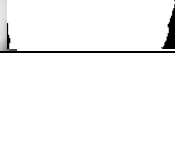
28	<i>Benign</i>		
29	<i>Benign</i>		
30	<i>Benign</i>		
31	<i>Benign</i>		
32	<i>Benign</i>		
33	<i>Benign</i>		
34	<i>Benign</i>		
35	<i>Benign</i>		
36	<i>Benign</i>		
37	<i>Benign</i>		











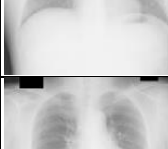







38	<i>Benign</i>		
39	<i>Benign</i>		
40	<i>Benign</i>		
41	<i>Benign</i>		
42	<i>Benign</i>		
43	<i>Benign</i>		
44	<i>Benign</i>		
45	<i>Benign</i>		
46	<i>Benign</i>		
47	<i>Benign</i>		









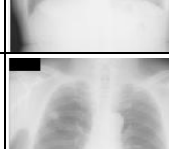

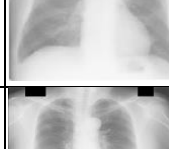



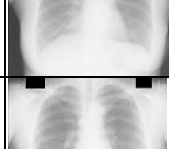



48	<i>Benign</i>		
49	<i>Benign</i>		
50	<i>Benign</i>		
51	<i>Benign</i>		
52	<i>Malignant</i>		
53	<i>Malignant</i>		
54	<i>Malignant</i>		
55	<i>Malignant</i>		
56	<i>Malignant</i>		
57	<i>Malignant</i>		
58	<i>Malignant</i>		







59	<i>Malignant</i>		
60	<i>Malignant</i>		
61	<i>Malignant</i>		
62	<i>Malignant</i>		
63	<i>Malignant</i>		
64	<i>Malignant</i>		
65	<i>Malignant</i>		
66	<i>Malignant</i>		
67	<i>Malignant</i>		
68	<i>Malignant</i>		

69	<i>Malignant</i>		
70	<i>Malignant</i>		
71	<i>Malignant</i>		
72	<i>Malignant</i>		
73	<i>Malignant</i>		
74	<i>Malignant</i>		
75	<i>Malignant</i>		
76	Normal		
77	Normal		
78	Normal		

79	Normal		
80	Normal		
81	Normal		
82	Normal		
83	Normal		
84	Normal		
85	Normal		
86	<i>Benign</i>		
87	<i>Benign</i>		

88	<i>Benign</i>		
89	<i>Benign</i>		
90	<i>Benign</i>		
91	<i>Benign</i>		
92	<i>Benign</i>		
93	<i>Malignant</i>		
94	<i>Malignant</i>		
95	<i>Malignant</i>		
96	<i>Malignant</i>		

97	<i>Malignant</i>	
98	<i>Malignant</i>	
99	<i>Malignant</i>	
100	<i>Malignant</i>	

**Lampiran 3.2 Hasil Ekstrasi (GLCM)**

No.	Energy (X1)	Contrast (X2)	Correlation (X3)	Sum of Square (X4)	IDM (X5)	Sum Average (X6)	Sum Variance (X7)	Sum Entropy (X8)	Entropy (X9)	Diff. Variance (X10)	Diff Entropy (X11)	Max Prob (X12)	Homogeneity (X13)	Dissimilarity (X14)	TARGET
1	0.5495	0.0196	0.9545	2.2754	0.9961	3.3707	7.9526	0.7043	0.7179	0.0196	0.0966	0.6755	0.9902	0.0196	1
2	0.5522	0.0184	0.9571	2.2844	0.9963	3.375	8.0032	0.6988	0.7114	0.0184	0.0919	0.6783	0.9908	0.0184	1
3	0.5578	0.0181	0.9573	2.2946	0.9964	3.3887	8.1023	0.6921	0.7044	0.0181	0.0906	0.6853	0.9909	0.0181	1
4	0.7324	0.0094	0.9637	2.6811	0.9981	3.6953	10.9083	0.4702	0.4767	0.0094	0.0531	0.843	0.9953	0.0094	1
5	0.5151	0.0204	0.9562	2.1427	0.9959	3.2647	7.2724	0.7423	0.7564	0.0204	0.0994	0.6222	0.9898	0.0204	1
6	0.5968	0.018	0.9533	2.4095	0.9964	3.4785	8.7578	0.6492	0.6617	0.018	0.0902	0.7302	0.991	0.018	1
7	0.5306	0.0223	0.9502	2.2158	0.9955	3.3236	7.6003	0.7299	0.7453	0.0223	0.1068	0.6507	0.9889	0.0223	1
8	0.601	0.0209	0.9447	2.4285	0.9958	3.4929	8.816	0.6504	0.6649	0.0209	0.1016	0.736	0.9895	0.0209	1
9	0.5526	0.0197	0.9541	2.2834	0.9961	3.379	8.0085	0.701	0.7145	0.0197	0.0967	0.6797	0.9902	0.0197	1
10	0.7651	0.0097	0.9569	2.7386	0.9981	3.7412	11.4096	0.4293	0.436	0.0097	0.0547	0.8657	0.9951	0.0097	1
11	0.537	0.0197	0.9556	2.2315	0.9961	3.3355	7.7205	0.7177	0.7314	0.0197	0.0968	0.6579	0.9902	0.0197	1
12	0.5597	0.0178	0.958	2.3017	0.9964	3.3927	8.1367	0.6893	0.7015	0.0178	0.0892	0.6875	0.9911	0.0178	1
13	0.5821	0.0223	0.9437	2.3821	0.9955	3.4559	8.5087	0.674	0.6895	0.0223	0.1069	0.7168	0.9888	0.0223	1
14	0.5454	0.0173	0.9604	2.2591	0.9965	3.3534	7.8794	0.7036	0.7152	0.0173	0.0874	0.668	0.9913	0.0173	1
15	0.5123	0.0227	0.9512	2.1388	0.9955	3.2625	7.2204	0.7501	0.7658	0.0227	0.1084	0.6199	0.9886	0.0227	1
16	0.5158	0.0201	0.9567	2.1476	0.996	3.2665	7.2871	0.741	0.7549	0.0201	0.0985	0.6232	0.9899	0.0201	1
17	0.5718	0.0201	0.9508	2.3471	0.996	3.4277	8.3409	0.6811	0.695	0.0201	0.0984	0.7038	0.99	0.0201	1
18	0.5681	0.0175	0.9578	2.3297	0.9965	3.4129	8.2828	0.6796	0.6917	0.0175	0.0881	0.6977	0.9913	0.0175	1
19	0.4842	0.0229	0.9535	1.9584	0.9954	3.1148	6.4166	0.7797	0.7955	0.0229	0.1092	0.546	0.9885	0.0229	1
20	0.5364	0.0189	0.9576	2.2253	0.9962	3.3314	7.7083	0.7166	0.7297	0.0189	0.0937	0.6563	0.9906	0.0189	1
21	0.52	0.0205	0.9554	2.165	0.9959	3.2833	7.381	0.7374	0.7516	0.0205	0.1	0.6314	0.9897	0.0205	1
22	0.8368	0.0068	0.9567	2.8504	0.9986	3.8289	12.5905	0.324	0.3286	0.0068	0.0406	0.9111	0.9966	0.0068	1
23	0.5127	0.0254	0.9452	2.1536	0.9949	3.2737	7.2445	0.7548	0.7724	0.0254	0.1182	0.6242	0.9873	0.0254	1
24	0.6886	0.014	0.9528	2.6078	0.9972	3.6363	10.2018	0.5346	0.5443	0.014	0.0739	0.8111	0.993	0.014	1

25	0.5286	0.0177	0.961	2.1892	0.9965	3.3032	7.5512	0.7223	0.7345	0.0177	0.089	0.6427	0.9911	0.0177	1
26	0.6022	0.0194	0.9488	2.4276	0.9961	3.4924	8.8407	0.6459	0.6594	0.0194	0.0957	0.7365	0.9903	0.0194	2
27	0.5374	0.0165	0.963	2.2225	0.9967	3.3276	7.7265	0.7103	0.7217	0.0165	0.0842	0.6555	0.9917	0.0165	2
28	0.7918	0.0049	0.9759	2.7754	0.999	3.7703	11.8837	0.3818	0.3852	0.0049	0.031	0.8827	0.9975	0.0049	2
29	0.4853	0.0218	0.9558	1.9594	0.9956	3.1154	6.4368	0.7762	0.7912	0.0218	0.105	0.5468	0.9891	0.0218	2
30	0.8922	0.0035	0.9662	2.9256	0.9993	3.8896	13.5857	0.2317	0.234	0.0035	0.0234	0.9431	0.9982	0.0035	2
31	0.53	0.0206	0.9542	2.2077	0.9959	3.3167	7.5848	0.7271	0.7414	0.0206	0.1004	0.648	0.9897	0.0206	2
32	0.7884	0.0082	0.9596	2.7752	0.9984	3.7701	11.7909	0.3949	0.4006	0.0082	0.0476	0.881	0.9959	0.0082	2
33	0.5496	0.0195	0.9549	2.2757	0.9961	3.3706	7.9548	0.7038	0.7173	0.0195	0.096	0.6756	0.9903	0.0195	2
34	0.7495	0.0059	0.976	2.7053	0.9988	3.7146	11.2084	0.4397	0.4437	0.0059	0.036	0.8544	0.9971	0.0059	2
35	0.675	0.0161	0.948	2.5854	0.9968	3.6178	9.9827	0.5551	0.5662	0.0161	0.0824	0.8009	0.992	0.0161	2
36	0.6454	0.0159	0.953	2.5218	0.9968	3.5675	9.5297	0.5897	0.6007	0.0159	0.0818	0.7758	0.992	0.0159	2
37	0.7934	0.0054	0.9734	2.779	0.9989	3.773	11.9031	0.381	0.3847	0.0054	0.0334	0.8838	0.9973	0.0054	2
38	0.5596	0.0194	0.9539	2.3076	0.9961	3.3967	8.1334	0.6929	0.7064	0.0194	0.0958	0.6886	0.9903	0.0194	2
39	0.8594	0.0065	0.9518	2.8817	0.9987	3.8553	12.9744	0.2899	0.2944	0.0065	0.039	0.9244	0.9968	0.0065	2
40	0.6875	0.0123	0.9592	2.6018	0.9975	3.6318	10.1973	0.532	0.5405	0.0123	0.0661	0.8098	0.9939	0.0123	2
41	0.7868	0.0053	0.9747	2.7678	0.9989	3.7642	11.7992	0.3895	0.3932	0.0053	0.0328	0.8795	0.9974	0.0053	2
42	0.5231	0.0198	0.9568	2.1769	0.996	3.2917	7.4441	0.7326	0.7463	0.0198	0.0972	0.6359	0.9901	0.0198	2
43	0.8003	0.0046	0.9764	2.7888	0.9991	3.7809	12.0232	0.3695	0.3727	0.0046	0.0293	0.8882	0.9977	0.0046	2
44	0.6244	0.018	0.9497	2.4791	0.9964	3.533	9.1948	0.6181	0.6305	0.018	0.0901	0.7575	0.991	0.018	2
45	0.812	0.0055	0.9699	2.8086	0.9989	3.7968	12.1986	0.356	0.3599	0.0055	0.0341	0.8957	0.9973	0.0055	2
46	0.5112	0.0215	0.954	2.1285	0.9957	3.2542	7.1908	0.7488	0.7637	0.0215	0.104	0.6163	0.9892	0.0215	2
47	0.6559	0.0152	0.9537	2.5429	0.997	3.5847	9.6957	0.5759	0.5865	0.0152	0.0789	0.7847	0.9924	0.0152	2
48	0.5631	0.0236	0.943	2.3313	0.9953	3.415	8.1896	0.6974	0.7138	0.0236	0.1117	0.6957	0.9882	0.0236	2
49	0.5119	0.0173	0.9633	2.1125	0.9965	3.2404	7.1827	0.7388	0.7508	0.0173	0.0874	0.6115	0.9913	0.0173	2
50	0.79	0.0057	0.9723	2.7734	0.9989	3.7689	11.8448	0.3864	0.3903	0.0057	0.0349	0.8816	0.9972	0.0057	2
51	0.7816	0.0081	0.9613	2.7634	0.9984	3.7611	11.6841	0.4038	0.4095	0.0081	0.0472	0.8765	0.9959	0.0081	2
52	0.7839	0.0059	0.9719	2.7637	0.9988	3.7612	11.7446	0.3952	0.3993	0.0059	0.0362	0.8777	0.997	0.0059	3

53	0.6087	0.017	0.9547	2.4431	0.9966	3.5008	8.9521	0.6337	0.6451	0.017	0.0861	0.7419	0.9915	0.017	3
54	0.6805	0.0129	0.9579	2.5895	0.9974	3.6217	10.0858	0.5419	0.5508	0.0129	0.0691	0.8044	0.9935	0.0129	3
55	0.4781	0.024	0.9519	1.8833	0.9952	3.0555	6.1142	0.7881	0.8047	0.024	0.1131	0.5158	0.988	0.024	3
56	0.4817	0.0232	0.9533	1.9288	0.9954	3.0935	6.3077	0.7827	0.7988	0.0232	0.1101	0.5352	0.9884	0.0232	3
57	0.6786	0.0102	0.9674	2.5802	0.998	3.6142	10.0781	0.5376	0.5447	0.0102	0.0567	0.802	0.9949	0.0102	3
58	0.7033	0.0064	0.9779	2.621	0.9987	3.6475	10.493	0.4982	0.5025	0.0064	0.0387	0.8206	0.9968	0.0064	3
59	0.7419	0.0078	0.9689	2.6955	0.9984	3.7066	11.0693	0.4544	0.4599	0.0078	0.0456	0.8494	0.9961	0.0078	3
60	0.8938	0.0032	0.9686	2.9255	0.9994	3.8911	13.6179	0.2284	0.2306	0.0032	0.0218	0.9439	0.9984	0.0032	3
61	0.5129	0.0224	0.9518	2.1414	0.9955	3.2638	7.2326	0.7489	0.7644	0.0224	0.1073	0.6207	0.9888	0.0224	3
62	0.5102	0.0233	0.9502	2.1334	0.9953	3.2566	7.1768	0.7534	0.7695	0.0233	0.1104	0.6167	0.9884	0.0233	3
63	0.6863	0.0146	0.9512	2.6042	0.9971	3.6336	10.1638	0.5385	0.5486	0.0146	0.0763	0.8095	0.9927	0.0146	3
64	0.5397	0.0197	0.9552	2.2431	0.9961	3.3436	7.7719	0.715	0.7287	0.0197	0.097	0.6619	0.9901	0.0197	3
65	0.5413	0.0186	0.9578	2.2431	0.9963	3.345	7.8013	0.7108	0.7237	0.0186	0.0925	0.6632	0.9907	0.0186	3
66	0.5103	0.0235	0.9497	2.135	0.9953	3.2577	7.18	0.7537	0.77	0.0235	0.1112	0.6171	0.9883	0.0235	3
67	0.5416	0.0213	0.9513	2.2536	0.9957	3.3535	7.8092	0.7162	0.731	0.0213	0.1031	0.6661	0.9893	0.0213	3
68	0.6078	0.0197	0.9473	2.4423	0.9961	3.5042	8.9285	0.6402	0.6538	0.0197	0.0967	0.7422	0.9902	0.0197	3
69	0.692	0.0131	0.9555	2.6126	0.9974	3.6402	10.2592	0.5286	0.5377	0.0131	0.0699	0.8135	0.9934	0.0131	3
70	0.8084	0.0047	0.9749	2.8015	0.9991	3.7913	12.1517	0.3587	0.362	0.0047	0.0298	0.8933	0.9977	0.0047	3
71	0.4882	0.0271	0.9441	2.0264	0.9946	3.1711	6.6425	0.7839	0.8027	0.0271	0.1247	0.572	0.9864	0.0271	3
72	0.7516	0.0103	0.9567	2.7169	0.9979	3.7236	11.1949	0.4481	0.4552	0.0103	0.0574	0.8566	0.9948	0.0103	3
73	0.515	0.0209	0.9549	2.146	0.9958	3.2666	7.2737	0.7436	0.7581	0.0209	0.1017	0.6228	0.9895	0.0209	3
74	0.619	0.0169	0.9535	2.4625	0.9966	3.5209	9.1157	0.6219	0.6336	0.0169	0.0858	0.752	0.9915	0.0169	3
75	0.5417	0.0197	0.955	2.2504	0.9961	3.3495	7.8102	0.7128	0.7265	0.0197	0.097	0.6649	0.9901	0.0197	3
76	0.4794	0.022	0.9559	1.8667	0.9956	3.042	6.0806	0.7827	0.7979	0.022	0.1057	0.51	0.989	0.022	1
77	0.5373	0.0205	0.9537	2.235	0.9959	3.3389	7.728	0.7191	0.7333	0.0205	0.1	0.6592	0.9897	0.0205	1
78	0.6585	0.0139	0.9576	2.5469	0.9972	3.5868	9.7429	0.57	0.5796	0.0139	0.0732	0.7865	0.9931	0.0139	1
79	0.8616	0.0044	0.9672	2.882	0.9991	3.8556	13.0371	0.2816	0.2846	0.0044	0.0282	0.9256	0.9978	0.0044	1
80	0.494	0.0193	0.9603	2.0132	0.9961	3.1613	6.7117	0.7619	0.7753	0.0193	0.0955	0.571	0.9903	0.0193	1

81	0.4923	0.0229	0.9529	2.0257	0.9954	3.1709	6.7063	0.7713	0.7871	0.0229	0.109	0.574	0.9886	0.0229	1
82	0.5401	0.0184	0.9584	2.2404	0.9963	3.3412	7.7799	0.7116	0.7243	0.0184	0.0917	0.6614	0.9908	0.0184	1
83	0.5751	0.019	0.9532	2.3524	0.9962	3.4331	8.3999	0.6751	0.6882	0.019	0.0941	0.707	0.9905	0.019	1
84	0.6706	0.0155	0.9506	2.5728	0.9969	3.6098	9.9188	0.5592	0.5698	0.0155	0.08	0.7971	0.9922	0.0155	1
85	0.5185	0.0209	0.9548	2.1602	0.9958	3.279	7.3493	0.7397	0.7542	0.0209	0.1014	0.6291	0.9896	0.0209	1
86	0.8712	0.0047	0.9624	2.8955	0.9991	3.867	13.1984	0.2677	0.2709	0.0047	0.0297	0.9312	0.9977	0.0047	2
87	0.52	0.0183	0.9604	2.1595	0.9963	3.2755	7.3717	0.7327	0.7452	0.0183	0.0914	0.6286	0.9908	0.0183	2
88	0.5396	0.0207	0.9529	2.245	0.9959	3.346	7.7704	0.7172	0.7316	0.0207	0.1009	0.6627	0.9896	0.0207	2
89	0.8311	0.005	0.9694	2.8374	0.999	3.8199	12.5169	0.3279	0.3314	0.005	0.0315	0.9074	0.9975	0.005	2
90	0.8235	0.0055	0.9681	2.8269	0.9989	3.8111	12.3862	0.3399	0.3436	0.0055	0.0339	0.9028	0.9973	0.0055	2
91	0.6389	0.0176	0.9487	2.5105	0.9965	3.559	9.4214	0.6007	0.6129	0.0176	0.0887	0.7707	0.9912	0.0176	2
92	0.5202	0.0198	0.9569	2.1644	0.996	3.2816	7.3821	0.7358	0.7495	0.0198	0.0974	0.6309	0.9901	0.0198	2
93	0.597	0.0174	0.9549	2.4096	0.9965	3.4777	8.7632	0.6477	0.6598	0.0174	0.0878	0.7301	0.9913	0.0174	3
94	0.8329	0.0054	0.9667	2.84	0.9989	3.8225	12.5405	0.3264	0.3301	0.0054	0.0335	0.9085	0.9973	0.0054	3
95	0.6047	0.0217	0.9421	2.4393	0.9957	3.5019	8.8737	0.6475	0.6626	0.0217	0.1045	0.7401	0.9892	0.0217	3
96	0.5183	0.0202	0.9562	2.1577	0.996	3.2763	7.3434	0.7385	0.7526	0.0202	0.0989	0.628	0.9899	0.0202	3
97	0.8418	0.0041	0.9733	2.8526	0.9992	3.8317	12.7048	0.3102	0.3131	0.0041	0.0267	0.9138	0.9979	0.0041	3
98	0.6723	0.0137	0.9565	2.5743	0.9973	3.6095	9.9554	0.5533	0.5628	0.0137	0.0723	0.7979	0.9932	0.0137	3
99	0.4941	0.0239	0.9504	2.0451	0.9952	3.1868	6.7745	0.7715	0.7881	0.0239	0.113	0.5814	0.988	0.0239	3
100	0.4997	0.0222	0.9536	2.0702	0.9956	3.2069	6.9111	0.7623	0.7777	0.0222	0.1066	0.5923	0.9889	0.0222	3

### Lampiran 3.3 M-File Ekstraksi Fitur Citra

#### 1. Pengubah Tipe Gambar dan Ekstraksi Gambar

```
function varargout = gldm_baru_devi(varargin)
% m-file ini digunakan untuk gldm_baru_devi.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @gldm_baru_devi_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @gldm_baru_devi_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function gldm_baru_devi_OpeningFcn(hObject, eventdata, handles,
varargin)
% mendefinisikan tombol ubah, proses, reset
set(handles.pushbutton_ubah, 'enable', 'off');
set(handles.pushbutton_proses, 'enable', 'off');
set(handles.pushbutton_reset, 'enable', 'off');
set(handles.uitable_fitur, 'enable', 'off');
handles.output = hObject;
guidata(hObject, handles);
function varargout = gldm_baru_devi_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
function pushbutton_keluar_Callback(hObject, eventdata, handles)
function pushbutton_buka_Callback(hObject, eventdata, handles)
    [FileName, PathName] = uigetfile('*.jpg');
if isempty(FileName)
    return
end
global I;
Filedata=[PathName,FileName];
I=imread(Filedata);
I=imresize(I, [256 256]); % ukuran citra foto input disamakan
axes(handles.axes_warna);
imshow(I); % menampilkan citra foto input
set(handles.edit_alamat, 'String', [PathName FileName]);
set(handles.pushbutton_ubah, 'enable', 'on');
set(handles.pushbutton_reset, 'enable', 'on');

function edit_alamat_Callback(hObject, eventdata, handles)
function edit_alamat_CreateFcn(hObject, eventdata, handles)
```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Aturan pada tombol_ubah.
function pushbutton_ubah_Callback(hObject, eventdata, handles)
global I;
global J;
level = graythresh(I);
J=im2bw(I,level); % Mengubah citra foto ke citra biner
axes(handles.axes_gray);
imshow(J); % menampilkan citra biner (citra foto input yang telah
diubah menjadi citra biner)
set(handles.pushbutton_proses,'enable','on');
set(handles.uitable_fitur,'enable','on');
% --- Aturan pada tombol_proses.
function pushbutton_proses_Callback(hObject, eventdata, handles)
global J;
global P;
GLCM=graycomatrix(J); % Ekstraksi dengan metode GLCM
stats_1 = GLCM_Features1(GLCM,0);
P(1,:)=stats_1.energ;% Energy
P(2,:)=stats_1.contr;% Contrast
P(3,:)=stats_1.corrm;% Correlation
P(4,:)=stats_1.sosvh;% Sum of Squares
P(5,:)=stats_1.idmnc;% Inverse difference moment
P(6,:)=stats_1.savgh;% Sum average
P(7,:)=stats_1.svarh;% Sum Variance
P(8,:)=stats_1.senth;% Sum Entropy
P(9,:)=stats_1.entro;% Entropy
P(10,:)=stats_1.dvarh;% Difference variance
P(11,:)=stats_1.denth;% Difference Entropy
P(12,:)=stats_1.maxpr;% Information measure of Correlation1
P(13,:)=stats_1.homom;% Information measure of Correlation2
P(14,:)=stats_1.dissi;% Dissimilarity
nama={'Energy','Contrast','Correlation','Sum of
Squares','IDM','Sum average','Sum Variance','Sum
Entropy','Entropy','Diff. Variance',...
'Diff. Entropy','Max.
Probability','Homogeneity','Dissimilarity'};
set(handles.uitable_fitur,'Data',P,'ColumnName',nama);

function axes_warna_CreateFcn(hObject, eventdata, handles)
axis off
function axes_gray_CreateFcn(hObject, eventdata, handles)
axis off
% --- Aturan pada tombol_reset
function pushbutton_reset_Callback(hObject, eventdata, handles)
global I;
global J;
I=[];
J=[];
axes(handles.axes_warna);
imshow(I);
axes(handles.axes_gray);
imshow(J);

```

```

set(handles.uitable_fitur, 'Data', '', 'ColumnName', '');
set(handles.pushbutton_ubah, 'enable', 'off');
set(handles.pushbutton_proses, 'enable', 'off');
set(handles.pushbutton_reset, 'enable', 'off');
set(handles.uitable_fitur, 'enable', 'off');
set(handles.edit_alamat, 'String', '');

% --- Aturan pada tombol save.
function pushbutton_save_Callback(hObject, eventdata, handles)
global P;
filename = 'Fitur.xlsx';
xlswrite(filename, [P]); %export hasil ekstraksi ke Ms.Excel

2. GLCM Ekstraksi Gambar
function [out] = GLCM_Features1(glcmin, pairs)
if ((nargin > 2) || (nargin == 0))
    error('Too many or too few input arguments. Enter GLCM and pairs.');
```

```

elseif ( (nargin == 2) )
    if ((size(glcmin,1) <= 1) || (size(glcmin,2) <= 1))
        error('The GLCM should be a 2-D or 3-D matrix.');
```

```

    elseif ( size(glcmin,1) ~= size(glcmin,2) )
        error('Each GLCM should be square with NumLevels rows and NumLevels cols');
```

```

    end
elseif (nargin == 1) % only GLCM is entered
    pairs = 0; % default is numbers and input 1 for percentage
    if ((size(glcmin,1) <= 1) || (size(glcmin,2) <= 1))
        error('The GLCM should be a 2-D or 3-D matrix.');
```

```

    elseif ( size(glcmin,1) ~= size(glcmin,2) )
        error('Each GLCM should be square with NumLevels rows and NumLevels cols');
```

```

    end
end

format long e
if (pairs == 1)
    newn = 1;
    for nglcm = 1:2:size(glcmin,3)
        glcm(:, :, newn) = glcmin(:, :, nglcm) + glcmin(:, :, nglcm+1);
        newn = newn + 1;
    end
elseif (pairs == 0)
    glcm = glcmin;
end

size_glcm_1 = size(glcm,1);
size_glcm_2 = size(glcm,2);
size_glcm_3 = size(glcm,3);

out.autoc = zeros(1, size_glcm_3); % AutoCorrelation
out.contr = zeros(1, size_glcm_3); % Contrast
out.corrm = zeros(1, size_glcm_3); % Correlation
out.corrp = zeros(1, size_glcm_3); % Correlation

```

```

out.cprom = zeros(1,size_glcm_3); % Cluster Prominence
out.cshad = zeros(1,size_glcm_3); % Cluster Shade
out.dissi = zeros(1,size_glcm_3); % Dissimilarity
out.energ = zeros(1,size_glcm_3); % Energy
out.entro = zeros(1,size_glcm_3); % Entropy
out.homom = zeros(1,size_glcm_3); % Homogeneity
out.homop = zeros(1,size_glcm_3); % Homogeneity
out.maxpr = zeros(1,size_glcm_3); % Maximum probability

out.sosvh = zeros(1,size_glcm_3); % Sum of Squares
out.savgh = zeros(1,size_glcm_3); % Sum Average
out.svarh = zeros(1,size_glcm_3); % Sum Variance
out.senth = zeros(1,size_glcm_3); % Sum Entropy
out.dvarh = zeros(1,size_glcm_3); % Difference variance
out.denth = zeros(1,size_glcm_3); % Difference entropy
out.inX1h = zeros(1,size_glcm_3); % Information measure of
Correlation1
out.inX2h = zeros(1,size_glcm_3); % Informaiton measure of
Correlation2
out.indnc = zeros(1,size_glcm_3); % Inverse difference normalized
out.idmnc = zeros(1,size_glcm_3); % Inverse difference moment
normalized
glcm_sum = zeros(size_glcm_3,1);
glcm_mean = zeros(size_glcm_3,1);
glcm_var = zeros(size_glcm_3,1);

u_x = zeros(size_glcm_3,1);
u_y = zeros(size_glcm_3,1);
s_x = zeros(size_glcm_3,1);
s_y = zeros(size_glcm_3,1);

p_x = zeros(size_glcm_1,size_glcm_3);
p_y = zeros(size_glcm_2,size_glcm_3);
p_xplusy = zeros((size_glcm_1*2 - 1),size_glcm_3);
p_xminusy = zeros((size_glcm_1),size_glcm_3); %[1]
hxy = zeros(size_glcm_3,1);
hxy1 = zeros(size_glcm_3,1);
hx = zeros(size_glcm_3,1);
hy = zeros(size_glcm_3,1);
hxy2 = zeros(size_glcm_3,1);

for k = 1:size_glcm_3

    glcm_sum(k) = sum(sum(glcm(:, :, k)));
    glcm(:, :, k) = glcm(:, :, k) ./ glcm_sum(k); % normalisasi masing-
masing GLCM
    glcm_mean(k) = mean2(glcm(:, :, k)); % menghitung rata-rata
setelah nomrmalisasi
    glcm_var(k) = (std2(glcm(:, :, k)))^2;

    for i = 1:size_glcm_1

        for j = 1:size_glcm_2

```

```

        out.contr(k) = out.contr(k) + (abs(i -
j))^2.*glcm(i,j,k);
        out.dissi(k) = out.dissi(k) + (abs(i -
j)*glcm(i,j,k));
        out.energ(k) = out.energ(k) + (glcm(i,j,k).^2);
        out.entro(k) = out.entro(k) -
(glcm(i,j,k)*log(glcm(i,j,k) + eps));
        out.homom(k) = out.homom(k) + (glcm(i,j,k)/(1 +
abs(i-j) ));
        out.homop(k) = out.homop(k) + (glcm(i,j,k)/(1 + (i -
j)^2));

        out.sosvh(k) = out.sosvh(k) + glcm(i,j,k)*((i -
glcm_mean(k))^2);

        out.indnc(k) = out.indnc(k) + (glcm(i,j,k)/(1 +
(abs(i-j)/size_glcm_1) ));
        out.idmnc(k) = out.idmnc(k) + (glcm(i,j,k)/(1 + ((i -
j)/size_glcm_1)^2));
        u_x(k) = u_x(k) + (i)*glcm(i,j,k);
        u_y(k) = u_y(k) + (j)*glcm(i,j,k);
    end
end
out.maxpr(k) = max(max(glcm(:, :, k)));
end
% masing-masing glcm telah dinormalisasi
for k = 1:size_glcm_3

    for i = 1:size_glcm_1

        for j = 1:size_glcm_2
            p_x(i,k) = p_x(i,k) + glcm(i,j,k);
            p_y(i,k) = p_y(i,k) + glcm(j,i,k);
            if (ismember((i + j), [2:2*size_glcm_1]))
                p_xplusy((i+j)-1,k) = p_xplusy((i+j)-1,k) +
glcm(i,j,k);
            end
            if (ismember(abs(i-j), [0:(size_glcm_1-1)]))
                p_xminusy((abs(i-j))+1,k) = p_xminusy((abs(i-
j))+1,k) +...
                    glcm(i,j,k);
            end
        end
    end
end
end
% menghitung sum average, Sum Variance and sum entropy
for k = 1:(size_glcm_3)

    for i = 1:(2*(size_glcm_1)-1)
        out.savgh(k) = out.savgh(k) + (i+1)*p_xplusy(i,k);
        out.senth(k) = out.senth(k) -
(p_xplusy(i,k)*log(p_xplusy(i,k) + eps));
    end
end
end

```

```

% menghitung Sum Variance dengan bantuan dari sum entropy
for k = 1:(size_glcm_3)

    for i = 1:(2*(size_glcm_1)-1)
        out.svarh(k) = out.svarh(k) + (((i+1) -
out.senth(k))^2)*p_xplusy(i,k);
    end

end

% menghitung difference variance, difference entropy,
for k = 1:size_glcm_3
    for i = 0:(size_glcm_1-1)
        out.denth(k) = out.denth(k) -
(p_xminusy(i+1,k)*log(p_xminusy(i+1,k) + eps));
        out.dvarh(k) = out.dvarh(k) + (i^2)*p_xminusy(i+1,k);
    end
end

% menghitung information measure of Correlation
for k = 1:size_glcm_3
    hxy(k) = out.entro(k);
    for i = 1:size_glcm_1

        for j = 1:size_glcm_2
            hxy1(k) = hxy1(k) - (glcm(i,j,k)*log(p_x(i,k)*p_y(j,k)
+ eps));
            hxy2(k) = hxy2(k) -
(p_x(i,k)*p_y(j,k)*log(p_x(i,k)*p_y(j,k) + eps));
        end
        hx(k) = hx(k) - (p_x(i,k)*log(p_x(i,k) + eps));
        hy(k) = hy(k) - (p_y(i,k)*log(p_y(i,k) + eps));
    end
    out.inX1h(k) = ( hxy(k) - hxy1(k) ) / ( max([hx(k),hy(k)] ) );
    out.inX2h(k) = ( 1 - exp( -2*( hxy2(k) - hxy(k) ) ) )^0.5;
% Maximal Correlation coefficient tidak dihitung karena
ketidakstabilan perhitungan
end

corm = zeros(size_glcm_3,1);
corp = zeros(size_glcm_3,1);
for k = 1:size_glcm_3
    for i = 1:size_glcm_1
        for j = 1:size_glcm_2
            s_x(k) = s_x(k) + (((i) - u_x(k))^2)*glcm(i,j,k);
            s_y(k) = s_y(k) + (((j) - u_y(k))^2)*glcm(i,j,k);
            corp(k) = corp(k) + ((i)*(j))*glcm(i,j,k);
            corm(k) = corm(k) + (((i) - u_x(k))*((j) -
u_y(k))*glcm(i,j,k));
            out.cprom(k) = out.cprom(k) + (((i + j - u_x(k) -
u_y(k))^4)*...
                glcm(i,j,k));
            out.cshad(k) = out.cshad(k) + (((i + j - u_x(k) -
u_y(k))^3)*...
                glcm(i,j,k));
        end
    end
end

```

```
end
s_x(k) = s_x(k) ^ 0.5;
s_y(k) = s_y(k) ^ 0.5;
out.autoc(k) = corp(k);
out.corrp(k) = (corp(k) - u_x(k)*u_y(k)) / (s_x(k)*s_y(k));
out.corm(k) = corm(k) / (s_x(k)*s_y(k));
end
```

**Lampiran 3.4** Hasil Ekstrasi (GLCM) Data *Training*

No.	Energy (X1)	Contrast (X2)	Correlation (X3)	Sum of Square (X4)	IDM (X5)	Sum Average (X6)	Sum Variance (X7)	Sum Entropy (X8)	Entropy (X9)	Diff. Variance (X10)	Diff Entropy (X11)	Max Prob (X12)	Homogeneity (X13)	Dissimilarity (X14)	TARGET
1	0.5495	0.0196	0.9545	2.2754	0.9961	3.3707	7.9526	0.7043	0.7179	0.0196	0.0966	0.6755	0.9902	0.0196	1
2	0.5522	0.0184	0.9571	2.2844	0.9963	3.375	8.0032	0.6988	0.7114	0.0184	0.0919	0.6783	0.9908	0.0184	1
3	0.5578	0.0181	0.9573	2.2946	0.9964	3.3887	8.1023	0.6921	0.7044	0.0181	0.0906	0.6853	0.9909	0.0181	1
4	0.7324	0.0094	0.9637	2.6811	0.9981	3.6953	10.9083	0.4702	0.4767	0.0094	0.0531	0.843	0.9953	0.0094	1
5	0.5151	0.0204	0.9562	2.1427	0.9959	3.2647	7.2724	0.7423	0.7564	0.0204	0.0994	0.6222	0.9898	0.0204	1
6	0.5968	0.018	0.9533	2.4095	0.9964	3.4785	8.7578	0.6492	0.6617	0.018	0.0902	0.7302	0.991	0.018	1
7	0.5306	0.0223	0.9502	2.2158	0.9955	3.3236	7.6003	0.7299	0.7453	0.0223	0.1068	0.6507	0.9889	0.0223	1
8	0.601	0.0209	0.9447	2.4285	0.9958	3.4929	8.816	0.6504	0.6649	0.0209	0.1016	0.736	0.9895	0.0209	1
9	0.5526	0.0197	0.9541	2.2834	0.9961	3.379	8.0085	0.701	0.7145	0.0197	0.0967	0.6797	0.9902	0.0197	1
10	0.7651	0.0097	0.9569	2.7386	0.9981	3.7412	11.4096	0.4293	0.436	0.0097	0.0547	0.8657	0.9951	0.0097	1
11	0.537	0.0197	0.9556	2.2315	0.9961	3.3355	7.7205	0.7177	0.7314	0.0197	0.0968	0.6579	0.9902	0.0197	1
12	0.5597	0.0178	0.958	2.3017	0.9964	3.3927	8.1367	0.6893	0.7015	0.0178	0.0892	0.6875	0.9911	0.0178	1
13	0.5821	0.0223	0.9437	2.3821	0.9955	3.4559	8.5087	0.674	0.6895	0.0223	0.1069	0.7168	0.9888	0.0223	1
14	0.5454	0.0173	0.9604	2.2591	0.9965	3.3534	7.8794	0.7036	0.7152	0.0173	0.0874	0.668	0.9913	0.0173	1
15	0.5123	0.0227	0.9512	2.1388	0.9955	3.2625	7.2204	0.7501	0.7658	0.0227	0.1084	0.6199	0.9886	0.0227	1
16	0.5158	0.0201	0.9567	2.1476	0.996	3.2665	7.2871	0.741	0.7549	0.0201	0.0985	0.6232	0.9899	0.0201	1
17	0.5718	0.0201	0.9508	2.3471	0.996	3.4277	8.3409	0.6811	0.695	0.0201	0.0984	0.7038	0.99	0.0201	1
18	0.5681	0.0175	0.9578	2.3297	0.9965	3.4129	8.2828	0.6796	0.6917	0.0175	0.0881	0.6977	0.9913	0.0175	1
19	0.4842	0.0229	0.9535	1.9584	0.9954	3.1148	6.4166	0.7797	0.7955	0.0229	0.1092	0.546	0.9885	0.0229	1
20	0.5364	0.0189	0.9576	2.2253	0.9962	3.3314	7.7083	0.7166	0.7297	0.0189	0.0937	0.6563	0.9906	0.0189	1
21	0.52	0.0205	0.9554	2.165	0.9959	3.2833	7.381	0.7374	0.7516	0.0205	0.1	0.6314	0.9897	0.0205	1
22	0.8368	0.0068	0.9567	2.8504	0.9986	3.8289	12.5905	0.324	0.3286	0.0068	0.0406	0.9111	0.9966	0.0068	1
23	0.5127	0.0254	0.9452	2.1536	0.9949	3.2737	7.2445	0.7548	0.7724	0.0254	0.1182	0.6242	0.9873	0.0254	1
24	0.6886	0.014	0.9528	2.6078	0.9972	3.6363	10.2018	0.5346	0.5443	0.014	0.0739	0.8111	0.993	0.014	1

25	0.5286	0.0177	0.961	2.1892	0.9965	3.3032	7.5512	0.7223	0.7345	0.0177	0.089	0.6427	0.9911	0.0177	1
26	0.6022	0.0194	0.9488	2.4276	0.9961	3.4924	8.8407	0.6459	0.6594	0.0194	0.0957	0.7365	0.9903	0.0194	2
27	0.5374	0.0165	0.963	2.2225	0.9967	3.3276	7.7265	0.7103	0.7217	0.0165	0.0842	0.6555	0.9917	0.0165	2
28	0.7918	0.0049	0.9759	2.7754	0.999	3.7703	11.8837	0.3818	0.3852	0.0049	0.031	0.8827	0.9975	0.0049	2
29	0.4853	0.0218	0.9558	1.9594	0.9956	3.1154	6.4368	0.7762	0.7912	0.0218	0.105	0.5468	0.9891	0.0218	2
30	0.8922	0.0035	0.9662	2.9256	0.9993	3.8896	13.5857	0.2317	0.234	0.0035	0.0234	0.9431	0.9982	0.0035	2
31	0.53	0.0206	0.9542	2.2077	0.9959	3.3167	7.5848	0.7271	0.7414	0.0206	0.1004	0.648	0.9897	0.0206	2
32	0.7884	0.0082	0.9596	2.7752	0.9984	3.7701	11.7909	0.3949	0.4006	0.0082	0.0476	0.881	0.9959	0.0082	2
33	0.5496	0.0195	0.9549	2.2757	0.9961	3.3706	7.9548	0.7038	0.7173	0.0195	0.096	0.6756	0.9903	0.0195	2
34	0.7495	0.0059	0.976	2.7053	0.9988	3.7146	11.2084	0.4397	0.4437	0.0059	0.036	0.8544	0.9971	0.0059	2
35	0.675	0.0161	0.948	2.5854	0.9968	3.6178	9.9827	0.5551	0.5662	0.0161	0.0824	0.8009	0.992	0.0161	2
36	0.6454	0.0159	0.953	2.5218	0.9968	3.5675	9.5297	0.5897	0.6007	0.0159	0.0818	0.7758	0.992	0.0159	2
37	0.7934	0.0054	0.9734	2.779	0.9989	3.773	11.9031	0.381	0.3847	0.0054	0.0334	0.8838	0.9973	0.0054	2
38	0.5596	0.0194	0.9539	2.3076	0.9961	3.3967	8.1334	0.6929	0.7064	0.0194	0.0958	0.6886	0.9903	0.0194	2
39	0.8594	0.0065	0.9518	2.8817	0.9987	3.8553	12.9744	0.2899	0.2944	0.0065	0.039	0.9244	0.9968	0.0065	2
40	0.6875	0.0123	0.9592	2.6018	0.9975	3.6318	10.1973	0.532	0.5405	0.0123	0.0661	0.8098	0.9939	0.0123	2
41	0.7868	0.0053	0.9747	2.7678	0.9989	3.7642	11.7992	0.3895	0.3932	0.0053	0.0328	0.8795	0.9974	0.0053	2
42	0.5231	0.0198	0.9568	2.1769	0.996	3.2917	7.4441	0.7326	0.7463	0.0198	0.0972	0.6359	0.9901	0.0198	2
43	0.8003	0.0046	0.9764	2.7888	0.9991	3.7809	12.0232	0.3695	0.3727	0.0046	0.0293	0.8882	0.9977	0.0046	2
44	0.6244	0.018	0.9497	2.4791	0.9964	3.533	9.1948	0.6181	0.6305	0.018	0.0901	0.7575	0.991	0.018	2
45	0.812	0.0055	0.9699	2.8086	0.9989	3.7968	12.1986	0.356	0.3599	0.0055	0.0341	0.8957	0.9973	0.0055	2
46	0.5112	0.0215	0.954	2.1285	0.9957	3.2542	7.1908	0.7488	0.7637	0.0215	0.104	0.6163	0.9892	0.0215	2
47	0.6559	0.0152	0.9537	2.5429	0.997	3.5847	9.6957	0.5759	0.5865	0.0152	0.0789	0.7847	0.9924	0.0152	2
48	0.5631	0.0236	0.943	2.3313	0.9953	3.415	8.1896	0.6974	0.7138	0.0236	0.1117	0.6957	0.9882	0.0236	2
49	0.5119	0.0173	0.9633	2.1125	0.9965	3.2404	7.1827	0.7388	0.7508	0.0173	0.0874	0.6115	0.9913	0.0173	2
50	0.79	0.0057	0.9723	2.7734	0.9989	3.7689	11.8448	0.3864	0.3903	0.0057	0.0349	0.8816	0.9972	0.0057	2
51	0.7816	0.0081	0.9613	2.7634	0.9984	3.7611	11.6841	0.4038	0.4095	0.0081	0.0472	0.8765	0.9959	0.0081	2
52	0.7839	0.0059	0.9719	2.7637	0.9988	3.7612	11.7446	0.3952	0.3993	0.0059	0.0362	0.8777	0.997	0.0059	3



53	0.6087	0.017	0.9547	2.4431	0.9966	3.5008	8.9521	0.6337	0.6451	0.017	0.0861	0.7419	0.9915	0.017	3
54	0.6805	0.0129	0.9579	2.5895	0.9974	3.6217	10.0858	0.5419	0.5508	0.0129	0.0691	0.8044	0.9935	0.0129	3
55	0.4781	0.024	0.9519	1.8833	0.9952	3.0555	6.1142	0.7881	0.8047	0.024	0.1131	0.5158	0.988	0.024	3
56	0.4817	0.0232	0.9533	1.9288	0.9954	3.0935	6.3077	0.7827	0.7988	0.0232	0.1101	0.5352	0.9884	0.0232	3
57	0.6786	0.0102	0.9674	2.5802	0.998	3.6142	10.0781	0.5376	0.5447	0.0102	0.0567	0.802	0.9949	0.0102	3
58	0.7033	0.0064	0.9779	2.621	0.9987	3.6475	10.493	0.4982	0.5025	0.0064	0.0387	0.8206	0.9968	0.0064	3
59	0.7419	0.0078	0.9689	2.6955	0.9984	3.7066	11.0693	0.4544	0.4599	0.0078	0.0456	0.8494	0.9961	0.0078	3
60	0.8938	0.0032	0.9686	2.9255	0.9994	3.8911	13.6179	0.2284	0.2306	0.0032	0.0218	0.9439	0.9984	0.0032	3
61	0.5129	0.0224	0.9518	2.1414	0.9955	3.2638	7.2326	0.7489	0.7644	0.0224	0.1073	0.6207	0.9888	0.0224	3
62	0.5102	0.0233	0.9502	2.1334	0.9953	3.2566	7.1768	0.7534	0.7695	0.0233	0.1104	0.6167	0.9884	0.0233	3
63	0.6863	0.0146	0.9512	2.6042	0.9971	3.6336	10.1638	0.5385	0.5486	0.0146	0.0763	0.8095	0.9927	0.0146	3
64	0.5397	0.0197	0.9552	2.2431	0.9961	3.3436	7.7719	0.715	0.7287	0.0197	0.097	0.6619	0.9901	0.0197	3
65	0.5413	0.0186	0.9578	2.2431	0.9963	3.345	7.8013	0.7108	0.7237	0.0186	0.0925	0.6632	0.9907	0.0186	3
66	0.5103	0.0235	0.9497	2.135	0.9953	3.2577	7.18	0.7537	0.77	0.0235	0.1112	0.6171	0.9883	0.0235	3
67	0.5416	0.0213	0.9513	2.2536	0.9957	3.3535	7.8092	0.7162	0.731	0.0213	0.1031	0.6661	0.9893	0.0213	3
68	0.6078	0.0197	0.9473	2.4423	0.9961	3.5042	8.9285	0.6402	0.6538	0.0197	0.0967	0.7422	0.9902	0.0197	3
69	0.692	0.0131	0.9555	2.6126	0.9974	3.6402	10.2592	0.5286	0.5377	0.0131	0.0699	0.8135	0.9934	0.0131	3
70	0.8084	0.0047	0.9749	2.8015	0.9991	3.7913	12.1517	0.3587	0.362	0.0047	0.0298	0.8933	0.9977	0.0047	3
71	0.4882	0.0271	0.9441	2.0264	0.9946	3.1711	6.6425	0.7839	0.8027	0.0271	0.1247	0.572	0.9864	0.0271	3
72	0.7516	0.0103	0.9567	2.7169	0.9979	3.7236	11.1949	0.4481	0.4552	0.0103	0.0574	0.8566	0.9948	0.0103	3
73	0.515	0.0209	0.9549	2.146	0.9958	3.2666	7.2737	0.7436	0.7581	0.0209	0.1017	0.6228	0.9895	0.0209	3
74	0.619	0.0169	0.9535	2.4625	0.9966	3.5209	9.1157	0.6219	0.6336	0.0169	0.0858	0.752	0.9915	0.0169	3
75	0.5417	0.0197	0.955	2.2504	0.9961	3.3495	7.8102	0.7128	0.7265	0.0197	0.097	0.6649	0.9901	0.0197	3

**Lampiran 3.5** Hasil Ekstrasi (GLCM) Data *Testing*

No.	Energy (X1)	Contrast (X2)	Correlation (X3)	Sum of Square (X4)	IDM (X5)	Sum Average (X6)	Sum Variance (X7)	Sum Entropy (X8)	Entropy (X9)	Diff. Variance (X10)	Diff Entropy (X11)	Max Prob (X12)	Homogeneity (X13)	Dissimilarity (X14)	TARGET
1	0.4794	0.022	0.9559	1.8667	0.9956	3.042	6.0806	0.7827	0.7979	0.022	0.1057	0.51	0.989	0.022	1
2	0.5373	0.0205	0.9537	2.235	0.9959	3.3389	7.728	0.7191	0.7333	0.0205	0.1	0.6592	0.9897	0.0205	1
3	0.6585	0.0139	0.9576	2.5469	0.9972	3.5868	9.7429	0.57	0.5796	0.0139	0.0732	0.7865	0.9931	0.0139	1
4	0.8616	0.0044	0.9672	2.882	0.9991	3.8556	13.0371	0.2816	0.2846	0.0044	0.0282	0.9256	0.9978	0.0044	1
5	0.494	0.0193	0.9603	2.0132	0.9961	3.1613	6.7117	0.7619	0.7753	0.0193	0.0955	0.571	0.9903	0.0193	1
6	0.4923	0.0229	0.9529	2.0257	0.9954	3.1709	6.7063	0.7713	0.7871	0.0229	0.109	0.574	0.9886	0.0229	1
7	0.5401	0.0184	0.9584	2.2404	0.9963	3.3412	7.7799	0.7116	0.7243	0.0184	0.0917	0.6614	0.9908	0.0184	1
8	0.5751	0.019	0.9532	2.3524	0.9962	3.4331	8.3999	0.6751	0.6882	0.019	0.0941	0.707	0.9905	0.019	1
9	0.6706	0.0155	0.9506	2.5728	0.9969	3.6098	9.9188	0.5592	0.5698	0.0155	0.08	0.7971	0.9922	0.0155	1
10	0.5185	0.0209	0.9548	2.1602	0.9958	3.279	7.3493	0.7397	0.7542	0.0209	0.1014	0.6291	0.9896	0.0209	1
11	0.8712	0.0047	0.9624	2.8955	0.9991	3.867	13.1984	0.2677	0.2709	0.0047	0.0297	0.9312	0.9977	0.0047	2
12	0.52	0.0183	0.9604	2.1595	0.9963	3.2755	7.3717	0.7327	0.7452	0.0183	0.0914	0.6286	0.9908	0.0183	2
13	0.5396	0.0207	0.9529	2.245	0.9959	3.346	7.7704	0.7172	0.7316	0.0207	0.1009	0.6627	0.9896	0.0207	2
14	0.8311	0.005	0.9694	2.8374	0.999	3.8199	12.5169	0.3279	0.3314	0.005	0.0315	0.9074	0.9975	0.005	2
15	0.8235	0.0055	0.9681	2.8269	0.9989	3.8111	12.3862	0.3399	0.3436	0.0055	0.0339	0.9028	0.9973	0.0055	2
16	0.6389	0.0176	0.9487	2.5105	0.9965	3.559	9.4214	0.6007	0.6129	0.0176	0.0887	0.7707	0.9912	0.0176	2
17	0.5202	0.0198	0.9569	2.1644	0.996	3.2816	7.3821	0.7358	0.7495	0.0198	0.0974	0.6309	0.9901	0.0198	2
18	0.597	0.0174	0.9549	2.4096	0.9965	3.4777	8.7632	0.6477	0.6598	0.0174	0.0878	0.7301	0.9913	0.0174	3
19	0.8329	0.0054	0.9667	2.84	0.9989	3.8225	12.5405	0.3264	0.3301	0.0054	0.0335	0.9085	0.9973	0.0054	3
20	0.6047	0.0217	0.9421	2.4393	0.9957	3.5019	8.8737	0.6475	0.6626	0.0217	0.1045	0.7401	0.9892	0.0217	3

21	0.5183	0.0202	0.9562	2.1577	0.996	3.2763	7.3434	0.7385	0.7526	0.0202	0.0989	0.628	0.9899	0.0202	3
22	0.8418	0.0041	0.9733	2.8526	0.9992	3.8317	12.7048	0.3102	0.3131	0.0041	0.0267	0.9138	0.9979	0.0041	3
23	0.6723	0.0137	0.9565	2.5743	0.9973	3.6095	9.9554	0.5533	0.5628	0.0137	0.0723	0.7979	0.9932	0.0137	3
24	0.4941	0.0239	0.9504	2.0451	0.9952	3.1868	6.7745	0.7715	0.7881	0.0239	0.113	0.5814	0.988	0.0239	3
25	0.4997	0.0222	0.9536	2.0702	0.9956	3.2069	6.9111	0.7623	0.7777	0.0222	0.1066	0.5923	0.9889	0.0222	3

**Lampiran 3.6** Hasil Normalisasi Data *Training* dan Data *Testing*

No	Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Max Prob (X12)	Homogeneity (X13)	TARGET
Data Training									
1	-0.64345	0.581704	-0.35664	-0.56499	-0.59555	0.644347	-0.52364	-0.57909	-1.21268
2	-0.62044	0.396644	-0.05068	-0.4109	-0.56987	0.60326	-0.49862	-0.39427	-1.21268
3	-0.57273	0.350379	-0.02714	-0.33386	-0.51958	0.559012	-0.43607	-0.36347	-1.21268
4	0.915016	-0.9913	0.725991	0.975895	0.904478	-0.8803	0.973077	0.991839	-1.21268
5	-0.93657	0.705077	-0.15659	-0.71908	-0.94075	0.887709	-0.99991	-0.7023	-1.21268
6	-0.24041	0.334958	-0.49785	-0.33386	-0.18691	0.289101	-0.03486	-0.33267	-1.21268
7	-0.80449	0.998088	-0.86265	-1.02726	-0.77434	0.817544	-0.74524	-0.97952	-1.21268
8	-0.20463	0.782185	-1.50988	-0.79612	-0.15737	0.309329	0.016966	-0.7947	-1.21268
9	-0.61704	0.597125	-0.40371	-0.56499	-0.56718	0.622855	-0.48611	-0.57909	-1.21268
10	1.193648	-0.94504	-0.07422	0.975895	1.158889	-1.13757	1.175916	0.930234	-1.21268
11	-0.74996	0.597125	-0.2272	-0.56499	-0.71334	0.729681	-0.68091	-0.57909	-1.21268
12	-0.55654	0.304114	0.05523	-0.33386	-0.50212	0.540681	-0.41641	-0.30186	-1.21268
13	-0.36567	0.998088	-1.62756	-1.02726	-0.31333	0.464828	-0.1546	-1.01032	-1.21268
14	-0.67839	0.227006	0.337656	-0.25681	-0.6327	0.62728	-0.59066	-0.24026	-1.21268
15	-0.96043	1.059774	-0.74498	-1.02726	-0.96714	0.947127	-1.02046	-1.07193	-1.21268
16	-0.9306	0.658812	-0.09775	-0.64204	-0.93329	0.878227	-0.99097	-0.67149	-1.21268
17	-0.45343	0.658812	-0.79205	-0.64204	-0.39849	0.499594	-0.27076	-0.64069	-1.21268
18	-0.48496	0.25785	0.031694	-0.25681	-0.42797	0.478734	-0.32527	-0.24026	-1.21268
19	-1.19986	1.090617	-0.47432	-1.1043	-1.37507	1.134863	-1.6808	-1.10273	-1.21268
20	-0.75507	0.473752	0.008159	-0.48795	-0.71953	0.718935	-0.6952	-0.45588	-1.21268
21	-0.89482	0.720498	-0.25073	-0.71908	-0.88564	0.857367	-0.9177	-0.7331	-1.21268
22	1.804594	-1.39226	-0.09775	1.361117	1.7582	-1.81645	1.581593	1.392271	-1.21268
23	-0.95702	1.476158	-1.45104	-1.48952	-0.95491	0.988846	-0.98204	-1.47236	-1.21268
24	0.541802	-0.28191	-0.55669	0.282496	0.545927	-0.45299	0.688031	0.283383	-1.21268
25	-0.82154	0.288693	0.408262	-0.25681	-0.79926	0.749277	-0.81673	-0.30186	-1.21268
26	-0.1944	0.55086	-1.0274	-0.56499	-0.14484	0.274563	0.021434	-0.54828	0.016388
27	-0.74655	0.103633	0.643617	-0.10273	-0.7103	0.668367	-0.70235	-0.11705	0.016388
28	1.421155	-1.68528	2.161656	1.669294	1.399497	-1.45868	1.327821	1.669493	0.016388

29	-1.19049	0.92098	-0.20366	-0.95021	-1.36482	1.107682	-1.67365	-0.91791	0.016388
30	2.276649	-1.90118	1.020185	1.900427	2.263267	-2.41443	1.867533	1.88511	0.016388
31	-0.80961	0.73592	-0.39194	-0.71908	-0.78221	0.792892	-0.76937	-0.7331	0.016388
32	1.392184	-1.17636	0.243514	1.207028	1.3524	-1.36134	1.312631	1.176654	0.016388
33	-0.6426	0.566282	-0.30957	-0.56499	-0.59443	0.640554	-0.52275	-0.54828	0.016388
34	1.060722	-1.53106	2.173424	1.515205	1.05678	-1.0889	1.074943	1.546283	0.016388
35	0.425918	0.041947	-1.12154	-0.02568	0.434733	-0.31456	0.596888	-0.02464	0.016388
36	0.173701	0.011104	-0.53316	-0.02568	0.204834	-0.09649	0.372603	-0.02464	0.016388
37	1.434788	-1.60817	1.867463	1.592249	1.409342	-1.46184	1.337651	1.607888	0.016388
38	-0.55739	0.55086	-0.42725	-0.56499	-0.50379	0.571654	-0.40658	-0.54828	0.016388
39	1.997165	-1.43853	-0.67437	1.438161	1.95303	-2.03263	1.700437	1.453876	0.016388
40	0.532429	-0.54407	0.196443	0.513629	0.543643	-0.47701	0.676415	0.560605	0.016388
41	1.37855	-1.62359	2.020443	1.592249	1.356613	-1.40811	1.299227	1.63869	0.016388
42	-0.8684	0.612547	-0.08598	-0.64204	-0.85361	0.823866	-0.87749	-0.60989	0.016388
43	1.493582	-1.73154	2.220495	1.746338	1.470293	-1.53769	1.376967	1.731098	0.016388
44	-0.00524	0.334958	-0.92149	-0.33386	0.034872	0.091883	0.209082	-0.33267	0.016388
45	1.593276	-1.59275	1.455592	1.592249	1.559309	-1.6186	1.443985	1.607888	0.016388
46	-0.9698	0.874715	-0.41548	-0.87317	-0.98216	0.933852	-1.05263	-0.88711	0.016388
47	0.26317	-0.09685	-0.45078	0.128407	0.28908	-0.18624	0.452131	0.098568	0.016388
48	-0.52757	1.198569	-1.70993	-1.18135	-0.47527	0.61843	-0.34314	-1.19514	0.016388
49	-0.96383	0.227006	0.67892	-0.25681	-0.98628	0.85231	-1.09552	-0.24026	0.016388
50	1.405817	-1.5619	1.738017	1.592249	1.379755	-1.42644	1.317992	1.577085	0.016388
51	1.334242	-1.19178	0.443566	1.207028	1.298199	-1.30508	1.272421	1.176654	0.016388
52	1.35384	-1.53106	1.690946	1.515205	1.328903	-1.36955	1.283143	1.515481	1.245458
53	-0.13902	0.180741	-0.33311	-0.17977	-0.0883	0.184171	0.069686	-0.17865	1.245458
54	0.472783	-0.45155	0.043462	0.436585	0.487057	-0.41191	0.628162	0.437395	1.245458
55	-1.25184	1.260255	-0.6626	-1.25839	-1.52854	1.193017	-1.95066	-1.25674	1.245458
56	-1.22116	1.136882	-0.49785	-1.1043	-1.43034	1.155723	-1.77731	-1.13353	1.245458
57	0.456593	-0.86793	1.161398	0.898851	0.483149	-0.45047	0.606717	0.868629	1.245458
58	0.667059	-1.45395	2.397011	1.438161	0.693712	-0.71722	0.772919	1.453876	1.245458
59	0.995964	-1.23805	1.337914	1.207028	0.986186	-0.98649	1.030265	1.238258	1.245458
60	2.290283	-1.94744	1.302611	1.977471	2.279608	-2.43592	1.874681	1.946715	1.245458
61	-0.95531	1.013509	-0.67437	-1.02726	-0.96095	0.938277	-1.01331	-1.01032	1.245458
62	-0.97832	1.152304	-0.86265	-1.18135	-0.98927	0.970515	-1.04905	-1.13353	1.245458

63	0.522204	-0.18938	-0.74498	0.205452	0.526642	-0.42581	0.673734	0.190975	1.245458
64	-0.72695	0.597125	-0.27427	-0.56499	-0.68725	0.712614	-0.64516	-0.60989	1.245458
65	-0.71332	0.427487	0.031694	-0.4109	-0.67233	0.681009	-0.63355	-0.42507	1.245458
66	-0.97747	1.183147	-0.92149	-1.18135	-0.98765	0.973675	-1.04548	-1.16433	1.245458
67	-0.71076	0.843871	-0.73321	-0.87317	-0.66832	0.727153	-0.60763	-0.85631	1.245458
68	-0.14668	0.597125	-1.20392	-0.56499	-0.10028	0.239165	0.072367	-0.57909	1.245458
69	0.570773	-0.4207	-0.23896	0.436585	0.575058	-0.49471	0.709476	0.406592	1.245458
70	1.562601	-1.71612	2.043979	1.746338	1.535507	-1.60533	1.422539	1.731098	1.245458
71	-1.16578	1.738326	-1.58049	-1.72066	-1.26043	1.180375	-1.44848	-1.74958	1.245458
72	1.078616	-0.85251	-0.09775	0.821806	1.049928	-1.0162	1.094602	0.837827	1.245458
73	-0.93742	0.782185	-0.30957	-0.79612	-0.94009	0.898454	-0.99455	-0.7947	1.245458
74	-0.05125	0.16532	-0.47432	-0.17977	-0.00527	0.111479	0.159936	-0.17865	1.245458
75	-0.70991	0.597125	-0.2978	-0.56499	-0.66782	0.698708	-0.61836	-0.60989	1.245458
<i>Data Testing</i>									
1	-1.06115	0.900755	-0.21982	-0.89944	-1.29305	0.995438	-1.64401	-0.89916	-1.06708
2	-0.64006	0.679982	-0.52513	-0.67899	-0.58632	0.647564	-0.50786	-0.69347	-1.06708
3	0.241397	-0.29142	0.016098	0.276299	0.278063	-0.18012	0.461527	0.305596	-1.06708
4	1.718495	-1.68965	1.348365	1.67249	1.691264	-1.76871	1.520767	1.686655	-1.06708
5	-0.95497	0.503363	0.390798	-0.53202	-1.02231	0.873736	-1.17949	-0.51716	-1.06708
6	-0.96734	1.033219	-0.63616	-1.04641	-1.02463	0.937279	-1.15665	-1.01669	-1.06708
7	-0.6197	0.370899	0.12712	-0.38505	-0.56406	0.599098	-0.4911	-0.37024	-1.06708
8	-0.36515	0.459209	-0.59452	-0.45854	-0.29808	0.404698	-0.14386	-0.45839	-1.06708
9	0.329398	-0.05593	-0.95535	0.055848	0.353524	-0.23289	0.542245	0.041138	-1.06708
10	-0.77679	0.738855	-0.37248	-0.75247	-0.74878	0.760111	-0.73707	-0.72285	-1.06708
11	1.788313	-1.6455	0.682231	1.67249	1.760461	-1.84249	1.56341	1.657271	0.092789
12	-0.76588	0.356181	0.404676	-0.38505	-0.73917	0.711646	-0.74087	-0.37024	0.092789
13	-0.62333	0.709418	-0.63616	-0.67899	-0.56813	0.638409	-0.4812	-0.72285	0.092789
14	1.496675	-1.60134	1.653676	1.599006	1.4681	-1.51669	1.382174	1.598503	0.092789
15	1.441402	-1.52775	1.473265	1.525522	1.41203	-1.45099	1.347146	1.539734	0.092789
16	0.098851	0.253153	-1.21902	-0.23809	0.140141	-0.0008	0.341211	-0.2527	0.092789
17	-0.76443	0.576954	-0.08105	-0.60551	-0.73471	0.734801	-0.72336	-0.57593	0.092789
18	-0.20588	0.223717	-0.3586	-0.23809	-0.14222	0.251762	0.032044	-0.22332	1.252656
19	1.509766	-1.54247	1.278976	1.525522	1.478224	-1.52369	1.390551	1.539734	1.252656
20	-0.14988	0.856601	-2.13496	-0.82596	-0.09482	0.26684	0.108193	-0.84039	1.252656

21	-0.77824	0.635827	-0.17819	-0.60551	-0.75131	0.751495	-0.74544	-0.6347	1.252656
22	1.574494	-1.73381	2.194909	1.745974	1.548708	-1.61524	1.43091	1.71604	1.252656
23	0.341761	-0.32086	-0.13656	0.349783	0.369225	-0.27059	0.548337	0.33498	1.252656
24	-0.95424	1.180401	-0.9831	-1.19338	-0.99537	0.942665	-1.1003	-1.193	1.252656
25	-0.91352	0.930192	-0.53901	-0.89944	-0.93677	0.88666	-1.0173	-0.92854	1.252656

**Lampiran 3.7** *Target, Output RNN dan Error*

	No	Target	Output	Error	Status Output	Ket
Data Training	1	1	1.764176	-0.76418	2	Tidak Cocok
	2	1	1.373493	-0.37349	1	Cocok
	3	1	1.347956	-0.34796	1	Cocok
	4	1	1.255733	-0.25573	1	Cocok
	5	1	1.630424	-0.63042	2	Tidak Cocok
	6	1	1.424067	-0.42407	1	Cocok
	7	1	1.879436	-0.87944	2	Tidak Cocok
	8	1	1.397762	-0.39776	1	Cocok
	9	1	1.786667	-0.78667	2	Tidak Cocok
	10	1	1.61933	-0.61933	2	Tidak Cocok
	11	1	1.633657	-0.63366	2	Tidak Cocok
	12	1	1.319779	-0.31978	1	Cocok
	13	1	1.173485	-0.17349	1	Cocok
	14	1	1.307107	-0.30711	1	Cocok
	15	1	2.567016	-1.56702	3	Tidak Cocok
	16	1	1.516209	-0.51621	2	Tidak Cocok
	17	1	1.186779	-0.18678	1	Cocok
	18	1	1.301543	-0.30154	1	Cocok
	19	1	2.244008	-1.24401	2	Tidak Cocok
	20	1	1.378377	-0.37838	1	Cocok
	21	1	1.831139	-0.83114	2	Tidak Cocok
	22	1	1.179455	-0.17946	1	Cocok
	23	1	1.264216	-0.26422	1	Cocok
	24	1	1.923402	-0.9234	2	Tidak Cocok
	25	1	1.339023	-0.33902	1	Cocok
	26	2	1.639907	0.360093	2	Cocok
	27	2	1.739304	0.260696	2	Cocok
	28	2	1.863881	0.136119	2	Cocok
	29	2	1.477432	0.522568	1	Tidak Cocok
	30	2	2.01418	-0.01418	2	Cocok
	31	2	2.07194	-0.07194	2	Cocok
	32	2	1.787518	0.212482	2	Cocok
	33	2	1.702829	0.297171	2	Cocok
	34	2	2.008601	-0.0086	2	Cocok
	35	2	2.281066	-0.28107	2	Cocok
	36	2	1.951124	0.048876	2	Cocok
	37	2	2.494253	-0.49425	2	Cocok
	38	2	1.686761	0.313239	2	Cocok
	39	2	1.929855	0.070145	2	Cocok



	40	2	2.089271	-0.08927	2	Cocok
	41	2	2.006021	-0.00602	2	Cocok
	42	2	1.514118	0.485882	2	Cocok
	43	2	2.007007	-0.00701	2	Cocok
	44	2	2.306495	-0.3065	2	Cocok
	45	2	2.794456	-0.79446	3	Tidak Cocok
	46	2	2.237979	-0.23798	2	Cocok
	47	2	1.958123	0.041877	2	Cocok
	48	2	1.177433	0.822567	1	Tidak Cocok
	49	2	2.163999	-0.164	2	Cocok
	50	2	2.640209	-0.64021	3	Tidak Cocok
	51	2	2.059999	-0.06	2	Cocok
	52	3	2.598979	0.401021	3	Cocok
	53	3	2.799473	0.200527	3	Cocok
	54	3	2.751046	0.248954	3	Cocok
	55	3	2.695662	0.304338	3	Cocok
	56	3	2.287344	0.712656	2	Tidak Cocok
	57	3	2.799608	0.200392	3	Cocok
	58	3	2.795999	0.204001	3	Cocok
	59	3	2.636022	0.363978	3	Cocok
	60	3	2.779134	0.220866	3	Cocok
	61	3	2.53623	0.46377	3	Cocok
	62	3	2.591535	0.408465	3	Cocok
	63	3	2.018173	0.981827	2	Tidak Cocok
	64	3	1.72349	1.27651	2	Tidak Cocok
	65	3	1.355663	1.644337	1	Tidak Cocok
	66	3	2.56173	0.43827	3	Cocok
	67	3	1.911739	1.088261	2	Tidak Cocok
	68	3	2.610676	0.389324	3	Cocok
	69	3	2.572816	0.427184	3	Cocok
	70	3	2.575643	0.424357	3	Cocok
	71	3	2.675452	0.324548	3	Cocok
	72	3	2.444324	0.555676	2	Tidak Cocok
	73	3	1.974414	1.025586	2	Tidak Cocok
	74	3	2.796221	0.203779	3	Cocok
	75	3	1.753981	1.246019	2	Tidak Cocok
Data Testing	1	1	1.436922	-0.43692	1	Cocok
	2	1	2.14146	-1.14146	2	Tidak Cocok
	3	1	2.795365	-1.79537	3	Tidak Cocok
	4	1	2.743995	-1.74399	3	Tidak Cocok
	5	1	1.298676	-0.29868	1	Cocok
	6	1	2.479233	-1.47923	2	Tidak Cocok

7	1	1.332294	-0.33229	1	Cocok
8	1	1.339793	-0.33979	1	Cocok
9	1	1.57461	-0.57461	2	Tidak Cocok
10	1	2.003934	-1.00393	2	Tidak Cocok
11	2	1.430666	0.569334	1	Tidak Cocok
12	2	1.316384	0.683616	1	Tidak Cocok
13	2	2.180216	-0.18022	2	Cocok
14	2	2.797195	-0.79719	3	Tidak Cocok
15	2	2.794938	-0.79494	3	Tidak Cocok
16	2	1.70496	0.29504	2	Cocok
17	2	1.476079	0.523921	1	Tidak Cocok
18	3	1.80941	1.19059	2	Tidak Cocok
19	3	2.76288	0.23712	3	Cocok
20	3	2.430366	0.569634	2	Tidak Cocok
21	3	1.581869	1.418131	2	Tidak Cocok
22	3	2.794475	0.205525	3	Cocok
23	3	2.747536	0.252464	3	Cocok
24	3	2.725619	0.274381	3	Cocok
25	3	2.299485	0.700515	2	Tidak Cocok

Lampiran 3.8 Nilai  $m_{it}^s$  dan Pembagian Cluster Data

Data Ke-	Matriks Partisi (U)			Maksimal $m_{it}^s$	Ketentuan $m_{it}^s$			Cluster
	$m_{it}^1$	$m_{it}^2$	$m_{it}^3$		$m_{it}^1$	$m_{it}^2$	$m_{it}^3$	
1	0.011027	0.986837	0.002136	0.987112	0	1	0	2
2	0.01073	0.987112	0.002158	0.982007	0	1	0	2
3	0.014779	0.982007	0.003214	0.72719	0	1	0	2
4	0.104934	0.167875	0.72719	0.828636	0	0	1	3
5	0.150065	0.828636	0.021299	0.822948	0	1	0	2
6	0.124781	0.822948	0.05227	0.932315	0	1	0	2
7	0.058471	0.932315	0.009214	0.803955	0	1	0	2
8	0.135424	0.803955	0.060621	0.987025	0	1	0	2
9	0.010796	0.987025	0.002179	0.82806	0	1	0	2
10	0.069623	0.102318	0.82806	0.960451	0	0	1	3
11	0.033882	0.960451	0.005667	0.978573	0	1	0	2
12	0.017508	0.978573	0.003919	0.899978	0	1	0	2
13	0.075847	0.899978	0.024174	0.982914	0	1	0	2
14	0.014428	0.982914	0.002657	0.810874	0	1	0	2
15	0.165813	0.810874	0.023314	0.833646	0	1	0	2
16	0.145624	0.833646	0.02073	0.943306	0	1	0	2
17	0.044666	0.943306	0.012028	0.955622	0	1	0	2
18	0.035363	0.955622	0.009015	0.599853	0	1	0	2
19	0.349063	0.599853	0.051084	0.958019	0	1	0	2
20	0.035999	0.958019	0.005983	0.865329	0	1	0	2
21	0.117553	0.865329	0.017118	0.84561	0	1	0	2
22	0.067075	0.087315	0.84561	0.818868	0	0	1	3
23	0.158725	0.818868	0.022407	0.476793	0	1	0	2
24	0.178389	0.344817	0.476793	0.918888	0	0	1	3
25	0.070266	0.918888	0.010847	0.534854	0	1	0	2
26	0.373264	0.534854	0.091881	0.583428	0	1	0	2
27	0.583428	0.398125	0.018447	0.991723	1	0	0	1
28	0.003794	0.004484	0.991723	0.597553	0	0	1	3
29	0.597553	0.362385	0.040063	0.82372	1	0	0	1
30	0.082761	0.093519	0.82372	0.618472	0	0	1	3
31	0.618472	0.363816	0.017712	0.994725	1	0	0	1
32	0.002413	0.002862	0.994725	0.512118	0	0	1	3
33	0.512118	0.465757	0.022126	0.958829	1	0	0	1
34	0.018585	0.022586	0.958829	0.473875	0	0	1	3
35	0.227385	0.298741	0.473875	0.419638	0	0	1	3
36	0.306865	0.419638	0.273497	0.990896	0	1	0	2
37	0.004174	0.00493	0.990896	0.511102	0	0	1	3
38	0.461242	0.511102	0.027656	0.889787	0	1	0	2

	39	0.051382	0.058831	0.889787	0.5823	0	0	1	3
	40	0.182265	0.235435	0.5823	0.994521	0	0	1	3
	41	0.002507	0.002972	0.994521	0.640013	0	0	1	3
	42	0.640013	0.341928	0.018059	0.984306	1	0	0	1
	43	0.007212	0.008482	0.984306	0.487581	0	0	1	3
	44	0.346086	0.487581	0.166333	0.971159	0	1	0	2
	45	0.013294	0.015548	0.971159	0.649061	0	0	1	3
	46	0.649061	0.329929	0.02101	0.378517	1	0	0	1
	47	0.281038	0.378517	0.340445	0.521837	0	1	0	2
	48	0.448168	0.521837	0.029995	0.64887	0	1	0	2
	49	0.64887	0.329986	0.021145	0.993196	1	0	0	1
	50	0.003116	0.003688	0.993196	0.99583	0	0	1	3
	51	0.001904	0.002267	0.99583	0.928773	0	0	1	3
	52	0.035831	0.035397	0.928773	0.593666	0	0	1	3
	53	0.593666	0.267923	0.138411	0.516416	1	0	0	1
	54	0.267081	0.216503	0.516416	0.706435	0	0	1	3
	55	0.706435	0.241992	0.051573	0.739457	1	0	0	1
	56	0.739457	0.216182	0.044361	0.513361	1	0	0	1
	57	0.268991	0.217648	0.513361	0.682557	0	0	1	3
	58	0.168867	0.148576	0.682557	0.856491	0	0	1	3
	59	0.073825	0.069684	0.856491	0.802177	0	0	1	3
	60	0.097254	0.100569	0.802177	0.933577	0	0	1	3
	61	0.933577	0.055969	0.010455	0.923213	1	0	0	1
	62	0.923213	0.064718	0.012069	0.548976	1	0	0	1
	63	0.247031	0.203993	0.548976	0.959199	0	0	1	3
	64	0.959199	0.033801	0.007	0.955427	1	0	0	1
	65	0.955427	0.036855	0.007718	0.923818	1	0	0	1
	66	0.923818	0.064208	0.011974	0.954338	1	0	0	1
	67	0.954338	0.037738	0.007925	0.601437	1	0	0	1
	68	0.601437	0.265213	0.13335	0.588804	1	0	0	1
	69	0.223106	0.188089	0.588804	0.921845	0	0	1	3
	70	0.038988	0.039167	0.921845	0.806188	0	0	1	3
	71	0.806188	0.162328	0.031484	0.880245	1	0	0	1
	72	0.061284	0.058471	0.880245	0.940637	0	0	1	3
	73	0.940637	0.050002	0.009361	0.542043	1	0	0	1
	74	0.542043	0.281733	0.176224	0.954193	1	0	0	1
	75	0.954193	0.037854	0.007953	0.986837	1	0	0	1
Data Testing	1	0.03177	0.127395	0.840835	0.840835	0	0	1	3
	2	0.020057	0.142283	0.83766	0.83766	0	0	1	3
	3	0.123484	0.691659	0.184857	0.691659	0	1	0	2
	4	0.881886	0.080007	0.038107	0.881886	1	0	0	1
	5	0.015812	0.076274	0.907914	0.907914	0	0	1	3

6	0.015919	0.076649	0.907433	0.907433	0	0	1	3
7	0.021791	0.15803	0.820179	0.820179	0	0	1	3
8	0.046292	0.428767	0.524941	0.524941	0	0	1	3
9	0.149991	0.67228	0.177729	0.67228	0	1	0	2
10	0.01193	0.072426	0.915644	0.915644	0	0	1	3
11	0.974212	0.017794	0.007993	0.974212	1	0	0	1
12	0.00555	0.046264	0.948185	0.948185	0	0	1	3
13	0.01469	0.169722	0.815587	0.815587	0	0	1	3
14	0.993048	0.004986	0.001966	0.993048	1	0	0	1
15	0.984531	0.011189	0.00428	0.984531	1	0	0	1
16	0.009167	0.97101	0.019824	0.97101	0	1	0	2
17	0.005663	0.047543	0.946794	0.946794	0	0	1	3
18	0.039324	0.811548	0.149128	0.811548	0	1	0	2
19	0.923212	0.055198	0.02159	0.923212	1	0	0	1
20	0.038146	0.835275	0.126579	0.835275	0	1	0	2
21	0.042399	0.307215	0.650386	0.650386	0	0	1	3
22	0.930679	0.049348	0.019973	0.930679	1	0	0	1
23	0.112677	0.788914	0.098409	0.788914	0	1	0	2
24	0.042634	0.234794	0.722571	0.722571	0	0	1	3
25	0.041852	0.244506	0.713642	0.713642	0	0	1	3

Lampiran 3.9 Nilai Keanggotaan  $\mu_{As}(t)$  Data pada Bagian Antesenden

Data Ke-	Nilai Keanggotaan (Himpunan <i>Fuzzy</i> )		
	$\mu_i^1$	$\mu_i^1$	$\mu_i^2$
1	0.430365	0.566431	9.81E-06
2	0.114903	0.883061	1.65E-05
3	0.003534	0.996416	4.76E-05
4	0.002311	3.91E-09	0.999696
5	0.264051	0.742263	5.82E-06
6	0.000318	0.999708	4.14E-05
7	0.541819	0.465336	5.17E-06
8	0.015848	0.986369	4.8E-05
9	0.078035	0.921872	1.75E-05
10	3.08E-05	6.32E-07	0.99936
11	0.782615	0.214304	6.23E-06
12	0.002688	0.997261	5.12E-05
13	7.43E-06	0.999994	2.92E-05
14	0.055918	0.944545	8.17E-06
15	0.812246	0.185942	6.23E-06
16	0.096099	0.908617	5.6E-06
17	2.08E-05	0.999982	3.37E-05
18	0.001486	0.998467	6.22E-05
19	0.122876	0.880138	1.91E-06
20	0.566197	0.432904	6.05E-06
21	0.67709	0.322475	6.13E-06
22	4.64E-06	2.93E-07	0.999397
23	0.066441	0.940614	3.47E-06
24	3.86E-07	0.008722	0.988848
25	0.019332	0.981326	5.49E-06
26	0.032821	0.971514	4.93E-05
27	0.953415	0.034675	6.76E-05
28	3.89E-05	1.76E-08	0.99955
29	0.99026	0.008378	3.24E-07
30	0.000363	1.76E-09	0.999707
31	0.804795	0.192783	6.23E-06
32	2.31E-05	1.22E-07	0.999464
33	0.480201	0.516007	9.31E-06
34	3.53E-05	2.13E-08	0.999101
35	7.25E-07	0.028011	0.980085
36	0.000159	0.99719	0.005659
37	0.00023	3E-09	0.999701
38	0.00168	0.998356	4.73E-05

	39	4.41E-06	3.32E-07	0.999378
	40	0.004063	2.06E-07	0.999689
	41	6.13E-05	1.14E-08	0.999644
	42	0.431803	0.571936	5.95E-06
	43	5.23E-05	1.27E-08	0.999455
	44	0.010914	0.95373	0.001218
	45	0.000667	1.03E-09	0.999729
	46	0.725383	0.273826	6.18E-06
	47	6.47E-05	0.98282	0.026139
	48	9E-06	0.999993	2.75E-05
	49	0.999989	6.74E-06	4.21E-06
	50	0.000286	2.47E-09	0.999709
	51	0.000105	2.25E-08	0.99958
	52	0.000372	1.97E-09	0.999718
	53	0.992139	0.007737	9.64E-05
	54	0.002973	1.27E-06	0.99942
	55	0.980917	0.016762	3.95E-07
	56	0.867595	0.123835	6.72E-07
	57	0.003414	2.54E-07	0.998145
	58	0.040413	2.09E-11	0.993233
	59	0.000774	1.99E-09	0.999722
	60	0.000611	1.02E-09	0.999727
	61	0.811327	0.18683	6.24E-06
	62	0.799133	0.199494	6.15E-06
	63	3.9E-07	0.011651	0.98861
	64	0.794024	0.202763	6.3E-06
	65	0.577174	0.420326	6.21E-06
	66	0.791406	0.207446	6.11E-06
	67	0.548387	0.457744	5.29E-06
	68	0.939497	0.064743	8.16E-05
	69	2.71E-05	5.51E-05	0.998609
	70	0.00013	5.09E-09	0.999636
	71	0.743525	0.256944	5.88E-06
	72	5.1E-05	9.49E-07	0.999397
	73	0.666027	0.334063	6.13E-06
	74	0.966459	0.025011	0.000152
	75	0.787839	0.208837	6.39E-06
<i>Data Testing</i>	1	0.994772	0.004417	2.77E-07
	2	0.346681	0.654356	9E-06
	3	0.088738	0.057941	0.176256
	4	0.00067	9.91E-10	0.999729
	5	0.90266	0.088965	6.34E-07

6	0.788106	0.210305	6.24E-06
7	0.292867	0.701368	1.11E-05
8	3.62E-05	0.999968	3.63E-05
9	2.4E-06	0.21936	0.850507
10	0.807223	0.190303	6.22E-06
11	0.0002	3.71E-09	0.999675
12	0.017868	0.983192	5.82E-06
13	0.038575	0.962546	1.5E-05
14	0.000511	1.34E-09	0.999723
15	0.000629	1.16E-09	0.999728
16	0.000144	0.998144	0.005287
17	0.745434	0.251823	6.18E-06
18	0.08352	0.919309	6.01E-05
19	0.000694	1.06E-09	0.999728
20	0.036764	0.888202	0.000827
21	0.76444	0.232922	6.2E-06
22	0.0001	6.54E-09	0.999604
23	0.000498	0.005789	0.90254
24	0.805368	0.193119	6.18E-06
25	0.794156	0.204014	6.24E-06



**Lampiran 3.10** Data Cluster 1

	No	Input								TARGET
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	
Training	1	0.5374	0.0165	0.963	0.9967	7.7265	0.7217	0.6555	0.9917	2
	2	0.4853	0.0218	0.9558	0.9956	6.4368	0.7912	0.5468	0.9891	2
	3	0.53	0.0206	0.9542	0.9959	7.5848	0.7414	0.648	0.9897	2
	4	0.5496	0.0195	0.9549	0.9961	7.9548	0.7173	0.6756	0.9903	2
	5	0.5231	0.0198	0.9568	0.996	7.4441	0.7463	0.6359	0.9901	2
	6	0.5112	0.0215	0.954	0.9957	7.1908	0.7637	0.6163	0.9892	2
	7	0.5119	0.0173	0.9633	0.9965	7.1827	0.7508	0.6115	0.9913	2
	8	0.6087	0.017	0.9547	0.9966	8.9521	0.6451	0.7419	0.9915	3
	9	0.4781	0.024	0.9519	0.9952	6.1142	0.8047	0.5158	0.988	3
	10	0.4817	0.0232	0.9533	0.9954	6.3077	0.7988	0.5352	0.9884	3
	11	0.5129	0.0224	0.9518	0.9955	7.2326	0.7644	0.6207	0.9888	3
	12	0.5102	0.0233	0.9502	0.9953	7.1768	0.7695	0.6167	0.9884	3
	13	0.5397	0.0197	0.9552	0.9961	7.7719	0.7287	0.6619	0.9901	3
	14	0.5413	0.0186	0.9578	0.9963	7.8013	0.7237	0.6632	0.9907	3
	15	0.5103	0.0235	0.9497	0.9953	7.18	0.77	0.6171	0.9883	3
	16	0.5416	0.0213	0.9513	0.9957	7.8092	0.731	0.6661	0.9893	3
	17	0.6078	0.0197	0.9473	0.9961	8.9285	0.6538	0.7422	0.9902	3
	18	0.4882	0.0271	0.9441	0.9946	6.6425	0.8027	0.572	0.9864	3
	19	0.515	0.0209	0.9549	0.9958	7.2737	0.7581	0.6228	0.9895	3
	20	0.619	0.0169	0.9535	0.9966	9.1157	0.6336	0.752	0.9915	3
	21	0.5417	0.0197	0.955	0.9961	7.8102	0.7265	0.6649	0.9901	3
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	TARGET
Testing	1	0.8616	0.0044	0.9672	0.9991	13.0371	0.2846	0.9256	0.9978	1
	2	0.8712	0.0047	0.9624	0.9991	13.1984	0.2709	0.9312	0.9977	2
	3	0.8311	0.005	0.9694	0.999	12.5169	0.3314	0.9074	0.9975	2
	4	0.8235	0.0055	0.9681	0.9989	12.3862	0.3436	0.9028	0.9973	2
	5	0.8329	0.0054	0.9667	0.9989	12.5405	0.3301	0.9085	0.9973	3
	6	0.8418	0.0041	0.9733	0.9992	12.7048	0.3131	0.9138	0.9979	3

**Lampiran 3.11** Data Cluster 2

	No	Input								TARGET
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	
Training	1	0.5495	0.0196	0.9545	0.9961	7.9526	0.7179	0.6755	0.9902	1
	2	0.5522	0.0184	0.9571	0.9963	8.0032	0.7114	0.6783	0.9908	1
	3	0.5578	0.0181	0.9573	0.9964	8.1023	0.7044	0.6853	0.9909	1
	4	0.5151	0.0204	0.9562	0.9959	7.2724	0.7564	0.6222	0.9898	1

	5	0.5968	0.018	0.9533	0.9964	8.7578	0.6617	0.7302	0.991	1
	6	0.5306	0.0223	0.9502	0.9955	7.6003	0.7453	0.6507	0.9889	1
	7	0.601	0.0209	0.9447	0.9958	8.816	0.6649	0.736	0.9895	1
	8	0.5526	0.0197	0.9541	0.9961	8.0085	0.7145	0.6797	0.9902	1
	9	0.537	0.0197	0.9556	0.9961	7.7205	0.7314	0.6579	0.9902	1
	10	0.5597	0.0178	0.958	0.9964	8.1367	0.7015	0.6875	0.9911	1
	11	0.5821	0.0223	0.9437	0.9955	8.5087	0.6895	0.7168	0.9888	1
	12	0.5454	0.0173	0.9604	0.9965	7.8794	0.7152	0.668	0.9913	1
	13	0.5123	0.0227	0.9512	0.9955	7.2204	0.7658	0.6199	0.9886	1
	14	0.5158	0.0201	0.9567	0.996	7.2871	0.7549	0.6232	0.9899	1
	15	0.5718	0.0201	0.9508	0.996	8.3409	0.695	0.7038	0.99	1
	16	0.5681	0.0175	0.9578	0.9965	8.2828	0.6917	0.6977	0.9913	1
	17	0.4842	0.0229	0.9535	0.9954	6.4166	0.7955	0.546	0.9885	1
	18	0.5364	0.0189	0.9576	0.9962	7.7083	0.7297	0.6563	0.9906	1
	19	0.52	0.0205	0.9554	0.9959	7.381	0.7516	0.6314	0.9897	1
	20	0.5127	0.0254	0.9452	0.9949	7.2445	0.7724	0.6242	0.9873	1
	21	0.5286	0.0177	0.961	0.9965	7.5512	0.7345	0.6427	0.9911	1
	22	0.6022	0.0194	0.9488	0.9961	8.8407	0.6594	0.7365	0.9903	2
	23	0.6454	0.0159	0.953	0.9968	9.5297	0.6007	0.7758	0.992	2
	24	0.5596	0.0194	0.9539	0.9961	8.1334	0.7064	0.6886	0.9903	2
	25	0.6244	0.018	0.9497	0.9964	9.1948	0.6305	0.7575	0.991	2
	26	0.6559	0.0152	0.9537	0.997	9.6957	0.5865	0.7847	0.9924	2
	27	0.5631	0.0236	0.943	0.9953	8.1896	0.7138	0.6957	0.9882	2
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	TARGET
<i>Testing</i>	1	0.6585	0.0139	0.9576	0.9972	9.7429	0.5796	0.7865	0.9931	1
	2	0.6706	0.0155	0.9506	0.9969	9.9188	0.5698	0.7971	0.9922	1
	3	0.6389	0.0176	0.9487	0.9965	9.4214	0.6129	0.7707	0.9912	2
	4	0.597	0.0174	0.9549	0.9965	8.7632	0.6598	0.7301	0.9913	3
	5	0.6047	0.0217	0.9421	0.9957	8.8737	0.6626	0.7401	0.9892	3
	6	0.6723	0.0137	0.9565	0.9973	9.9554	0.5628	0.7979	0.9932	3

**Lampiran 3.12** Data Cluster 3

	No	<i>Input</i>								TARGET
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	
<i>Training</i>	1	0.7324	0.0094	0.9637	0.9981	10.9083	0.4767	0.843	0.9953	1
	2	0.7651	0.0097	0.9569	0.9981	11.4096	0.436	0.8657	0.9951	1
	3	0.8368	0.0068	0.9567	0.9986	12.5905	0.3286	0.9111	0.9966	1
	4	0.6886	0.014	0.9528	0.9972	10.2018	0.5443	0.8111	0.993	1
	5	0.7918	0.0049	0.9759	0.999	11.8837	0.3852	0.8827	0.9975	2
	6	0.8922	0.0035	0.9662	0.9993	13.5857	0.234	0.9431	0.9982	2
	7	0.7884	0.0082	0.9596	0.9984	11.7909	0.4006	0.881	0.9959	2

	8	0.7495	0.0059	0.976	0.9988	11.2084	0.4437	0.8544	0.9971	2
	9	0.675	0.0161	0.948	0.9968	9.9827	0.5662	0.8009	0.992	2
	10	0.7934	0.0054	0.9734	0.9989	11.9031	0.3847	0.8838	0.9973	2
	11	0.8594	0.0065	0.9518	0.9987	12.9744	0.2944	0.9244	0.9968	2
	12	0.6875	0.0123	0.9592	0.9975	10.1973	0.5405	0.8098	0.9939	2
	13	0.7868	0.0053	0.9747	0.9989	11.7992	0.3932	0.8795	0.9974	2
	14	0.8003	0.0046	0.9764	0.9991	12.0232	0.3727	0.8882	0.9977	2
	15	0.812	0.0055	0.9699	0.9989	12.1986	0.3599	0.8957	0.9973	2
	16	0.79	0.0057	0.9723	0.9989	11.8448	0.3903	0.8816	0.9972	2
	17	0.7816	0.0081	0.9613	0.9984	11.6841	0.4095	0.8765	0.9959	2
	18	0.7839	0.0059	0.9719	0.9988	11.7446	0.3993	0.8777	0.997	3
	19	0.6805	0.0129	0.9579	0.9974	10.0858	0.5508	0.8044	0.9935	3
	20	0.6786	0.0102	0.9674	0.998	10.0781	0.5447	0.802	0.9949	3
	21	0.7033	0.0064	0.9779	0.9987	10.493	0.5025	0.8206	0.9968	3
	22	0.7419	0.0078	0.9689	0.9984	11.0693	0.4599	0.8494	0.9961	3
	23	0.8938	0.0032	0.9686	0.9994	13.6179	0.2306	0.9439	0.9984	3
	24	0.6863	0.0146	0.9512	0.9971	10.1638	0.5486	0.8095	0.9927	3
	25	0.692	0.0131	0.9555	0.9974	10.2592	0.5377	0.8135	0.9934	3
	26	0.8084	0.0047	0.9749	0.9991	12.1517	0.362	0.8933	0.9977	3
	27	0.7516	0.0103	0.9567	0.9979	11.1949	0.4552	0.8566	0.9948	3
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	TARGET
Testing	1	0.4794	0.022	0.9559	0.9956	6.0806	0.7979	0.51	0.989	1
	2	0.5373	0.0205	0.9537	0.9959	7.728	0.7333	0.6592	0.9897	1
	3	0.494	0.0193	0.9603	0.9961	6.7117	0.7753	0.571	0.9903	1
	4	0.4923	0.0229	0.9529	0.9954	6.7063	0.7871	0.574	0.9886	1
	5	0.5401	0.0184	0.9584	0.9963	7.7799	0.7243	0.6614	0.9908	1
	6	0.5751	0.019	0.9532	0.9962	8.3999	0.6882	0.707	0.9905	1
	7	0.5185	0.0209	0.9548	0.9958	7.3493	0.7542	0.6291	0.9896	1
	8	0.52	0.0183	0.9604	0.9963	7.3717	0.7452	0.6286	0.9908	2
	9	0.5396	0.0207	0.9529	0.9959	7.7704	0.7316	0.6627	0.9896	2
	10	0.5202	0.0198	0.9569	0.996	7.3821	0.7495	0.6309	0.9901	2
	11	0.5183	0.0202	0.9562	0.996	7.3434	0.7526	0.628	0.9899	3
	12	0.4941	0.0239	0.9504	0.9952	6.7745	0.7881	0.5814	0.988	3
	13	0.4997	0.0222	0.9536	0.9956	6.9111	0.7777	0.5923	0.9889	3

Lampiran 3.13 Hasil Penyederhanaan Cluster 1

	No	Input								TARGET
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	
Training	1	0.5374	0.0165	0.963	0.9967	7.7265	0.7217	0.6555	0.9917	2.157245
	2	0.4853	0.0218	0.9558	0.9956	6.4368	0.7912	0.5468	0.9891	2.356029
	3	0.53	0.0206	0.9542	0.9959	7.5848	0.7414	0.648	0.9897	2.66125

	4	0.5496	0.0195	0.9549	0.9961	7.9548	0.7173	0.6756	0.9903	2.615966
	5	0.5231	0.0198	0.9568	0.996	7.4441	0.7463	0.6359	0.9901	2.338167
	6	0.5112	0.0215	0.954	0.9957	7.1908	0.7637	0.6163	0.9892	2.626497
	7	0.5119	0.0173	0.9633	0.9965	7.1827	0.7508	0.6115	0.9913	1.844593
	8	0.6087	0.017	0.9547	0.9966	8.9521	0.6451	0.7419	0.9915	2.872241
	9	0.4781	0.024	0.9519	0.9952	6.1142	0.8047	0.5158	0.988	2.93972
	10	0.4817	0.0232	0.9533	0.9954	6.3077	0.7988	0.5352	0.9884	2.7701
	11	0.5129	0.0224	0.9518	0.9955	7.2326	0.7644	0.6207	0.9888	2.644474
	12	0.5102	0.0233	0.9502	0.9953	7.1768	0.7695	0.6167	0.9884	2.600172
	13	0.5397	0.0197	0.9552	0.9961	7.7719	0.7287	0.6619	0.9901	2.790421
	14	0.5413	0.0186	0.9578	0.9963	7.8013	0.7237	0.6632	0.9907	2.511035
	15	0.5103	0.0235	0.9497	0.9953	7.18	0.77	0.6171	0.9883	2.69818
	16	0.5416	0.0213	0.9513	0.9957	7.8092	0.731	0.6661	0.9893	2.896563
	17	0.6078	0.0197	0.9473	0.9961	8.9285	0.6538	0.7422	0.9902	3.071139
	18	0.4882	0.0271	0.9441	0.9946	6.6425	0.8027	0.572	0.9864	3.269514
	19	0.515	0.0209	0.9549	0.9958	7.2737	0.7581	0.6228	0.9895	2.542808
	20	0.619	0.0169	0.9535	0.9966	9.1157	0.6336	0.752	0.9915	2.989028
	21	0.5417	0.0197	0.955	0.9961	7.8102	0.7265	0.6649	0.9901	2.804858
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	TARGET
<i>Testing</i>	1	0.8616	0.0044	0.9672	0.9991	13.0371	0.2846	0.9256	0.9978	-3.45394
	2	0.8712	0.0047	0.9624	0.9991	13.1984	0.2709	0.9312	0.9977	-4.54225
	3	0.8311	0.005	0.9694	0.999	12.5169	0.3314	0.9074	0.9975	-1.45224
	4	0.8235	0.0055	0.9681	0.9989	12.3862	0.3436	0.9028	0.9973	-1.25322
	5	0.8329	0.0054	0.9667	0.9989	12.5405	0.3301	0.9085	0.9973	-1.73471
	6	0.8418	0.0041	0.9733	0.9992	12.7048	0.3131	0.9138	0.9979	-1.61234

**Lampiran 3.14** Hasil Penyederhanaan *Cluster 2*

	<i>No</i>	<i>Input</i>								TARGET
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	
<i>Training</i>	1	0.5495	0.0196	0.9545	0.9961	7.9526	0.7179	0.6755	0.9902	1.102606
	2	0.5522	0.0184	0.9571	0.9963	8.0032	0.7114	0.6783	0.9908	1.067197
	3	0.5578	0.0181	0.9573	0.9964	8.1023	0.7044	0.6853	0.9909	1.041101
	4	0.5151	0.0204	0.9562	0.9959	7.2724	0.7564	0.6222	0.9898	1.001077
	5	0.5968	0.018	0.9533	0.9964	8.7578	0.6617	0.7302	0.991	1.39243
	6	0.5306	0.0223	0.9502	0.9955	7.6003	0.7453	0.6507	0.9889	1.117205
	7	0.601	0.0209	0.9447	0.9958	8.816	0.6649	0.736	0.9895	1.5507
	8	0.5526	0.0197	0.9541	0.9961	8.0085	0.7145	0.6797	0.9902	1.081003
	9	0.537	0.0197	0.9556	0.9961	7.7205	0.7314	0.6579	0.9902	1.048296

	10	0.5597	0.0178	0.958	0.9964	8.1367	0.7015	0.6875	0.9911	1.07213
	11	0.5821	0.0223	0.9437	0.9955	8.5087	0.6895	0.7168	0.9888	1.40737
	12	0.5454	0.0173	0.9604	0.9965	7.8794	0.7152	0.668	0.9913	0.967363
	13	0.5123	0.0227	0.9512	0.9955	7.2204	0.7658	0.6199	0.9886	1.049705
	14	0.5158	0.0201	0.9567	0.996	7.2871	0.7549	0.6232	0.9899	0.981161
	15	0.5718	0.0201	0.9508	0.996	8.3409	0.695	0.7038	0.99	1.249876
	16	0.5681	0.0175	0.9578	0.9965	8.2828	0.6917	0.6977	0.9913	1.132337
	17	0.4842	0.0229	0.9535	0.9954	6.4166	0.7955	0.546	0.9885	0.958004
	18	0.5364	0.0189	0.9576	0.9962	7.7083	0.7297	0.6563	0.9906	1.020336
	19	0.52	0.0205	0.9554	0.9959	7.381	0.7516	0.6314	0.9897	1.014112
	20	0.5127	0.0254	0.9452	0.9949	7.2445	0.7724	0.6242	0.9873	1.14347
	21	0.5286	0.0177	0.961	0.9965	7.5512	0.7345	0.6427	0.9911	0.921709
	22	0.6022	0.0194	0.9488	0.9961	8.8407	0.6594	0.7365	0.9903	1.508067
	23	0.6454	0.0159	0.953	0.9968	9.5297	0.6007	0.7758	0.992	1.930834
	24	0.5596	0.0194	0.9539	0.9961	8.1334	0.7064	0.6886	0.9903	1.159639
	25	0.6244	0.018	0.9497	0.9964	9.1948	0.6305	0.7575	0.991	1.71246
	26	0.6559	0.0152	0.9537	0.997	9.6957	0.5865	0.7847	0.9924	2.073732
	27	0.5631	0.0236	0.943	0.9953	8.1896	0.7138	0.6957	0.9882	1.296082
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	TARGET
<i>Testing</i>	1	0.6585	0.0139	0.9576	0.9972	9.7429	0.5796	0.7865	0.9931	2.038258
	2	0.6706	0.0155	0.9506	0.9969	9.9188	0.5698	0.7971	0.9922	2.345002
	3	0.6389	0.0176	0.9487	0.9965	9.4214	0.6129	0.7707	0.9912	1.91656
	4	0.597	0.0174	0.9549	0.9965	8.7632	0.6598	0.7301	0.9913	1.378673
	5	0.6047	0.0217	0.9421	0.9957	8.8737	0.6626	0.7401	0.9892	1.615077
	6	0.6723	0.0137	0.9565	0.9973	9.9554	0.5628	0.7979	0.9932	2.269623

**Lampiran 3.15** Hasil Penyederhanaan *Cluster 3*

	<i>No</i>	<i>Input</i>								TARGET
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	
<i>Training</i>	1	0.7324	0.0094	0.9637	0.9981	10.9083	0.4767	0.843	0.9953	-1.56221
	2	0.7651	0.0097	0.9569	0.9981	11.4096	0.436	0.8657	0.9951	-9.8659
	3	0.8368	0.0068	0.9567	0.9986	12.5905	0.3286	0.9111	0.9966	-3.76784
	4	0.6886	0.014	0.9528	0.9972	10.2018	0.5443	0.8111	0.993	2.635767
	5	0.7918	0.0049	0.9759	0.999	11.8837	0.3852	0.8827	0.9975	5.537237
	6	0.8922	0.0035	0.9662	0.9993	13.5857	0.234	0.9431	0.9982	-2.93649
	7	0.7884	0.0082	0.9596	0.9984	11.7909	0.4006	0.881	0.9959	-4.27856
	8	0.7495	0.0059	0.976	0.9988	11.2084	0.4437	0.8544	0.9971	-1.52057
	9	0.675	0.0161	0.948	0.9968	9.9827	0.5662	0.8009	0.992	-13.5482

	10	0.7934	0.0054	0.9734	0.9989	11.9031	0.3847	0.8838	0.9973	5.957853
	11	0.8594	0.0065	0.9518	0.9987	12.9744	0.2944	0.9244	0.9968	0.742677
	12	0.6875	0.0123	0.9592	0.9975	10.1973	0.5405	0.8098	0.9939	9.533622
	13	0.7868	0.0053	0.9747	0.9989	11.7992	0.3932	0.8795	0.9974	6.103501
	14	0.8003	0.0046	0.9764	0.9991	12.0232	0.3727	0.8882	0.9977	2.871018
	15	0.812	0.0055	0.9699	0.9989	12.1986	0.3599	0.8957	0.9973	11.93847
	16	0.79	0.0057	0.9723	0.9989	11.8448	0.3903	0.8816	0.9972	2.287347
	17	0.7816	0.0081	0.9613	0.9984	11.6841	0.4095	0.8765	0.9959	2.664656
	18	0.7839	0.0059	0.9719	0.9988	11.7446	0.3993	0.8777	0.997	6.130293
	19	0.6805	0.0129	0.9579	0.9974	10.0858	0.5508	0.8044	0.9935	12.96887
	20	0.6786	0.0102	0.9674	0.998	10.0781	0.5447	0.802	0.9949	17.60401
	21	0.7033	0.0064	0.9779	0.9987	10.493	0.5025	0.8206	0.9968	0.195826
	22	0.7419	0.0078	0.9689	0.9984	11.0693	0.4599	0.8494	0.9961	10.29193
	23	0.8938	0.0032	0.9686	0.9994	13.6179	0.2306	0.9439	0.9984	2.209426
	24	0.6863	0.0146	0.9512	0.9971	10.1638	0.5486	0.8095	0.9927	-7.53339
	25	0.692	0.0131	0.9555	0.9974	10.2592	0.5377	0.8135	0.9934	5.679003
	26	0.8084	0.0047	0.9749	0.9991	12.1517	0.362	0.8933	0.9977	5.896289
	27	0.7516	0.0103	0.9567	0.9979	11.1949	0.4552	0.8566	0.9948	-6.23462
		Energy (X1)	Contrast (X2)	Correlation (X3)	IDM (X5)	Sum Variance (X7)	Entropy (X9)	Maz Prob (X12)	Homogenitas (X13)	TARGET
Testing	1	0.4794	0.022	0.9559	0.9956	6.0806	0.7979	0.51	0.989	148.5803
	2	0.5373	0.0205	0.9537	0.9959	7.728	0.7333	0.6592	0.9897	178.3524
	3	0.494	0.0193	0.9603	0.9961	6.7117	0.7753	0.571	0.9903	217.1485
	4	0.4923	0.0229	0.9529	0.9954	6.7063	0.7871	0.574	0.9886	235.8533
	5	0.5401	0.0184	0.9584	0.9963	7.7799	0.7243	0.6614	0.9908	174.4604
	6	0.5751	0.019	0.9532	0.9962	8.3999	0.6882	0.707	0.9905	112.0364
	7	0.5185	0.0209	0.9548	0.9958	7.3493	0.7542	0.6291	0.9896	209.1909
	8	0.52	0.0183	0.9604	0.9963	7.3717	0.7452	0.6286	0.9908	198.4854
	9	0.5396	0.0207	0.9529	0.9959	7.7704	0.7316	0.6627	0.9896	175.9369
	10	0.5202	0.0198	0.9569	0.996	7.3821	0.7495	0.6309	0.9901	213.0844
	11	0.5183	0.0202	0.9562	0.996	7.3434	0.7526	0.628	0.9899	211.3316
	12	0.4941	0.0239	0.9504	0.9952	6.7745	0.7881	0.5814	0.988	242.5341
	13	0.4997	0.0222	0.9536	0.9956	6.9111	0.7777	0.5923	0.9889	236.0821

Lampiran 3.16 Nilai  $\mu_s(t)$  Data Training

Data ke-	Cluster				Cluster		
	1	2	3		1	2	3
1	2.384274	1.714728	1.865298	39	1.72784	1.759374	1.970814

2	1.840453	1.698375	1.383215	40	2.271382	2.118826	2.155403
3	1.260959	1.457419	1.332237	41	2.120544	2.055468	2.14498
4	1.375289	1.301977	1.288175	42	1.306674	1.37365	1.498092
5	1.364549	1.442223	1.521103	43	2.130502	1.999205	2.134477
6	1.350747	1.32871	1.408121	44	2.348118	2.465053	2.391077
7	1.707549	1.747081	1.674553	45	2.292847	2.79852	2.600908
8	1.350051	1.285809	1.455235	46	2.474303	2.280413	2.22694
9	1.827399	1.537797	1.675114	47	2.138471	1.944786	2.145608
10	2.204833	1.923286	1.659162	48	1.173232	1.224378	1.175873
11	1.656077	1.821653	1.715658	49	2.145497	2.031512	2.074089
12	1.208369	1.291272	1.282054	50	2.373736	2.733474	2.36199
13	1.173178	1.188092	1.17583	51	1.865289	1.462749	1.654376
14	1.25216	1.250562	1.303167	52	2.428308	2.722317	2.436705
15	2.686704	2.734844	2.774501	53	2.714583	2.77097	2.775229
16	1.336953	1.428975	1.456474	54	2.432172	2.632906	2.702808
17	1.185972	1.213795	1.173811	55	2.686281	2.626824	2.746908
18	1.189352	1.259885	1.200695	56	2.260173	2.253082	2.340919
19	2.141318	2.076219	2.051456	57	2.794534	2.740464	2.758405
20	1.245181	1.329547	1.40077	58	2.736723	2.800266	2.792023
21	1.719365	1.822383	1.851172	59	2.615311	2.626574	2.646022
22	1.384421	1.51775	1.204988	60	2.708468	2.793177	2.746672
23	1.338678	1.408522	1.247841	61	2.678768	2.653975	2.751871

24	2.155377	2.098338	1.704269	62	2.603681	2.68394	2.760804
25	1.372234	1.242944	1.3105	63	2.135129	2.000291	2.118143
26	1.793388	1.993639	1.385168	64	2.040358	2.080582	2.026494
27	1.704669	1.946485	1.86691	65	1.242789	1.382544	1.393757
28	2.108174	1.954171	2.114672	66	2.54728	2.700092	2.670187
29	1.636469	1.317892	1.597877	67	1.700337	1.54832	2.159205
30	2.02811	2.029603	1.980527	68	2.623084	2.542084	2.375264
31	2.397476	2.258283	2.380102	69	2.288613	2.641527	2.601506
32	1.778547	1.493064	2.1829	70	2.332161	2.571251	2.268581
33	1.911001	1.558034	1.706022	71	2.692338	2.707937	2.740614
34	2.083242	2.044879	1.990726	72	2.292821	2.35868	2.182506
35	1.994719	1.984244	2.566225	73	1.932879	1.89877	1.926248
36	2.158314	2.013011	2.282373	74	2.64708	2.779905	2.756087
37	2.32809	2.60371	2.309218	75	2.117734	2.087324	2.076858
38	1.619601	1.302419	1.563347				

**Lampiran 3.17** Nilai  $\mu_s(t)$  Data Testing

Data ke-	Cluster			Data ke	Cluster		
	1	2	3		1	2	3
1	2.11087	1.641312	1.655695	14	2.425735	2.786921	2.495744
2	1.866991	1.543055	2.386479	15	2.3229	2.794028	2.560265
3	2.249478	2.378762	2.579985	16	1.882614	1.984413	2.558316
4	2.150691	2.799208	2.624814	17	1.390549	1.442841	1.67769



5	1.457795	1.212688	1.275463	18	2.222971	2.089519	2.623802
6	2.764034	2.678997	2.569123	19	2.024518	2.797268	2.573969
7	1.22495	1.292387	1.408194	20	2.754151	1.985177	2.691234
8	1.18393	1.203696	1.184444	21	1.709529	1.811984	1.843149
9	1.967126	1.951372	2.32188	22	2.251567	2.319464	2.258006
10	2.441661	2.007331	2.421494	23	2.097747	2.406181	2.703259
11	1.28003	1.55076	1.200832	24	2.719859	2.730702	2.454325
12	1.320396	1.230906	1.322033	25	2.728705	2.6495	2.550423
13	1.512292	1.425862	1.680602				

**Lampiran 3.18** Target, *Output* dan Error Model RNF

	Data ke-	Target	<i>Output</i> (RNF)	Status <i>Output</i>	Error	
Data <i>Training</i>	1	1	2.003802	2	-1.0038	Tidak Cocok
	2	1	1.714728	2	-0.71473	Tidak Cocok
	3	1	1.865298	2	-0.8653	Tidak Cocok
	4	1	1.269811	1	-0.26981	Cocok
	5	1	1.398375	1	-0.39838	Cocok
	6	1	1.383215	1	-0.38322	Cocok
	7	1	1.260959	1	-0.26096	Cocok
	8	1	1.457419	1	-0.45742	Cocok
	9	1	1.332237	1	-0.33224	Cocok
	10	1	1.375289	1	-0.37529	Cocok
	11	1	1.301977	1	-0.30198	Cocok
	12	1	1.288175	1	-0.28818	Cocok
	13	1	1.364549	1	-0.36455	Cocok
	14	1	1.442223	1	-0.44222	Cocok
	15	1	1.521103	2	-0.5211	Tidak Cocok
	16	1	1.350747	1	-0.35075	Cocok
	17	1	1.32871	1	-0.32871	Cocok
	18	1	1.408121	1	-0.40812	Cocok
	19	1	1.707549	2	-0.70755	Tidak Cocok
	20	1	1.747081	2	-0.74708	Tidak Cocok

21	1	1.674553	2	-0.67455	Tidak Cocok
22	1	1.350051	1	-0.35005	Cocok
23	1	1.285809	1	-0.28581	Cocok
24	1	1.455235	1	-0.45524	Cocok
25	1	1.827399	2	-0.8274	Tidak Cocok
26	2	1.537797	2	0.462203	Cocok
27	2	1.675114	2	0.324886	Cocok
28	2	2.204833	2	-0.20483	Cocok
29	2	1.923286	2	0.076714	Cocok
30	2	1.659162	2	0.340838	Cocok
31	2	1.656077	2	0.343923	Cocok
32	2	1.821653	2	0.178347	Cocok
33	2	1.715658	2	0.284342	Cocok
34	2	1.208369	1	0.791631	Tidak Cocok
35	2	1.291272	1	0.708728	Tidak Cocok
36	2	1.282054	1	0.717946	Tidak Cocok
37	2	1.173178	1	0.826822	Tidak Cocok
38	2	1.188092	1	0.811908	Tidak Cocok
39	2	1.17583	1	0.82417	Tidak Cocok
40	2	1.25216	1	0.74784	Tidak Cocok
41	2	1.250562	1	0.749438	Tidak Cocok
42	2	1.303167	1	0.696833	Tidak Cocok
43	2	2.686704	3	-0.6867	Tidak Cocok
44	2	2.734844	3	-0.73484	Tidak Cocok
45	2	2.774501	3	-0.7745	Tidak Cocok
46	2	1.336953	1	0.663047	Tidak Cocok
47	2	1.428975	1	0.571025	Tidak Cocok
48	2	1.456474	1	0.543526	Tidak Cocok
49	2	1.185972	1	0.814028	Tidak Cocok
50	2	1.213795	1	0.786205	Tidak Cocok
51	2	1.173811	1	0.826189	Tidak Cocok
52	3	1.189352	1	1.810648	Tidak Cocok
53	3	1.259885	1	1.740115	Tidak Cocok
54	3	1.200695	1	1.799305	Tidak Cocok
55	3	2.141318	2	0.858682	Tidak Cocok
56	3	2.076219	2	0.923781	Tidak Cocok
57	3	2.051456	2	0.948544	Tidak Cocok
58	3	1.245181	1	1.754819	Tidak Cocok
59	3	1.329547	1	1.670453	Tidak Cocok
60	3	1.40077	1	1.59923	Tidak Cocok
61	3	1.719365	2	1.280635	Tidak Cocok
62	3	1.822383	2	1.177617	Tidak Cocok

	63	3	1.851172	2	1.148828	Tidak Cocok
	64	3	1.384421	1	1.615579	Tidak Cocok
	65	3	1.51775	2	1.48225	Tidak Cocok
	66	3	1.204988	1	1.795012	Tidak Cocok
	67	3	1.338678	1	1.661322	Tidak Cocok
	68	3	1.408522	1	1.591478	Tidak Cocok
	69	3	1.247841	1	1.752159	Tidak Cocok
	70	3	2.155377	2	0.844623	Tidak Cocok
	71	3	2.098338	2	0.901662	Tidak Cocok
	72	3	1.704269	2	1.295731	Tidak Cocok
	73	3	1.372234	1	1.627766	Tidak Cocok
	74	3	1.242944	1	1.757056	Tidak Cocok
	75	3	1.3105	1	1.6895	Tidak Cocok
Data Testing	1	1	2.11087	2	-1.11087	Tidak Cocok
	2	1	1.641312	2	-0.64131	Tidak Cocok
	3	1	1.655695	2	-0.6557	Tidak Cocok
	4	1	1.866991	2	-0.86699	Tidak Cocok
	5	1	1.543055	2	-0.54306	Tidak Cocok
	6	1	2.386479	2	-1.38648	Tidak Cocok
	7	1	2.249478	2	-1.24948	Tidak Cocok
	8	1	2.378762	2	-1.37876	Tidak Cocok
	9	1	2.579985	3	-1.57999	Tidak Cocok
	10	1	2.150691	2	-1.15069	Tidak Cocok
	11	2	2.799208	3	-0.79921	Tidak Cocok
	12	2	2.624814	3	-0.62481	Tidak Cocok
	13	2	1.457795	1	0.542205	Tidak Cocok
	14	2	1.212688	1	0.787312	Tidak Cocok
	15	2	1.275463	1	0.724537	Tidak Cocok
	16	2	2.764034	3	-0.76403	Tidak Cocok
	17	2	2.678997	3	-0.679	Tidak Cocok
	18	3	2.569123	3	0.430877	Cocok
	19	3	1.22495	1	1.77505	Tidak Cocok
	20	3	1.292387	1	1.707613	Tidak Cocok
	21	3	1.408194	1	1.591806	Tidak Cocok
	22	3	1.18393	1	1.81607	Tidak Cocok
	23	3	1.203696	1	1.796304	Tidak Cocok
	24	3	1.184444	1	1.815556	Tidak Cocok
	25	3	1.967126	2	1.032874	Tidak Cocok

**Lampiran 3.19** *Performance Measure Model RNF*

	Data ke-	<i>Output</i>	Status <i>Output</i>	Target	<i>Performance Measure</i>
Data Training	1	2.003802	Tumor Jinak	Normal	FP
	2	1.714728	Tumor Jinak	Normal	FP
	3	1.865298	Tumor Jinak	Normal	FP
	4	1.269811	Normal	Normal	TN
	5	1.398375	Normal	Normal	TN
	6	1.383215	Normal	Normal	TN
	7	1.260959	Normal	Normal	TN
	8	1.457419	Normal	Normal	TN
	9	1.332237	Normal	Normal	TN
	10	1.375289	Normal	Normal	TN
	11	1.301977	Normal	Normal	TN
	12	1.288175	Normal	Normal	TN
	13	1.364549	Normal	Normal	TN
	14	1.442223	Normal	Normal	TN
	15	1.521103	Tumor Jinak	Normal	FP
	16	1.350747	Normal	Normal	TN
	17	1.32871	Normal	Normal	TN
	18	1.408121	Normal	Normal	TN
	19	1.707549	Tumor Jinak	Normal	FP
	20	1.747081	Tumor Jinak	Normal	FP
	21	1.674553	Tumor Jinak	Normal	FP
	22	1.350051	Normal	Normal	TN
	23	1.285809	Normal	Normal	TN
	24	1.455235	Normal	Normal	TN
	25	1.827399	Tumor Jinak	Normal	FP
	26	1.537797	Tumor Jinak	Tumor Jinak	TP
	27	1.675114	Tumor Jinak	Tumor Jinak	TP
	28	2.204833	Tumor Jinak	Tumor Jinak	TP
	29	1.923286	Tumor Jinak	Tumor Jinak	TP
	30	1.659162	Tumor Jinak	Tumor Jinak	TP
	31	1.656077	Tumor Jinak	Tumor Jinak	TP
	32	1.821653	Tumor Jinak	Tumor Jinak	TP
	33	1.715658	Tumor Jinak	Tumor Jinak	TP
	34	1.208369	Normal	Tumor Jinak	FN
	35	1.291272	Normal	Tumor Jinak	FN
	36	1.282054	Normal	Tumor Jinak	FN
	37	1.173178	Normal	Tumor Jinak	FN
	38	1.188092	Normal	Tumor Jinak	FN
	39	1.17583	Normal	Tumor Jinak	FN

	40	1.25216	Normal	Tumor Jinak	FN
	41	1.250562	Normal	Tumor Jinak	FN
	42	1.303167	Normal	Tumor Jinak	FN
	43	2.686704	Tumor Ganas	Tumor Jinak	TP
	44	2.734844	Tumor Ganas	Tumor Jinak	TP
	45	2.774501	Tumor Ganas	Tumor Jinak	TP
	46	1.336953	Normal	Tumor Jinak	FN
	47	1.428975	Normal	Tumor Jinak	FN
	48	1.456474	Normal	Tumor Jinak	FN
	49	1.185972	Normal	Tumor Jinak	FN
	50	1.213795	Normal	Tumor Jinak	FN
	51	1.173811	Normal	Tumor Jinak	FN
	52	1.189352	Normal	Tumor Ganas	FN
	53	1.259885	Normal	Tumor Ganas	FN
	54	1.200695	Normal	Tumor Ganas	FN
	55	2.141318	Tumor Jinak	Tumor Ganas	TP
	56	2.076219	Tumor Jinak	Tumor Ganas	TP
	57	2.051456	Tumor Jinak	Tumor Ganas	TP
	58	1.245181	Normal	Tumor Ganas	FN
	59	1.329547	Normal	Tumor Ganas	FN
	60	1.40077	Normal	Tumor Ganas	FN
	61	1.719365	Tumor Jinak	Tumor Ganas	TP
	62	1.822383	Tumor Jinak	Tumor Ganas	TP
	63	1.851172	Tumor Jinak	Tumor Ganas	TP
	64	1.384421	Normal	Tumor Ganas	FN
	65	1.51775	Tumor Jinak	Tumor Ganas	TP
	66	1.204988	Normal	Tumor Ganas	FN
	67	1.338678	Normal	Tumor Ganas	FN
	68	1.408522	Normal	Tumor Ganas	FN
	69	1.247841	Normal	Tumor Ganas	FN
	70	2.155377	Tumor Jinak	Tumor Ganas	TP
	71	2.098338	Tumor Jinak	Tumor Ganas	TP
	72	1.704269	Tumor Jinak	Tumor Ganas	TP
	73	1.372234	Normal	Tumor Ganas	FN
	74	1.242944	Normal	Tumor Ganas	FN
	75	1.3105	Normal	Tumor Ganas	FN
<i>Data Testing</i>	1	2.11087	Tumor Jinak	Normal	FP
	2	1.641312	Tumor Jinak	Normal	FP
	3	1.655695	Tumor Jinak	Normal	FP
	4	1.866991	Tumor Jinak	Normal	FP
	5	1.543055	Tumor Jinak	Normal	FP
	6	2.386479	Tumor Jinak	Normal	FP

	7	2.249478	Tumor Jinak	Normal	FP
	8	2.378762	Tumor Jinak	Normal	FP
	9	2.579985	Tumor Ganas	Normal	FP
	10	2.150691	Tumor Jinak	Normal	FP
	11	2.799208	Tumor Ganas	Tumor Jinak	TP
	12	2.624814	Tumor Ganas	Tumor Jinak	TP
	13	1.457795	Normal	Tumor Jinak	FN
	14	1.212688	Normal	Tumor Jinak	FN
	15	1.275463	Normal	Tumor Jinak	FN
	16	2.764034	Tumor Ganas	Tumor Jinak	TP
	17	2.678997	Tumor Ganas	Tumor Jinak	TP
	18	2.569123	Tumor Ganas	Tumor Ganas	TP
	19	1.22495	Normal	Tumor Ganas	FN
	20	1.292387	Normal	Tumor Ganas	FN
	21	1.408194	Normal	Tumor Ganas	FN
	22	1.18393	Normal	Tumor Ganas	FN
	23	1.203696	Normal	Tumor Ganas	FN
	24	1.184444	Normal	Tumor Ganas	FN
	25	1.967126	Tumor Jinak	Tumor Ganas	TP

**Lampiran 3.20** *Performance Measure Model RNN*

	Data ke-	<i>Output</i>	Status <i>Output</i>	Target	<i>Performance Measure</i>
Data Training	1	1.764176	Tumor Jinak	Normal	FP
	2	1.373493	Normal	Normal	TN
	3	1.347956	Normal	Normal	TN
	4	1.255733	Normal	Normal	TN
	5	1.630424	Tumor Jinak	Normal	FP
	6	1.424067	Normal	Normal	TN
	7	1.879436	Tumor Jinak	Normal	FP
	8	1.397762	Normal	Normal	TN
	9	1.786667	Tumor Jinak	Normal	FP
	10	1.61933	Tumor Jinak	Normal	FP
	11	1.633657	Tumor Jinak	Normal	FP
	12	1.319779	Normal	Normal	TN
	13	1.173485	Normal	Normal	TN
	14	1.307107	Normal	Normal	TN
	15	2.567016	Tumor Ganas	Normal	FP
	16	1.516209	Tumor Jinak	Normal	FP
	17	1.186779	Normal	Normal	TN
	18	1.301543	Normal	Normal	TN

19	2.244008	Tumor Jinak	Normal	FP
20	1.378377	Normal	Normal	TN
21	1.831139	Tumor Jinak	Normal	FP
22	1.179455	Normal	Normal	TN
23	1.264216	Normal	Normal	TN
24	1.923402	Tumor Jinak	Normal	FP
25	1.339023	Normal	Normal	TN
26	1.639907	Tumor Jinak	Tumor Jinak	TP
27	1.739304	Tumor Jinak	Tumor Jinak	TP
28	1.863881	Tumor Jinak	Tumor Jinak	TP
29	1.477432	Normal	Tumor Jinak	FN
30	2.01418	Tumor Jinak	Tumor Jinak	TP
31	2.07194	Tumor Jinak	Tumor Jinak	TP
32	1.787518	Tumor Jinak	Tumor Jinak	TP
33	1.702829	Tumor Jinak	Tumor Jinak	TP
34	2.008601	Tumor Jinak	Tumor Jinak	TP
35	2.281066	Tumor Jinak	Tumor Jinak	TP
36	1.951124	Tumor Jinak	Tumor Jinak	TP
37	2.494253	Tumor Jinak	Tumor Jinak	TP
38	1.686761	Tumor Jinak	Tumor Jinak	TP
39	1.929855	Tumor Jinak	Tumor Jinak	TP
40	2.089271	Tumor Jinak	Tumor Jinak	TP
41	2.006021	Tumor Jinak	Tumor Jinak	TP
42	1.514118	Tumor Jinak	Tumor Jinak	TP
43	2.007007	Tumor Jinak	Tumor Jinak	TP
44	2.306495	Tumor Jinak	Tumor Jinak	TP
45	2.794456	Tumor Ganas	Tumor Jinak	TP
46	2.237979	Tumor Jinak	Tumor Jinak	TP
47	1.958123	Tumor Jinak	Tumor Jinak	TP
48	1.177433	Normal	Tumor Jinak	FN
49	2.163999	Tumor Jinak	Tumor Jinak	TP
50	2.640209	Tumor Ganas	Tumor Jinak	TP
51	2.059999	Tumor Jinak	Tumor Jinak	TP
52	2.598979	Tumor Ganas	Tumor Ganas	TP
53	2.799473	Tumor Ganas	Tumor Ganas	TP
54	2.751046	Tumor Ganas	Tumor Ganas	TP
55	2.695662	Tumor Ganas	Tumor Ganas	TP
56	2.287344	Tumor Jinak	Tumor Ganas	TP
57	2.799608	Tumor Ganas	Tumor Ganas	TP
58	2.795999	Tumor Ganas	Tumor Ganas	TP
59	2.636022	Tumor Ganas	Tumor Ganas	TP
60	2.779134	Tumor Ganas	Tumor Ganas	TP

	61	2.53623	Tumor Ganas	Tumor Ganas	TP
	62	2.591535	Tumor Ganas	Tumor Ganas	TP
	63	2.018173	Tumor Jinak	Tumor Ganas	TP
	64	1.72349	Tumor Jinak	Tumor Ganas	TP
	65	1.355663	Normal	Tumor Ganas	FN
	66	2.56173	Tumor Ganas	Tumor Ganas	TP
	67	1.911739	Tumor Jinak	Tumor Ganas	TP
	68	2.610676	Tumor Ganas	Tumor Ganas	TP
	69	2.572816	Tumor Ganas	Tumor Ganas	TP
	70	2.575643	Tumor Ganas	Tumor Ganas	TP
	71	2.675452	Tumor Ganas	Tumor Ganas	TP
	72	2.444324	Tumor Jinak	Tumor Ganas	TP
	73	1.974414	Tumor Jinak	Tumor Ganas	TP
	74	2.796221	Tumor Ganas	Tumor Ganas	TP
	75	1.753981	Tumor Jinak	Tumor Ganas	TP
<i>Data Testing</i>	1	1.436922	Normal	Normal	TN
	2	2.14146	Tumor Jinak	Normal	FP
	3	2.795365	Tumor Ganas	Normal	FP
	4	2.743995	Tumor Ganas	Normal	FP
	5	1.298676	Normal	Normal	TN
	6	2.479233	Tumor Jinak	Normal	FP
	7	1.332294	Normal	Normal	TN
	8	1.339793	Normal	Normal	TN
	9	1.57461	Tumor Jinak	Normal	FP
	10	2.003934	Tumor Jinak	Normal	FP
	11	1.430666	Normal	Tumor Jinak	FN
	12	1.316384	Normal	Tumor Jinak	FN
	13	2.180216	Tumor Jinak	Tumor Jinak	TP
	14	2.797195	Tumor Ganas	Tumor Jinak	TP
	15	2.794938	Tumor Ganas	Tumor Jinak	TP
	16	1.70496	Tumor Jinak	Tumor Jinak	TP
	17	1.476079	Normal	Tumor Jinak	FN
	18	1.80941	Tumor Jinak	Tumor Ganas	TP
	19	2.76288	Tumor Ganas	Tumor Ganas	TP
	20	2.430366	Tumor Jinak	Tumor Ganas	TP
	21	1.581869	Tumor Jinak	Tumor Ganas	TP
	22	2.794475	Tumor Ganas	Tumor Ganas	TP
	23	2.747536	Tumor Ganas	Tumor Ganas	TP
	24	2.725619	Tumor Ganas	Tumor Ganas	TP
	25	2.299485	Tumor Jinak	Tumor Ganas	TP



### Lampiran 3.21 M-File Proses RNN dan RNF

#### 1. Pembelajaran RNN (mencari banyak neuron lapisan tersembunyi)

```
%mencari banyak neuron lapisan tersembunyi
%yang memberikan MSE kecil
syms y t;
jawab = 'y';
while jawab=='y';
n11= input ('berapa neuron pada lapisan tersembunyi= ');
%jumlah neuron lapisan tersembunyi diubah-ubah hingga diperoleh
%neuron yang MSE-nya kecil
X1=[];% X1 adalah matriks input-target training, 12 Fitur
X2=[];% X2 adalah matriks input-target testing, 12 Fitur
P=[X1(:,1:12)']; %input training
Pc=[X2(:,1:12)']; %input testing
[m, n]=size(P);
[mc, nc]=size(Pc);
T=X1(:,13)';%target training
Tc=X2(:,13)';%target testing
%normalisasi input dan target training dan testing
[Pn,meanp, stdp, Tn,meant, stdt]=prestd(P,T);
[Pcn,meanpc, stdpc, Tcn,meantc, stdtc]=prestd(Pc,Tc);
%inisialisasi RNN
net=newelm(minmax(Pn),[n11 1],{'logsig' 'tansig'}, 'traingdx');
BobotAwal_Input=net.IW{1,1};
BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.06;
net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;
%melakukan pebelajaran
net=train(net,Pn,Tn);
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%menampilkan target,output jaringan dan error data training
ab = sim(net,Pn);
a=poststd(ab,meant, stdt);
e1 = T-a;
MSE1 = mse(e1);
H1=[(1:size(P,2))' T' a' (T'-a')];
```

```

d1=['MSE training adalah ',num2str(MSE1)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H1');
%menampilkan target, output jaringan dan error data testing
Qn=trastd(Pc,meanp,stdp);
bn=sim(net,Qn);
b=poststd(bn,meant,stdt);
E2=Tc-b;
MSE2=mse(E2);
H2=(1:size(Pc,2))' Tc' b' (Tc'-b)];
d2=['MSE testing adalah ',num2str(MSE2)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H2');
%display MSE
disp(d1);
disp(d2);
jawab=input('Ingin mengulang lagi?(y/t)');
end

```

## 2. Pembelajaran II (mencari input optimal dengan 10 neuron pada lapisan tersembunyi)

```

%input diubah-ubah hingga diperoleh
%neuron yang MSE-nya kecil
X1=[];% X1 adalah matriks input-target training, 14 Fitur
X2=[];% X2 adalah matriks input-target testing, 14 Fitur
P=[X1(:,1:3)';X1(:,5)';X1(:,7)';X1(:,9)';X1(:,12:13)'];
%input training disesuaikan
Pc=[X2(:,1:3)';X2(:,5)';X2(:,7)';X2(:,9)';X2(:,12:13)'];
%input testing disesuaikan
[m, n]=size(P);
[mc, nc]=size(Pc);
T=X1(:,15)';%target training
Tc=X2(:,15)';%target testing
%normalisasi input dan target training dan testing
[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);
[Pcn,meanpc,stdpc,Tcn,meantc,stdtc]=prestd(Pc,Tc);
%inisialisasi RNN
n11=10; %banyak neuron lap tersembunyi hasil pembelajaran RNN I
net=newelm(minmax(Pn),[n11 1],{'logsig' 'tansig'}, 'traingdx');
BobotAwal_Input=net.IW{1,1};
BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.06;
net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;

```

```

%melakukan pebelajaran
net=train(net,Pn,Tn);
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%menampilkan target, output jaringan dan error data training
ab = sim(net,Pn);
a=poststd(ab,meant,stdt);
e1 = T-a;
MSE1 = mse(e1);
H1=[(1:size(P,2))' T' a' (T'-a')];
d1=['MSE training adalah ',num2str(MSE1)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H1');
%menampilkan target, output jaringan dan error data testing
Qn=trastd(Pc,meanp,stdp);
bn=sim(net,Qn);
b=poststd(bn,meant,stdt);
E2=Tc-b;
MSE2=mse(E2);
H2=[(1:size(Pc,2))' Tc' b' (Tc'-b')];
d2=['MSE testing adalah ',num2str(MSE2)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H2');
%display MSE
disp(d1);
disp(d2);
%export hasil ke ms.excel
xlswrite('OutputRNNtrain.xlsx',[H1]);
xlswrite('OutputRNNtest.xlsx',[H2]);

```

### 3. Clustering

```

%PROSES CLUSTERING
%input dan target hasil pemodelan RNN
X1=[];% X1 adalah matriks input-target training, 8 Fitur
X2=[];% X2 adalah matriks input-target testing, 8 Fitur
P=[X1(:,1:8)']; %input training
Pc=[X2(:,1:8)']; %input testing
[m, n]=size(P);
[mc, nc]=size(Pc);
T=X1(:,9)';%target training
Tc=X2(:,9)';%target testing
n11=10;
%normalisasi input dan target training dan testing
[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);
[Pcn,meanpc,stdpc,Tcn,meantc,stdtc]=prestd(Pc,Tc);
%inisialisasi RNN
net=newelm(minmax(Pn),[n11 1],{'logsig' 'tansig'}, 'traingdx');
BobotAwal_Input=net.IW{1,1};

```

```

BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.06;
net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;
%melakukan pebelajaran
net=train(net,Pn,Tn);
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%menampilkan target, output jaringan dan error data training
ab = sim(net,Pn);
a=poststd(ab,meant,stdt);
e1 = T-a;
MSE1 = mse(e1);
H1=[(1:size(P,2))' T' a' (T'-a')];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H1');
%menampilkan target, output jaringan dan error data testing
Qn=trastd(Pc,meanp,stdp);
bn=sim(net,Qn);
b=poststd(bn,meant,stdt);
E2=Tc-b;
MSE2=mse(E2);
H2=[(1:size(Pc,2))' Tc' b' (Tc'-b')];
%melakukan simulasi
PPn = [Pn Pcn];
TTn = [Tn Tcn];
y_1 = sim(net,PPn);
SSE = sum((y_1-TTn).^2);
[m0,b,r] = postreg(y_1(1,:),TTn);
%clustering dengan FCM
X3=[P; T]; %training yang akan dicluster
X4=[Pc; Tc]; %testing yang akan dicluster
X3=X3';
X4=X4';
C=3; %ditentukan 3 cluster
[V,U,obj_fcn]=fcm(X3,C)
[Vc,Uc,obj_fcn_c]=fcm(X4,C)
[DT, II]=max(U)
[DTc, IIc]=max(Uc)

```

```

%export hasil ke ms.excel
xlswrite('V.xlsx', [V]);
%V pusat cluster data training
xlswrite('U.xlsx', [U]);
%U matriks pasrtisi fuzzy akhir
%atau matriks fungsi keanggotaan
%atau m_is(t)training
xlswrite('fcn.xlsx', [obj_fcn]);
xlswrite('Vc.xlsx', [Vc]);
%Vc pusat cluster data testing
xlswrite('Uc.xlsx', [Uc]);
%Uc matriks pasrtisi fuzzy akhir
%atau matriks fungsi keanggotaan
%atau m_is(t)testing
xlswrite('fcnc.xlsx', [obj_fcn_c]);
xlswrite('DT.xlsx', [DT]);
%nilai maks dari m_is(t)training
xlswrite('II.xlsx', [II]);
%pembagian training ke dalam 3 cluster
xlswrite('DTc.xlsx', [DTc]);
%nilai maks dari m_is(t)testing
xlswrite('IIc.xlsx', [IIc]);
%pembagian testing ke dalam 3 cluster
for k=1:C,
for j=1:n,
if II(j)==k,
TA(j,k)=1;
else
TA(j,k)=0;
end;
end;
end;
for k=1:C,
for j=1:nc,
if IIc(j)==k,
TAc(j,k)=1;
else
TAc(j,k)=0;
end;
end;
end;
%menghitung nilai keanggotaan tiap data pada bagian anteseden
%inisialisasi RNN
net=newelm(minmax(Pn), [n11 C], {'logsig' 'logsig' 'traingdx'});
BobotAwal_Input=net.IW{1,1};
BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;

```

```

net.trainParam.max_perf_inc=1.06;
net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;
%melakukan pembelajaran
net=train(net,Pn, 'TA');
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%melakukan simulasi
muA=sim(net,Pn);%nilai keanggotaan tarining data pada bagian
anteseden
muAc=sim(net,Pcn);%nilai keanggotaan TESTING pada bagian anteseden
%export hasil ke ms.excel
xlswrite('muA.xlsx',[muA]);
xlswrite('muAc.xlsx',[muAc]);

```

#### 4. Pembelajaran RNNs (mencari neuron terbaik masing-masing cluster)

```

%neuron terbaik
syms y t;
jawab ='y';
while jawab=='y';
n11= input ('berapa neuron pada lapisan tersembunyi= ');
%jumlah neuron lapisan tersembunyi diubah-ubah hingga diperoleh
%neuron yang MSE-nya kecil
X1=[];% X1 adalah matriks input-target training, cluster ke-s
X2=[];% X2 adalah matriks input-target testing, cluster ke-s
P=[X1(:,1:8)']; %input training
Pc=[X2(:,1:8)']; %input testing
[m, n]=size(P);
[mc, nc]=size(Pc);
T=X1(:,9)';%target training
Tc=X2(:,9)';%target testing
%normalisasi input dan target training dan testing
[Pn,meanp, stdp, Tn,meant, stdt]=prestd(P,T);
[Pcn,meanpc, stdpc, Tcn,meantc, stdtc]=prestd(Pc,Tc);
%inisialisasi RNN
net=newelm(minmax(Pn),[n11 1],{'logsig' 'tansig'}, 'traingdx');
BobotAwal_Input=net.IW{1,1};
BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.06;

```

```

net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;
%melakukan pebelajaran
net=train(net,Pn,Tn);
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%menampilkan target, output jaringan dan error data training
ab = sim(net,Pn);
a=poststd(ab,meant,stdt);
e1 = T-a;
MSE1 = mse(e1);
H1=[(1:size(P,2))' T' a' (T'-a')];
d1=['MSE training adalah ',num2str(MSE1)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H1');
%menampilkan target, output jaringan dan error data testing
Qn=trastd(Pc,meanp,stdp);
bn=sim(net,Qn);
b=poststd(bn,meant,stdt);
E2=Tc-b;
MSE2=mse(E2);
H2=[(1:size(Pc,2))' Tc' b' (Tc'-b')];
d2=['MSE testing adalah ',num2str(MSE2)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H2');
%display MSE MAPE
disp(d1);
disp(d2);
jawab=input('Ingin mengulang lagi?(y/t)');
end

```

## 5. Penyederhanaan RNNs (mencari input optimal masing-masing cluster)

```

%input diubah-ubah hingga diperoleh
%neuron yang MSE-nya kecil
X1=[];% X1 adalah matriks input-target training, cluster ke-s
X2=[];% X2 adalah matriks input-target testing, cluster ke-s
P=[X1(:,1:9)']; %input training
P=[X1(:,2:7)'];
Pc=[X2(:,2:7)']; %input testing
[m, n]=size(P);
[mc, nc]=size(Pc);
T=X1(:,9)';%target training
Tc=X2(:,9)';%target testing
%normalisasi input dan target training dan testing
[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);

```

```

[Pcn,meanpc, stdpc, Tcn,meantc, stdtc]=prestd(Pc, Tc);
%inisialisasi RNN
n11=6; %banyak neuron lap tersembunyi hasil pembelajaran RNNs I
net=newelm(minmax(Pn), [n11 1], {'logsig' 'tansig'}, 'traingdx');
BobotAwal_Input=net.IW{1,1};
BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.06;
net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;
%melakukan pebelajaran
net=train(net, Pn, Tn);
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%menampilkan target, output jaringan dan error data training
ab = sim(net, Pn);
a=poststd(ab, meant, stdt);
e1 = T-a;
MSE1 = mse(e1);
H1=[(1:size(P,2))' T' a' (T'-a')];
d1=['MSE training adalah ', num2str(MSE1)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H1');
%menampilkan target, output jaringan dan error data testing
Qn=trastd(Pc, meanp, stdp);
bn=sim(net, Qn);
b=poststd(bn, meant, stdt);
E2=Tc-b;
MSE2=mse(E2);
H2=[(1:size(Pc,2))' Tc' b' (Tc'-b')];
d2=['MSE testing adalah ', num2str(MSE2)];
%dibulantakn 2 angka di belakang koma
sprintf('%2d %9.2f %7.2f %5.2f\n' ,H2');
%display MSE MAPE
disp(d1);
disp(d2);

```



## 6. Penyederhanaan bagian konsekuen (THEN) atau mencari parameter dengan LSE

```
function T=LSE(A,y);
[n m]=size(A);
n1=m;
n2=n-n1;
A1=A(1:n1,:);
y1=y(1:n1,:);
A2=A(n1+1:end,:);
y2=y(n1+1:end,:);
P=inv(A1'*A1);
T=P*A1'*y1;
for i=1:n2,
    P=P-(P*A2(i,:)'*A2(i,:)*P)/(1+A2(i,:)*P*A2(i,:)');
    T=T+P*A2(i,:)'*(y2(i,:)-A2(i,:)*T);
end;
D=A*T;
k=1:n;

%cluster s
E1=[];%E1 adalah training cluster s
%setelah proses penyederhanaan RNNs
E2=[];%E1 adalah training cluster s
%setelah proses penyederhanaan RNNs
R1=LSE(E1(:,1:8),E1(:,9));
yr1=E1(:,1:8)*R1;
E=E1(:,9)-yr1;
e=mean(E);
R=[R1' e];
ysed=R1'*E1(:,1:8)'+e;
ysed2=R1'*E2(:,1:8)'+e;
%export hasil ke ms.excel
xlswrite('TrainSeds.xlsx',[ysed]);
xlswrite('TestSeds.xlsx',[ysed2]);
%export hasil ke ms.excel
xlswrite('KoefRs.xlsx',[R]);
```

## 7. Pembelajaran setiap RNNs bagian konsekuen (THEN)

```
X1=[];% matriks input-output training awal, hasil pemodelan RNN
X2=[];%matriks input-output testing awal, hasil pemodelan RNN
P=[X1(:,1:8)'];%input training
Pc=[X2(:,1:8)'];%input testing
[m,n]=size(P);
[mc,nc]=size(Pc);
T=X1(:,9)';%target training
Tc=X2(:,9)';%target testing
[Pn,meanp, stdp, Tn,meant, stdt]=prestd(P,T);
[Pcn,meanpc, stdpc, Tcn,meantc, stdtc]=prestd(Pc,Tc);
E1=[];%matriks training setiap kelas hasil penyederhanaan
E2=[];%matriks testing setiap kelas hasil penyederhanaan
P1=[E1(:,1:8)']; % input training RNNs
[m1,n1]=size(P1);
```

```

Pc1=E2(:,1:8)'; %input testing RNNs
[mc1,nc1]=size(Pc1);
T1=E1(:,9)';%target training RNNs
Tc1=E2(:,9)';%target testing RNNs
%normalisasi training dan testing
[p1n,meanp1,stdp1,t1n,meant1,stdt1]=prestd(P,T);
[pc1n,meanpc1,stdpc1,tc1n,meantc1,stdtc1]=prestd(Pc,Tc);
[p2n,meanp2,stdp2,t2n,meant2,stdt2]=prestd(P1,T1);
[pc2n,meanpc2,stdpc2,tc2n,meantc2,stdtc2]=prestd(Pc1,Tc1);
%inisialisasi
n11=10;
net=newelm(minmax(Pn),[n11 1],{'logsig' 'tansig' 'traingdx'});
BobotAwal_Input=net.IW{1,1};
BobotAwal_Bias_Input=net.b{1,1};
BobotAwal_Delay=net.LW{1,1};
BobotAwal_Lapisan=net.LW{2,1};
BobotAwal_Bias_Lapisan=net.b{2,1};
%set parameter
net.trainParam.epochs=5000;
net.trainParam.goal=1e-5;
net.trainParam.max_perf_inc=1.06;
net.trainParam.lr=0.1;
net.trainParam.lr_inc=1.2;
net.trainParam.lr_dec=0.6;
net.trainParam.mc=0.8;
net.trainParam.show=500;
%melakukan pembelajaran
net=train(net,p1n,t1n);
%melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input=net.IW{1,1};
BobotAkhir_Bias_Input=net.b{1,1};
BobotAkhir_Delay=net.LW{1,1};
BobotAkhir_Lapisan=net.LW{2,1};
BobotAkhir_Bias_Lapisan=net.b{2,1};
%melakukan simulasi
muS01=sim(net,p1n);
muS1=sim(net,pc1n);
muS0a=poststd(muS01,meant1,stdt1);
muSa=poststd(muS1,meant1,stdt1);
muAS=[];
muAA=[];
muAS=[muAS;muSa];
muAA=[muAA;muS0a];
%export hasil ke ms.excel
xlswrite('fAS1train.xlsx',[muAS]);
xlswrite('fAA1test.xlsx',[muAA]);

```

## 8. Penentuan Output Akhir

```

X1=[];% matriks input-output training awal
X2=[];%matriks input-output testing awal
P=[X1(:,1:8)'];%input training
Pc=[X2(:,1:8)'];%input testing

```

```

[m,n]=size(P);
[mc,nc]=size(Pc);
T=X1(:,9)';%target training
Tc=X2(:,9)';%target testing
muA=[];%matriks training hasil pembelajaran jaringan syaraf bagian
IF
muAc=[];%matriks testing hasil pembelajaran jaringan syaraf bagian
IF
muAA=[];%matriks training hasil pembelajaran jaringan syaraf
bagian THEN
muAS=[];%matriks testing hasil pembelajaran jaringan syaraf bagian
THEN
muA=muA';
muAc=muAc';
muAA=muAA';
muAS=muAS';
%menghitung output jaringan untuk TRAINING
for i=1:n,
yt0(i)=0;
st0(i)=0;
yt0(i)=yt0(i)+muA(i)*muAA(i);
st0(i)=st0(i)+muA(i);
yt0(i)=yt0(i)/st0(i)
end;
%menghitung output jaringan untuk TESTING
for i=1:nc,
yt1(i)=0;
st1(i)=0;
yt1(i)=yt1(i)+muAc(i)*muAS(i);
st1(i)=st1(i)+muAc(i);
yt1(i)=yt1(i)/st1(i)
end;
E=T-yt0;
mse_TRAINING=mse(E)
mapel=[abs((E./T).*100)];
MAPE1=sum(mapel)/n
Ec=Tc-yt1;
mse_TESTING=mse(Ec)
mape2=[abs((Ec./Tc).*100)];
MAPE2=sum(mape2)/nc
%export hasil ke ms.excel
xlswrite('Outputtraining.xlsx',[yt0]);
xlswrite('Outputtesting.xlsx',[yt1]);

```

### Lampiran 3.22 Hitungan Manual *Output* RNN

Berdasarkan pelatihan menggunakan struktur jaringan Elman RNN dengan algoritma *backpropagation* dengan menggunakan program MatlabR2010a, diperoleh bobot bobot sebagai berikut:

$$b_k =$$

[3.7563; 3.0789; -3.755; 1.8254; 1.8270; -1.062; 0.5799; -2.692; 5.9570; -2.211]

$$b_0 = 0.118248$$

$$W_{ik(x)} =$$

-0.48027	0.755582	-0.1287	0.057204	0.501807	0.034096	-0.43936	0.771386
-0.31745	0.552705	-0.53015	0.224093	-0.10786	0.655946	-0.38709	-0.72416
0.245141	0.639334	0.177901	-0.18525	-0.79681	0.28141	0.325359	-0.07534
-0.37525	-0.36762	-0.04009	0.489976	-0.34726	0.714213	0.509818	-0.1159
0.553747	0.154799	0.288503	0.357038	-0.54994	-0.43205	-0.24586	-0.06394
0.650347	-0.6183	0.232291	0.604826	-0.27446	0.231632	0.349992	0.348834
0.399871	-0.12474	0.207498	0.052122	-0.09678	-0.35757	0.281954	-0.29545
-0.23578	-0.27181	-0.60749	-0.25293	0.038128	0.253259	-0.68981	0.41208
0.147367	-0.57026	-0.6519	-0.66214	-0.06942	0.333882	0.165837	-0.04811
-0.57152	-0.44983	-0.22671	0.154857	0.508742	-0.61442	-0.15277	-0.64822]

$$W_{kk(u)} =$$

-2.2216	-1.5828	1.01179	-0.282	0.56374	0.30649	-0.6540	1.34524	-1.043	1.61991
1.18990	1.42461	0.96056	0.86896	0.21859	0.79044	-0.2770	-2.1794	-1.8780	-0.5649
-0.1966	-2.3088	1.00696	2.00437	2.74339	0.32544	-0.9950	0.18839	0.63843	1.3601
-1.9665	-1.2033	2.52093	-0.8481	-1.3320	1.25174	1.49217	0.17183	-1.3469	1.15548
-0.9606	-2.4724	-1.7600	0.9796	-1.1000	0.13602	0.77336	2.18374	-2.754	-0.3488
0.51253	0.36372	0.99883	-0.4001	-1.8173	2.35630	1.29808	-0.0722	1.2167	-2.2929
-1.8962	1.12797	-1.622	2.10340	2.70564	-1.7305	2.66273	-0.6554	-1.5822	-1.0404
1.27542	0.24924	-2.0359	1.78088	0.77860	-2.1087	2.52885	0.91704	-0.3081	-0.4048
-1.5966	-0.4611	0.66625	-1.5122	-0.1275	-2.4234	-1.9122	1.49798	1.16603	-1.426
2.1544	0.74549	-0.2573	0.58559	0.71904	-2.2523	-1.8638	0.10349	-0.7265	-1.5635]

$v =$

[1.1537; -0.2513; 1.39060; -0.3902; 0.9650; -0.370; 1.10637; 0.91474; -0.4396; -1.018]

Operasi keluaran dari lapisan *input* ke lapisan tersembunyi (data *testing* pertama) adalah sebagai berikut (Persamaan 3.1):

$$\begin{aligned} y_{in(1)}_1 &= X_1w_{11(x)} + X_2w_{21(x)} + X_3w_{31(x)} + X_5w_{41(x)} + X_7w_{51(x)} + X_9w_{61(x)} + \\ &\quad X_{12}w_{71(x)} + X_{13}w_{81(x)} + b_1 \\ &= (-1.06115)(-0.48027) + (0.900755)(0.755582) + (-0.21982)(-0.1287) \\ &\quad + (-0.89944)(0.057204) + (-1.29305)(0.501807) + (0.995438)(0.034096) \\ &\quad + (-1.64401)(-0.43936) + (-0.89916)(0.771386) + (3.756371) \\ &= 4.337235 \end{aligned}$$

Sinyal terbobot  $y_{in(1)}_1$  kemudian diaktivasi dengan *logsig* (Persamaan 3.8), yaitu sebagai berikut:

$$y(1)_1 = f(y_{in(1)}_1) = \frac{1}{1 + e^{-(4.337235)}} = 0.987096$$

Tabel A berikut adalah perhitungan untuk  $k = 1, 2, \dots, 10$  dengan bantuan MS Excel.

**Tabel A** Hitungan Manual  $y(1)_k$

$k$	$y_{in(1)}_k$	$y(1)_k$
2	6.671022	0.998735
3	4.915414	0.992721
4	4.249478	0.985929
5	4.050908	0.982891
6	2.205736	0.900763
7	2.791012	0.942188
8	4.728066	0.991234
9	3.279071	0.963704
10	3.522216	0.971313

Operasi keluaran dari lapisan tersembunyi ke- $k$  (dengan neuron tambahan konteks) menuju ke lapisan *output* yang dihitung dengan Persamaan 3.6 adalah sebagai berikut:

$$\begin{aligned}
y_{in_1} = & X_1w_{11(x)} + X_2w_{21(x)} + X_3w_{31(x)} + X_5w_{41(x)} + X_7w_{51(x)} + X_9w_{61(x)} \\
& + X_{12}w_{71(x)} + X_{13}w_{81(x)} + b_1 + u_1w_{11(u)} + u_2w_{21(u)} + u_3w_{31(u)} \\
& + u_4w_{41(u)} + u_5w_{51(u)} + u_6w_{61(u)} + u_7w_{71(u)} + u_8w_{81(u)} + u_9w_{91(u)} \\
& + u_{10}w_{10_1(u)} + u_{11}w_{11_1(u)}
\end{aligned}$$

yang dapat disederhanakan menjadi

$$\begin{aligned}
y_{in_1} = & y_{in(1)_1} + u_1w_{11(u)} + u_2w_{21(u)} + u_3w_{31(u)} + u_4w_{41(u)} + u_5w_{51(u)} \\
& + u_6w_{61(u)} + u_7w_{71(u)} + u_8w_{81(u)} + u_9w_{91(u)} + u_{10}w_{10_1(u)} + u_{11}w_{11_1(u)} \\
= & 0,268618 + (0.987096)(-2.22161) + (0.998735)(-1.58289) + \\
& (0.992721)(1.011795) + (0.985929)(-0.2829) + \\
& (0.982891)(0.563749) + (0.900763)(0.306491) + \\
& (0.942188)(-0.65402) + (0.991234)(1.345241) + \\
& (0.963704)(-1.0438) + (0.971313)(1.619913) \\
= & 3,403863
\end{aligned}$$

Sinyal terbobot  $y_{in_1}$  kemudian diaktivasi dengan *logsig* (Persamaan 3.8), yaitu sebagai berikut:

$$y_1 = f(y_{in_1}) = \frac{1}{1 + e^{-(3,403863)}} = -0,93565$$

Tabel B menunjukkan perhitungan untuk  $k = 1,2, \dots, 10$  dengan bantuan Ms. Excel :

**Tabel B** Hitungan Manual  $y_k$

$k$	$y_{in_k}$	$y_k = f(y_{in_k})$
2	7.225536	-0.99855
3	9.566466	-0.99986
4	3.991619	-0.96373
5	-1.20701	0.539539
6	4.104963	-0.96755
7	2.932943	-0.8989
8	7.437926	-0.99882
9	-2.47397	0.844593
10	1.520731	-0.64129

Selanjutnya, berdasarkan persamaan (3.10) diperoleh operasi pada lapisan *output*:

$$\begin{aligned}
 z_{in} &= \sum_{k=1}^{10} v_k y_k + b_0 \\
 &= v_1 y_1 + v_2 y_2 + v_3 y_3 + v_4 y_4 + v_5 y_5 + v_6 y_6 + v_7 y_7 + v_8 y_8 + v_9 y_9 + \\
 &\quad v_{10} y_{10} + b_0 \\
 &= (1.15374)(-0,93565) + (-0.25131)(-0.998546) + \\
 &\quad (1.390605)(-0.99986) + (-0.39023)(-0.96373) + (0.965085)(0.53954) \\
 &\quad + (-0.37012)(-0.967554) + (1.106378)(-0.898902) \\
 &\quad + (0.914745)(-0.998824) + (-0.43968)(0.84459) \\
 &\quad + (-1.0184)(-0.641292) + 0.118248 \\
 &= -5.39
 \end{aligned}$$

Sinyal terbobot  $z_{in}$  kemudian diaktivasi dengan *tansig* (Persamaan 3.12), yaitu sebagai berikut:

$$z = f(z_{in}) = \frac{1 - e^{5.39}}{1 + e^{5.39}} = -0,99092 \approx -1$$

Diperoleh nilai  $z = -1$ . Nilai tersebut masih dalam bentuk normal. Untuk mengembalikan dalam bentuk data semula, digunakan fungsi *poststd* pada MatlabR2010a. *Output* klasifikasi data *testing* pertama adalah 2,706. Hasil klasifikasi untuk 25 data *testing* dalam bentuk data normalisasi ditunjukkan pada Tabel C berikut:

**Tabel C** Nilai Normalisasi dan Denormalisasi Data *Testing*

Data ke-	Nilai Normal	Nilai Denormalisasi
2	-1.06708	2.14146
3	-1.06708	2.795365
4	-1.06708	2.743995
5	-1.06708	1.298676
6	-1.06708	2.479233
7	-1.06708	1.332294

8	-1.06708	1.339793
9	-1.06708	1.57461
10	-1.06708	2.003934
11	0.092789	1.430666
12	0.092789	1.316384
13	0.092789	2.180216
14	0.092789	2.797195
15	0.092789	2.794938
16	0.092789	1.70496
17	0.092789	1.476079
18	1.252656	1.80941
19	1.252656	2.76288
20	1.252656	2.430366
21	1.252656	1.581869
22	1.252656	2.794475
23	1.252656	2.747536
24	1.252656	2.725619
25	1.252656	2.299485



**Lampiran 3.23 Percobaan Lainnya**

1. Hasil Percobaan dengan menggunakan Citra Abu-abu

Percobaan		Akurasi	Sensitivitas	Spesifisitas
14 Fitur	<i>Training</i>	78,67%	96%	44%
	<i>Testing</i>	64%	100%	10%
12 Fitur Percobaan 1	<i>Training</i>	86%	100%	93,3%
	<i>Testing</i>	40%	40%	60%
12 Fitur Percobaan 2	<i>Training</i>	90,67%	94%	94,6%
	<i>Testing</i>	40%	66,7%	40%
12 Fitur Percobaan 3	<i>Training</i>	90,9%	96%	97,33%
	<i>Testing</i>	40%	66,7%	40%
8 Fitur	<i>Training</i>	90,67%	92%	88%
	<i>Testing</i>	46%	66,7%	40%

Keterangan: percobaan dengan menggunakan citra abu-abu memberikan akurasi data *training* tinggi tetapi akurasi untuk data *testing* terlalu rendah

2. Nilai MSE Percobaan dengan Fungsi Aktivasi dan Ukuran Pembagian Data

a. Ukuran Pembagian Data 80%-20% (neuron lapisan tersembunyi 4 dengan 12 fitur sebagai variable *input*)

No.	Fungsi Aktivasi 1	Fungsi Aktivasi 2	MSE <i>Training</i>	MSE <i>Testing</i>	Banyak data tidak cocok	
					<i>Training</i>	<i>Testing</i>
1.	Purelin	Tansig	0.5900	0.6485	57	13
2.	Logsig	Tansig	0.4663	0.7398	43	15
3.	Tansig	Purelin	0.4550*)	0.8283*)	40	11
2.	Tansig	Logsig	0.5897	0.7371	49	14

b. Ukuran Pembagian Data 75%-25% (neuron lapisan tersembunyi 4 dengan 12 fitur sebagai variable *input*)

No.	Fungsi Aktivasi 1	Fungsi Aktivasi 2	MSE <i>Training</i>	MSE <i>Testing</i>	Banyak data tidak cocok	
					<i>Training</i>	<i>Testing</i>
1.	Purelin	Tansig	0.5853	0.6749	49	17
<b>2.</b>	<b>Logsig</b>	<b>Tansig</b>	<b>0.3530</b>	<b>1.0616</b>	<b>28</b>	<b>18</b>
3.	Tansig	Purelin	0.3440*)	0.9156*)	26	17
2.	Tansig	Logsig	0.5660	0.7697	45	18

Keterangan :

Tansig-purelin memberikan nilai MSE paling sederhana tetapi tidak memberikan range *output* [1,3] baik dengan ukuran pembagaian data 80%-20% maupun 75%-25%, jadi dipilih fungsi aktivasi logsig-tansig dengan ukuran pembagaian data 75%-25% yang memberikan range *output* [1,3] .

3. Iterasi Lengkap Penentuan *Input* Optimal melalui Eliminasi *Input* (langkah pemodelan terbaik RNN)

Eliminasi	<i>Input</i>	MSE <i>Training</i>	MSE <i>Testing</i>
-	12 Fitur ( $X_1, X_2, X_3, X_4, X_5, X_6, X_7,$ $X_8, X_9, X_{11}, X_{12}, X_{13}$ )	0.34816	0.89514
<b>Iterasi 1</b>			
$X_1$	11 Fitur	0.35733	0.99861
$X_2$	11 Fitur	0.37624	0.99774
$X_3$	11 Fitur	0.37533	0.99714
$X_4$	11 Fitur	0.38743	1.10542
$X_5$	11 Fitur	0.35911	1.13996
$X_6$	11 Fitur	0.36171	1.13006
$X_7$	11 Fitur	0.37891	1.10176
$X_8$	11 Fitur ( $X_1, X_2, X_3, X_4, X_5, X_6, X_7,$ $X_9, X_{11}, X_{12}, X_{13}$ )	0.34973*)	0.99191*)
$X_9$	11 Fitur	0.35889	0.99923
$X_{11}$	11 Fitur	0.34653	1.11421
$X_{12}$	11 Fitur	0.34126	1.13031
$X_{13}$	11 Fitur	0.35013	1.12774
<b>Iterasi Kedua</b>			
$X_1$	10 Fitur	0.36118	1.12823
$X_2$	10 Fitur	0.35989	1.12809

$X_3$	10 Fitur	0.37063	1.10432
$X_4$	10 Fitur ( $X_1, X_2, X_3, X_5, X_6, X_7,$ $X_9, X_{11}, X_{12}, X_{13}$ )	0.35946*	0.92078*
$X_5$	10 Fitur	0.36604	0.95809
$X_6$	10 Fitur	0.36496	0.94986
$X_7$	10 Fitur	0.37615	1.20896
$X_9$	10 Fitur	0.36796	1.19806
$X_{11}$	10 Fitur	0.36234	1.18289
$X_{12}$	10 Fitur	0.37009	1.12078
$X_{13}$	10 Fitur	0.36907	0.92078
<b>Iterasi Ketiga</b>			
$X_1$	9 Fitur	0.37706	1.2004
$X_2$	9 Fitur	0.38609	1.1054
$X_3$	9 Fitur	0.37172	1.1901
$X_5$	9 Fitur	0.37101	1.1905
$X_6$	9 Fitur ( $X_1, X_2, X_3, X_5, X_7,$ $X_9, X_{11}, X_{12}, X_{13}$ )	0.36176*	1.1866*
$X_7$	9 Fitur	0.39014	1.2043
$X_9$	9 Fitur	0.36789	1.1995
$X_{11}$	9 Fitur	0.39177	1.1067
$X_{12}$	9 Fitur	0.39069	1.1939
$X_{13}$	9 Fitur	0.37003	1.2004

Iterasi Keempat			
$X_1$	8 Fitur	0.34451	0.80056
$X_2$	8 Fitur	0.35106	0.77376
$X_3$	8 Fitur	0.34892	0.76587
$X_5$	8 Fitur	0.35004	0.75011
$X_7$	8 Fitur	0.36443	0.78331
$X_9$	8 Fitur ( $X_1, X_2, X_3, X_5, X_7, X_{11}, X_{12}, X_{13}$ )	0.33742*)	0.74137*)
$X_{11}$	8 Fitur	0.35252	0.78985
$X_{12}$	8 Fitur	0.37861	0.8007
$X_{13}$	8 Fitur	0.37007	0.79087

4. Hasil Percobaan dengan Pemilihan *Cluster* dan Kategori

		Data Training			Data Testing		
		Sensi- tivitas	Spesi- fisitas	Akurasi	Sensi- tivitas	Spesi- fisitas	Akurasi
<b>3 Kategori dengan 3 cluster (Normal, Tumor Jinak, Tumor Ganas)</b>	RNN	94%*)	56%*)	81.3%*)	80%*)	40%*)	64%*)
	RNF	42%*)	68%*)	50,6%*)	40%*)	0%*)	24%*)
<b>3 Kategori dengan 2 cluster (Normal, Tumor Jinak, Tumor Ganas)</b>	RNN	94%	56%	78%	73,33%	30%	56%
	RNF	41%	44%	65,33%	35%	0%	44%
<b>2 Kategori (Normal dan Tumor)</b>	RNN	90%	52%	77,3%	86,67%	20%	60%
	RNF	34,64%	62%	44,55%	66,67%	10%	44%

<b>2 Kategori 65 data (Tumor Jinak dan Tumor Ganas)</b>	<b>RNN</b>	83,3%	100%	82%	75%	85,7%	55%
	<b>RNF</b>	12,5%	88,5%	42%	12,5%	85,7%	33%

5. Hasil Percobaan dengan Metode Inferensi Sugeno Orde Nol

		<b>Data Training</b>			<b>Data Testing</b>		
		<b>Sensi-tivitas</b>	<b>Spesi-fisitas</b>	<b>Akurasi</b>	<b>Sensiti-vitas</b>	<b>Spesi-fisitas</b>	<b>Akurasi</b>
<b>Orde 1</b>	<b>RNN</b>	94%*)	56%*)	81.3%*)	80%*)	40%*)	64%*)
	<b>RNF</b>	42%*)	68%*)	50,6%*)	40%*)	0%*)	24%*)
<b>Orde 0</b>	<b>RNN</b>	94%	64%	84%	73,33%	30%	56%
	<b>RNF</b>	42%	72%	45,33%	35%	0%	20%