

**KLASIFIKASI STADIUM KANKER KOLOREKTAL
MENGGUNAKAN MODEL
*RECURRENT NEURAL NETWORK***

SKRIPSI

Diajukan Kepada Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Yogyakarta

untuk Memenuhi Sebagian Persyaratan guna Memperoleh Gelar Sarjana Sains



Oleh

Reni Setianingrum

NIM 10305141006

**PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2014**

**KLASIFIKASI STADIUM KANKER KOLOREKTAL
MENGGUNAKAN MODEL
*RECURRENT NEURAL NETWORK***

SKRIPSI

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



Oleh
Reni Setianingrum
NIM 10305141006

**PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2014**

PERSETUJUAN

Skripsi yang berjudul:
**“KLASIFIKASI STADIUM KANKER KOLOREKTAL
MENGGUNAKAN MODEL
RECURRENT NEURAL NETWORK”**

Oleh:

Reni Setianingrum

NIM. 10305141006

Telah disetujui pada tanggal 13 Juni 2014
Untuk diujikan di hadapan dewan pengaji Skripsi
Program Studi Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta

Yogyakarta, 13 Juni 2014
Dosen Pembimbing


Dr. Dhoriva Urwatul Wutsqa
NIP. 19660331 199303 2 001

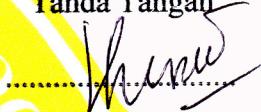
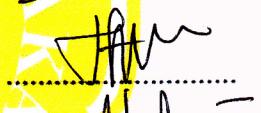
PENGESAHAN

Skripsi yang berjudul:
**“KLASIFIKASI STADIUM KANKER KOLOREKTAL
MENGGUNAKAN MODEL
RECURRENT NEURAL NETWORK”**

Yang disusun oleh:

Nama : Reni Setianingrum
NIM : 10305141006
Prodi : Matematika

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

DEWAN PENGUJI			
Nama	Jabatan	Tanda Tangan	Tanggal
<u>Dr. Dhoriva Urwatal Wutsqa</u>	Ketua Pengaji		30 Juni 2014
NIP. 19660331 199303 2 001			
<u>Musthofa, M.Sc.</u>	Sekretaris Pengaji		1 Juli 2014
NIP. 19801107 200604 1 001			
<u>Dr. Agus Maman Abadi</u>	Pengaji Utama		30 Juni 2014
NIP. 19700828 199502 1 001			
<u>Kuswari H., M.Kom.</u>	Pengaji Pendamping		1 Juli 2014
NIP. 19760414 200501 2 002			

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



Dr. Hartono
NIP. 19620329 198702 1 002

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, saya:

Nama : Reni Setianingrum
NIM : 10305141006
Program Studi : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Judul Skripsi : KLASIFIKASI STADIUM KANKER KOLOREKTAL
MENGGUNAKAN MODEL *RECURRENT NEURAL NETWORK*

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, 13 Juni 2014
Yang Menyatakan,



Reni Setianingrum
NIM 10305141006

MOTTO

"Learn from the mistakes in the past, try by using a different way, and always hope for a successful future."

"The best sword that you have is a limitless patience."

"*wa man jaahada fa-innamaa yujaahidu linafsihi*"

"Barangsiaapa bersungguh-sungguh, sesungguhnya kesungguhannya itu adalah untuk dirinya sendiri" (QS Al-Ankabut [29]: 6)

"Jangan tunda sampai besuk apa yang bisa engkau kerjakan hari ini"

"Impian dan mimpi yang tak pernah padam"

"*In ahsantum ahsantum li-anfusikum, wa ina as'tum falaha..*"

"Jika kamu berbuat baik (berarti) kamu berbuat baik bagi dirimu sendiri, dan jika kamu berbuat jahat, maka kejahatan itu untuk dirimu sendiri.." (QS. Al-Isra': 7)

PERSEMBAHAN

Syukur Alhamdullillah akhirnya saya dapat menyelesaikan skripsi ini, skripsi ini saya persembahkan untuk:

Kedua orangtua, Bapak Samidi dan Ibu Sri Sudiningsih serta keluargaku yang telah mendoakan, menyemangati, serta memberikan dukungan moril maupun materil.

Kak Rahmat Khairul yang telah menjadi penyulut semangat, dan telah sabar, serta memberikan banyak pengalaman.

Devi, dan Uswah pejuang skripsi yang telah berkerja keras berjuang bersama untuk menyelesaikan skripsi. You are the best guys!!

Noor Uswah Makhfudhoh, Fajarani Julianisti, City Wulang Mandini, Diasnita Putri Larasati, dan Kinanthi Nisa Amani yang menjadi sahabat dalam suka maupun duka dan telah merelakan waktu bersama.

Sheila, Putri, Linda, Hariti, Nita, dan sahabat lain yang telah memberikan nasehat dan bersedia untuk menjadi tempat bercerita.

Semua guru dan dosen yang telah memberikan ilmu sehingga saya menjadi seperti sekarang.

Nanang Hermawan, Andreand Rumi, Mbak Dita, Mbak Yushaila dan semua teman-teman matsub 10 dan sekolah, terimakasih atas dukungan dalam menyelesaikan skripsi ini

KLASIFIKASI STADIUM KANKER KOLOREKTAL
MENGGUNAKAN MODEL
RECURRENT NEURAL NETWORK

Oleh:
Reni Setianingrum
NIM 10305141006

ABSTRAK

Kanker kolorektal (*colo-rectal carcinoma*) atau disebut juga kanker usus besar merupakan suatu tumor ganas yang ditemukan di *colon* atau *rectum*. Salah satu cara mendekteksi adanya kanker kolorektal yaitu dengan pemeriksaan menggunakan *colonoscopy*. Setelah kanker kolorektal terdeteksi dilakukan klasifikasi untuk menentukan stadium kanker. Pada penelitian ini digunakan model *Recurrent Neural Network* (RNN) untuk klasifikasi stadium kanker kolorektal. Penelitian ini bertujuan untuk menjelaskan prosedur dan tingkat keakuratan pemodelan RNN jaringan Elman pada klasifikasi stadium kanker kolorektal dari foto *colonoscopy*.

Proses yang dilakukan adalah mengubah citra *reed green blue* (rgb) ke citra *grayscale* pada data hasil *colonoscopy*. Setelah itu dilakukan pengekstrakan gambar dengan *Gray Level Cooccurrence Matrix* (GLCM) yang didesain menggunakan *Graphical User Interface* (GUI) dengan *Matlab R2009a*. Didapat 14 fitur, yaitu energi, kontras, korelasi, *Sum of Square*, *Inverse Different Moment* (IDM), *sum average*, *sum variance*, *sum entropy*, *entropy*, *differential variance*, *differential entropy*, maksimum probabilitas, homogenitas, dan *dissimilarity*. Fitur tersebut digunakan sebagai *input*, yang selanjutnya data dibagi menjadi data *training* dan data *testing*. Setelah itu, dilakukan pemodelan RNN jaringan Elman dengan normalisasi data, perancangan model terbaik dan denormalisasi data. Perancangan model terbaik dilakukan dengan mencari jumlah neuron tersembunyi dan eliminasi *input* jaringan menggunakan algoritma *backpropagation*.

Struktur jaringan terbaik yang diperoleh adalah dengan 9 neuron *input*, 1 neuron tersembunyi dengan fungsi aktivasi *sigmoid bipolar* (*tansig*) pada lapisan tersembunyi, dan fungsi linear (*purelin*) pada lapisan *output*. Hasil model terbaik data *training* dan data *testing* diukur dengan menggunakan *sensitivitas*, *spesificitas*, dan akurasi. Sehingga dari 74 data *training* didapat 92% tingkat akurasi, 96% tingkat *sensitivitas* sebagai indikator yang dapat diandalkan ketika hasil menunjukkan kanker kolorektal, dan 79% tingkat *spesificitas* sebagai indikator yang baik ketika hasil menunjukkan kolorektal normal. Sedangkan pada 18 data *testing* menunjukkan 94% tingkat akurasi, 100%, tingkat *sensitivitas*, 80% tingkat *spesificitas*.

Kata Kunci: *recurrent neural network*, jaringan elman, klasifikasi, kanker kolorektal, citra foto *colonoscopy*

KATA PENGANTAR

Syukur Alhamdullillah penulis panjatkan kepada Allah SWT atas nikmat serta karunia yang diberikan kepada penulis untuk menyelesaikan Tugas Akhir Skripsi. Skripsi yang berjudul “Klasifikasi Stadium Kanker Kolorektal Menggunakan Model *Recurrent Neural Network*” disusun untuk memenuhi salah satu syarat kelulusan guna meraih gelar Sarjana Sains pada Program Studi Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta.

Penulis menyadari sepenuhnya bahwa dalam penulisan skripsi ini tidak lepas dari dukungan, motivasi, serta bimbingan beberapa pihak. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Bapak Dr. Hartono, M.Si, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta yang telah memberikan kelancaran pelayanan dalam urusan akademik.
2. Bapak Dr. Sugiman, M.Si, selaku Ketua Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta yang telah memberikan kelancaran pelayanan dalam urusan akademik.
3. Bapak 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. Ibu Dr. Dhoriva U. W selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan bimbingan serta masukan dalam penyusunan skripsi ini.

5. Seluruh guru dan dosen Jurusan Pendidikan Matematikan Universitas Negeri Yogyakarta yang telah memberikan ilmu kepada penulis.
6. Bapak, Ibu, serta keluarga tercinta yang selalu menyemangati penulis agar terus berjuang untuk menyelesaikan skripsi ini.
7. Teman-teman Matematika 2010 yang telah menghibur serta sama-sama berjuang menyelesaikan skripsi.
8. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah memberikan dukungan, bantuan, dan motivasi kepada penulis sehingga penyusunan serta penulisan skripsi dapat terselesaikan.

Penulis menyadari adanya ketidaktelitian, 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, 13 Juni 2014
Yang Menyatakan,



Reni Setianingrum
NIM 10305141006

DAFTAR ISI

PERSETUJUAN	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN	iv
MOTTO	v
HALAMAN PERSEMAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	vii
DAFTAR ISI.....	viix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	8
C. Tujuan Penelitian	8
D. Manfaat Penelitian	8
BAB II KAJIAN TEORI.....	10
A. Kolon dan Rektum.....	10
B. Kanker Kolorektal.....	12
1. Faktor Resiko Kanker Kolorektal	13
2. Gejala Kanker Kolorektal	16
3. Deteksi Dini	17
4. Klasifikasi Stadium Kanker Kolorektal	20
C. Pengolahan Citra Digital.....	21
1. Jenis Citra Digital.....	22
2. Ekstraksi Fitur	23
D. <i>Artificial Neural Network</i>	27
1. Pengertian <i>Neural Network</i>	29

2. Arsitektur <i>Neural Network</i>	30
3. Fungsi Aktivasi	34
4. Aturan Pembelajaran.....	37
5. Validasi Silang	38
E.Pengukuran Ketepatan Klasifikasi.....	38
F. <i>Mean Square Eror (MSE)</i>	40
BAB III PEMBAHASAN	41
A. <i>Recurrent Neural Network</i> untuk Klasifikasi	41
1. Model <i>Recurrent Neural Network</i>	41
2. Algoritma Pembelajaran <i>Recurrent Neural Network</i>	44
3. Membangun Model Recurrent Neural Network pada Jaringan Elman	50
4. Prosedur Pemodelan <i>Recurrent Neural Network</i> pada Jaringan Elman	51
B. Penerapan Model Recurrent Neural Network pada Jaringan Elman untuk Klasifikasi Stadium Kanker Kolorektal.....	55
C. Ekstraksi Gambar.....	57
1. Mengubah Tipe Gambar	57
2. <i>Gray Level Cooccurrence Matrix (GLCM)</i>	58
D. Klasifikasi Recurrent Neural Network	60
1. Menentukan Variabel	61
2. Pembagian Data	62
3. Perancangan Model.....	63
4. Pengukuran Ketepatan Klasifikasi	69
BAB IV KESIMPULAN DAN SARAN	72
A. Kesimpulan	72
B. Saran	74
DAFTAR PUSTAKA.....	75
LAMPIRAN	78

DAFTAR GAMBAR

Gambar 2.1	Lapisan Pada Dinding Usus Besar	10
Gambar 2.2	Penyebaran Kanker Kolrektal.....	12
Gambar 2.3	Neuron Jaringan Syaraf Manusia	29
Gambar 2.4	Arsitektur Jaringan Lapisan Tunggal	32
Gambar 2.5	Arsitektur Jaringan Lapisan Jamak	33
Gambar 2.6	Arsitektur Jaringan <i>Recurrent</i>	34
Gambar 2.7	Fungsi Aktivasi Sederhana.....	35
Gambar 2.8	Fungsi Linear.....	36
Gambar 2.9	Fungsi <i>Sigmoid Bipolar</i>	36
Gambar 3.1	Arsitektur Model <i>RecurrentNeural Network</i> Jaringan Elman	42
Gambar 3.2	Tahap-tahap Pemodelan <i>RecurrentNeural Network</i>	57
Gambar 3.3	Gambar Asli (RGB) dan Tipe <i>Grayscale</i>	58
Gambar 3.4	Tampilan GUI dalam Mengubah Gambar & Ekstraksi Gambar	60
Gambar 3.5	Arsitektur Jaringan Elman <i>RecurrentNeural Network</i> Model Terbaik.....	66

DAFTAR TABEL

Tabel 3.1	Nilai MSE dan Akurasi Hasil Pembelajaran	64
Tabel 3.2	Nilai MSE dan Akurasi Hasil Eliminasi.....	65
Tabel 3.3	Hasil Klasifikasi Data <i>Training</i> Menggunakan RNN	70
Tabel 3.4	Hasil Klasifikasi Data <i>Testing</i> Menggunakan RNN.....	71

DAFTAR LAMPIRAN

Lampiran 1	Gambar <i>Colonoscopy Data Training</i>	79
Lampiran 2	Gambar <i>Colonoscopy Data Testing</i>	82
Lampiran 3	Mfile GLCM Ekstraksi 14 Fitur	83
Lampiran 4	Mfile Program <i>Grapihical User Inteface (GUI)</i>	86
Lampiran 5	Tabel Hasil Ekstraksi Gambar Data <i>Training</i>	90
Lampiran 6	Tabel Hasil Ekstraksi Gambar Data <i>Testing</i>	93
Lampiran 7	Proses Pembelajaran Model <i>Recurrent Neural Network</i>	94
Lampiran 8	Nilai MSE dan Akurasi Pada Tahapan Eliminasi <i>Input</i>	95
Lampiran 9	Data <i>Target, Output, Error</i> , dan Hasil Klasifikasi Data <i>Training</i>	96
Lampiran 10	Data <i>Target, Output, Error</i> , dan Hasil Klasifikasi Data <i>Testing</i>	98
Lampiran 11	<i>Input</i> dan <i>Output</i> Data dengan Normalisasi	99

BAB I

PENDAHULUAN

A. Latar Belakang

Kanker kolorektal (*colo-rectal carcinoma*) atau disebut juga kanker usus besar merupakan suatu tumor ganas yang ditemukan di *colon* atau *rectum*. *Colon* atau *rectum* adalah bagian dari usus besar pada sistem pencernaan yang disebut juga *traktus gastrointestinal* yang berfungsi sebagai penghasil energi bagi tubuh dan membuang zat-zat yang tidak berguna. Kanker kolorektal merupakan penyakit kanker yang menempati urutan ketiga terbesar di dunia dan penyebab kematian keempat terbanyak di dunia yang disebabkan karena kanker (Gontar Alamsyah Siregar, 2007: 4). Menurut *International Agency for Research on Cancer* (IARC) (2013) dari evaluasi data-data didapatkan 1,4 juta kasus kanker kolorektal di dunia. Sedangkan di Indonesia kolorektal merupakan urutan ke sepuluh setelah kanker lain (leher rahim, payudara, kelenjar getah bening, kulit, nasofaring, ovarium, dan tiroid). Di Indonesia, terjadi kenaikan kasus pasien dengan penyakit kanker kolorektal walaupun belum terdapat angka yang pasti dari insiden kanker kolorektal. Kanker kolorektal biasanya ditandai dengan adanya polip pada kolon yang selanjutnya berubah menjadi kanker. Polip dan kanker pada stadium awal terkadang tidak menunjukkan gejala, hal ini membuat penderita kanker kolorektal baru menyadari setelah dilakukannya pemeriksaan untuk pengobatan dan didiagnosis. Pada stadium lanjut kanker kolorektal dapat menyebar ke organ-organ tubuh lainnya. Pemeriksaan dini sangatlah diperlukan

dalam upaya pencegahan atau mengetahui tingkat klasifikasi penyakit. Sehingga kita dapat mengetahui tindakan apa yang harus dilakukan setelah mengetahui tingkat klasifikasi kanker kolorektal yang diderita.

Pada tahap awal kanker, biasanya kanker tidak menunjukkan gejala, oleh karena itu pemeriksaan dini dapat mempermudah penyembuhan kanker. Pemeriksaan dini kanker kolorektal dilakukan pada laki-laki atau perempuan yang berusia lebih dari 50 tahun dan memiliki faktor resiko penyakit kanker kolorektal. Pada usia tersebut dianjurkan untuk melakukan *screening* yaitu test darah samar pada feses setiap tahun. *Screening* juga harus dilakukan oleh penderita polip dan orang yang memiliki faktor resiko kanker kolorektal. Terdapat banyak tahapan yang dapat dilakukan pada deteksi dini. Pemeriksaan lanjut untuk mendeteksi adanya sel abnormal kanker kolorektal dan mengetahui posisi sel kanker, antara lain: *colonoscopy* secara *virtual*, yaitu CT-Scan untuk membangun model 3D dari usus besar. *Double Contrast Barium Enema* (DCBE), yaitu sinar X pada usus menggunakan cairan berkapur yang dikenal sebagai barium. Selain itu untuk mengetahui penyakit kanker lebih lanjut dilakukan pemeriksaan *endoscopy* dengan *colonoscopy* setiap 10 tahun karena kanker kolorektal tersembunyi. Prosedur *colonoscopy* dilakukan dengan memasukkan kamera kecil untuk memeriksa seluruh usus besar dan rektum. Jika seseorang positif terkena kanker kolorektal, maka tindakan lanjut adalah melakukan *Carcinoembryonic Antigen* (CEA) untuk mengetahui perkembangan penyakit sebelum pengobatan dimulai.

Setelah hasil kanker terdeteksi maka dilakukan penentuan stadium kanker kolorektal untuk mengetahui seberapa jauh kanker telah menyebar ke organ

lainnya. Penentuan stadium diperlukan untuk melakukan tindakan pengobatan. Pengobatan sangatlah tergantung dari lokasi, ukuran, stadium kanker, dan kesehatan umum pasien. Berikut adalah tingkat stadium kanker (National Cancer Institute, 2006: 12) :

1. Stadium 0 (*Carsinoma in Situ*): kanker hanya pada lapisan terdalam dari kolon atau rektum.
2. Stadium I: sel kanker telah tumbuh pada dinding dalam kolon atau rektum, tapi belum menembus ke luar dinding.
3. Stadium II: sel kanker telah menyebar ke dalam lapisan otot dari kolon atau rektum. Tetapi sel kanker di sekitarnya belum menyebar ke kelenjar getah bening.
4. Stadium III: kanker telah menyebar ke satu atau lebih kelenjar getah bening di daerah tersebut, tetapi tidak ke bagian tubuh yang lain.
5. Stadium IV: kanker telah menyebar di bagian lain dari tubuh, seperti hati, paru-paru, atau tulang.

Klasifikasi penyakit adalah pengelompokan kategori penyakit menurut sel abnormal yang terdapat pada penyakit tersebut sesuai dengan kriteria yang ditetapkan. Klasifikasi penyakit kanker kolorektal berupa penentuan stadium penyakit kanker kolorektal tersebut. Penentuan stadium awal kanker kolorektal sangat diperlukan, hal ini mendorong peneliti-peneliti untuk melakukan penelitian. Banyak diantara peneliti tersebut yang menggunakan sistem komputer agar dapat mendiagnosa kanker kolorektal. Penelitian ini diharap dapat membantu

dokter menentukan lebih awal dimana sel kanker itu tumbuh sehingga dapat menentukan tingkat resiko kanker yang diderita.

Terdapat banyak penelitian mengenai klasifikasi ada kanker kolorektal menggunakan model tertentu, diantaranya klasifikasi ada tidaknya kanker usus besar menjadi tujuan penelitian Kyung-Joong Kim, dan Sung-Bae Cho (2003). Peneliti menggunakan model *Evolutionary Artificial Neural Network* (EANN) dalam pengklasifikasian. Objek dalam penelitian ini adalah *Deoxyribonucleic Acid* (DNA), karena DNA merupakan data yang penting untuk mendiagnosa sel kanker dari pasien.

Klasifikasi mengenai resiko kanker kolorektal juga dilakukan oleh Nwoye, et al (2005). Peneliti mengklasifikasikan kanker usus besar kedalam jenis adenokarsinoma normal dan ganas menggunakan *backpropagation*. Data berupa gambar yang ditangkap oleh kamera CCD pada saat proses biopsi polip kolon dengan format TIF. Gambar tersebut diubah menjadi matriks *greyscale*, yang selanjutnya diolah menggunakan *backpropagation* untuk mengklasifikasikan gambar.

Biglarian, et al (2012) juga memprediksi penyebaran atau *distant metastasis* (DM) dari kanker kolorektal menggunakan model *Artificial Neural Network*. Peneliti menggunakan data dari 1.219 pasien kanker kolorektal dan data tersebut berupa faktor-faktor yang mempengaruhi kanker kolorektal, dari data tersebut peneliti menggunakan *neural network* dan regresi logistik untuk membandingkan hasil prediksi. Berdasarkan pengujian didapat kesimpulan bahwa model *neural network* lebih baik dibanding regresi logistik dalam prediksi.

Bottaci, et al (1997), memprediksi kematian pasien yang menderita kanker kolorektal (kolon) dalam 6 jangka waktu (9, 12, 15, 18, 21, dan 24 bulan) menggunakan *artificial neural network*. Data berupa faktor-faktor yang mempengaruhi kanker kolorektal, seperti umur, jenis kelamin, riwayat keluarga, hasil tes darah, dan lain-lain.

Di Indonesia sendiri, penelitian tentang kanker kolorektal juga telah dilakukan. Salah satunya adalah klasifikasikan jenis kanker usus besar menggunakan model *Backpropagation*, dengan menggunakan input citra data gambar yang diubah ke citra *greyscale* selanjutnya dilakukan ekstraksi citra statistik yang kemudian dilanjutkan dengan model *Backpropagation*. Hasil dari klasifikasi kanker usus besar berupa pengelongan jenis kanker, yaitu *lymphoma* dan *carcinoma* (Agung Radistya Putra, Achmad Rizal, & M. Syahrul Mubarok, 2012).

Penelitian-penelitian yang sebelumnya, meneliti tentang klasifikasi kanker kolorektal berupa ada tidaknya sel kanker pada pasien, jenis kanker kolorektal, dan kematian yang disebabkan oleh kanker kolorektal. Terdapat beberapa model klasifikasi yang digunakan, yaitu analisis diskriminan, regresi logistik, dan *neural network*. Model yang digunakan dalam proses klasifikasi akan sangat berpengaruh terhadap *output* dan tingkat akurasi. Model *neural network* adalah model yang terinspirasi oleh sistem syaraf manusia. Arsitektur jaringan yang sering digunakan para peneliti adalah jaringan *feedforward* karena pada jaringan ini tidak terdapat koneksi umpan balik. Perkembangan selanjutnya para ilmuwan mengembangkan arsitektur jaringan yang memiliki koneksi umpan balik atau *recurrent*. *Recurrent*

Neural Network (RNN) memiliki beberapa lapisan, yaitu lapisan *input*, lapisan *output*, dan lapisan tersembunyi. *Recurrent Neural Network* adalah jenis jaringan pada *neural network* dimana terdapat *loop* sebagai koneksi umpan balik dalam jaringan (Lin & Lee, 1996: 340). Perbedaan antara *feedforward* dan *RNN* terletak pada ada tidaknya umpan balik antar lapisan pada struktur jaringan. Pada *RNN* paling sedikit terdapat satu lapisan umpan balik. Tentunya hal ini akan berpengaruh terhadap hasil output yang semakin baik.

Penelitian yang sudah dilakukan mengenai klasifikasi menggunakan model *RNN*, antara lain adalah Medika Risnasari (2011) melakukan klasifikasi penyakit *Disritmia* atau ketidaknormalan ritme jantung. Penelitian ini menggunakan data ekstrasi dari sinyal elektrokardiogram, dan dari tiga kelas data yang diujicobakan (*Normal Sinur Rhytm* (NSR), *Atrial Fibrillation* (AF), *Ventricular Fibrillation*.

Selain itu, Chen & Chaudhari (2003) melakukan penelitian untuk klasifikasi protein menggunakan *RNN*. Tujuan dari klasifikasi itu adalah untuk mengetahui jenis protein, yaitu globin dan non globin. Protein globin adalah protein yang dapat larut dalam air. Sedangkan protein non globin adalah protein yang tidak dapat larut dalam air

Model *RNN* memiliki beberapa keunggulan dibanding *Backpropogation*, yaitu jaringan *recurrent* merupakan perbaikan dari jaringan sebelumnya atau *feedforward*. Sehingga jaringan *RNN* memiliki kemampuan penggambaran yang sangat bagus dan dapat mengatasi kelemahan *feedforward*. Jaringan *RNN* memiliki dua macam model jaringan, yaitu jaringan Elman dan jaringan Hopfield.

Jaringan Elman dan jaringan Hopfield mempunyai kesamaan, yaitu keduanya memiliki struktur *feedback*. Pada penelitian ini digunakan jaringan Elman yang terdapat pada RNN, karena jaringan ini merupakan jaringan yang sederhana. Struktur jaringan *recurrent* khususnya pada jaringan Elman hanya terdiri dari 1 lapisan tersembunyi.

Penggunaan model *RNN* umumnya dapat diterapkan di berbagai bidang, salah satunya dalam pengklasifikasian stadium penyakit kanker kolorektal. Pengklasifikasian kanker kolorektal ini dilakukan sampai pada tahap stadium 0 (normal) sampai dengan stadium 4, sehingga penelitian ini akan menghasilkan klasifikasi yang lebih rinci. Selain itu, dalam penelitian ini data input yang digunakan adalah fitur *colonoscopy* yang selanjutnya akan diubah terlebih dahulu ke citra *greyscale*. Sebelum masuk ke dalam proses klasifikasi menggunakan *RNN*, maka dilakukan ekstraksi fitur (*feature extraction*) yang akan menghasilkan beberapa parameter. Parameter tersebut selanjutnya akan dijadikan *input*. *Input* ini yang digunakan dalam pengklasifikasian. Dengan adanya klasifikasi stadium kanker kolorektal menggunakan *RNN*, diharapkan dapat memberikan alternatif lain dalam menentukan stadium kanker kolorektal.

Berdasarkan uraian latar belakang sebelumnya, maka akan diteliti “Klasifikasi Stadium Kanker Kolorektal Menggunakan Model *Recurrent Neural Network*”. Tulisan ini diharapkan dapat berguna bagi perkembangan ilmu kesehatan dan ilmu matematika nantinya.

B. Rumusan Masalah

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

1. Bagaimana proses penentuan stadium kanker kolorektal menggunakan *RNN* ?
2. Bagaimana tingkat keakuratan penentuan stadium kanker kolorektal dengan menggunakan model *RNN* ?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk:

1. Menjelaskan proses penentuan stadium kanker kolorektal dengan menggunakan *RNN*.
2. Mendeskripsikan tingkat keakuratan penentuan stadium kanker kolorektal dengan menggunakan model *RNN*.

D. Manfaat Penelitian

Hasil penelitian ini diharapkan memiliki manfaat sebagai berikut:

1. Manfaat Teoritis

Secara teoritis, tulisan ini mampu menambah ilmu pengetahuan dan perkembangan di bidang kesehatan, dan ilmu matematika sendiri. Serta penelitian ini dapat dijadikan referensi untuk penelitian selanjutnya.

2. Manfaat Praktis

a. Bagi Peneliti

Menambah pengetahuan bagaimana proses *Gray Level Co-occurrence Matrix* (GLCM) yang dilakukan dalam mengekstrak gambar menggunakan MATLAB, serta menambah wawasan tentang model *RNN* dalam aplikasi pengklasifikasian penyakit kanker kolorektal.

b. Bagi Dokter

- 1) Memberikan alternatif dalam mendiagnosa tingkat penyakit kanker kolorektal.
- 2) Mempercepat proses pengklasifikasian.

c. Bagi UNY

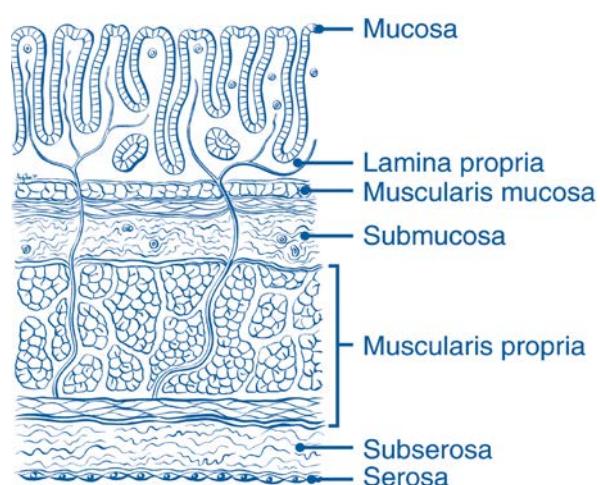
Untuk menambah koleksi bahan pustaka yang bermanfaat bagi UNY pada umumnya, dan mahasiswa Matematika dan Ilmu Pengetahuan Alam pada khususnya.

BAB II

KAJIAN TEORI

A. Kolon dan Rektum

Kolon dan rektum merupakan bagian dari usus besar pada sistem pencernaan yang disebut juga *traktus gastrointestinal*. Fungsi usus besar adalah membantu tubuh menyerap nutrisi dan cairan dari makanan yang kita makan dan minum. Setelah menyerap nutrisi yang dibutuhkan tubuh, usus kemudian membuang zat-zat yang tidak berguna dalam tubuh dalam bentuk tinja melalui rektum dan anus. Panjang usus besar (kolon dan rektum) adalah 1500 cm, dan terdiri dari sekum, kolon asenden, kolon transversum, kolon desenden, kolon sigmoid, dan rektum. Kolon berada di bagian proksimal usus besar sedangkan panjang rektum sekitar 13 cm dan berada sekitar 2-3 cm di atas tulang ekor, dimana terdapat lengkungan ke arah bawah yang membentuk lubang anus (Tortora & Grabow, 2002). Dinding usus besar memiliki tiga lapisan, berikut ini gambar 2.1 yang merupakan gambar lapisan pada dinding usus besar.



Gambar 2.1 Lapisan Pada Dinding Usus Besar (AJCC, 2010)

Pada gambar 2.1 terlihat lapisan pada dinding usus besar, yaitu lapisan mukosa (bagian dalam) yang berfungsi sebagai pencerna dan absorpsi makanan, lapisan muskularis (bagian tengah) yang berfungsi untuk menolak makanan ke bawah, dan lapisan serosa (bagian luar) yang merupakan bagian sangat licin sehingga dinding usus tidak berlengketan satu sama lain di dalam rongga abdomen.

Terdapat beberapa penyakit yang terjadi di usus besar. Hal ini juga terjadi pada kolon dan rektum. Beberapa penyakit yang terjadi pada kolon dan rektum adalah sebagai berikut (Parker, 2009: 190):

1. Kanker Kolorektal

Kanker kolorektal adalah kanker yang terjadi pada kolon dan rektum.

2. Polip kolon

Polip kolon adalah pertumbuhan daging yang terjadi pada lapisan bagian dalam usus besar. Beberapa polip tidak berbahaya, namun beberapa polip dapat berubah menjadi kanker.

3. Kolitis Ulseratif

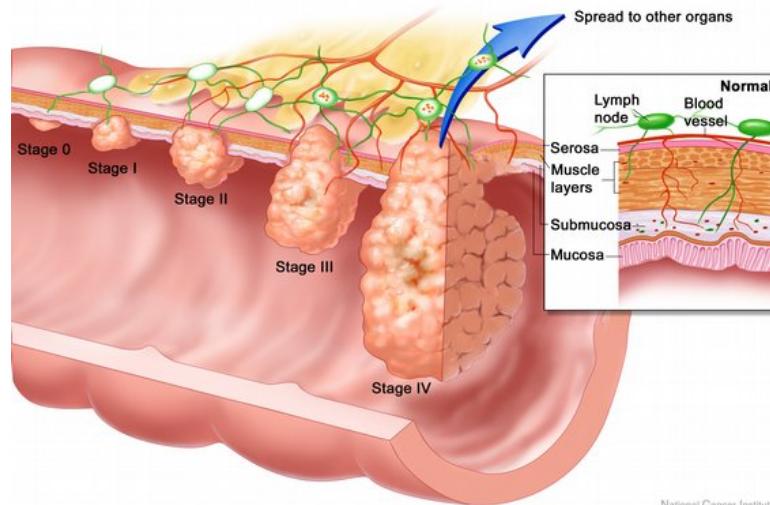
Kolitis ulseratif (*Ulcerative Colitis*) adalah peradangan pada lapisan usus besar (kolon dan rektum). Maka penyakit ini sering disebut juga penyakit radang usus(*inflammatory bowel disease*). Gejala yang sering ditimbulkan dari penyakit ini adalah pendarahan pada rektum, diare, kram perut, penurunan berat badan, dan demam. Penyakit kolitis ulseratif yang sudah diderita selama bertahun-tahun akan meningkatkan resiko terkena kanker kolorektal.

4. Penyakit Crohn

Pada penyakit Crohn, bercak radang disertai ulserasi dapat timbul di mana saja sepanjang saluran pencernaan, dari mulut hingga anus. Penyakit ini menyebabkan penyempitan anus.

B. Kanker Kolorektal

Kanker kolorektal (*colo-rectal carcinoma*) atau disebut juga kanker usus besar adalah kanker yang terjadi ketika sel-sel abnormal tumbuh pada lapisan kolon atau rektum. Umumnya, kanker kolorektal jarang ditemukan sebelum usia 40 tahun dan resiko terjadinya kanker kolorektal akan meningkat pada usia 50 tahun. Kanker kolorektal biasanya berkembang perlahan-lahan selama 10 sampai 15 tahun. Gejala adanya tumor pada kolon biasanya ditandai dengan adanya polip yang memiliki resiko kanker. Sekitar 96% penyebab kanker kolorektal adalah *adenocarcinomas* yang berkembang dari jaringan kelenjar (Alteri, et al, 2011: 2). Berikut adalah gambar 2.2 mengenai penyebaran kanker kolorektal.



Gambar 2.2 Penyebaran Kanker Kolorektal (National Cancer Institute)

Pada gambar 2.2 terlihat penyebaran kanker di kolon atau rektum. Pada saat *stage 0* atau normal tidak ditemukan adanya kanker yang tumbuh pada kolon atau rektum. Kanker tumbuh di usus besar melalui lapisan dan menembus lapisan diding usus besar atau rektum. Kanker yang telah menembus dinding juga dapat menembus darah atau kelenjar getah bening (*lymph node*), yang merupakan saluran tipis. Pada umumnya, sel-sel kanker pertama kali menyebar ke kelenjar getah bening di dekat sel kanker tersebut, kelenjar getah bening memiliki struktur seperti kacang yang membantu melawan infeksi. Sel-sel kanker itu dapat dibawa oleh pembuluh darah (*blood vessel*) ke hati, paru-paru, rongga perut, atau ovarium. Proses dimana sel-sel kanker menyebar ke organ lain melalui pembuluh darah disebut *metastasis* (Alteri, et al, 2011: 2).

1. Faktor Resiko Kanker Kolorektal

Penyebab pasti kanker kolorektal belum diketahui. Beberapa ahli tidak dapat menjelaskan, mengapa beberapa orang mengidap penyakit kanker kolorektal dan lainnya tidak, tetapi penyakit kanker kolorektal tidak menular. Penelitian menunjukkan bahwa orang yang memiliki resiko tertentu terserang kanker kolorektal. Faktor resiko adalah sesuatu yang meningkatkan kemungkinan mengembangnya penyakit. Beberapa faktor resiko yang mempengaruhi terjadinya kanker kolorektal, yaitu :

a. Usia

Usia mempengaruhi daya tahan tubuh manusia. Makin tua usia, makin beresiko seseorang terkena penyakit. Orang yang berusia kurang lebih 50

beresiko terkena kanker kolorektal. Lebih dari 90 persen dari orang yang didiagnosa terkena kanker kolorektal berusia 50 keatas.

b. Riwayat Keluarga

Riwayat keluarga mempengaruhi perkembangan keganasan kanker usus. Bahkan faktor ini adalah faktor resiko yang paling umum mempengaruhi kanker usus selain faktor usia. Terdapat banyak organisasi yang menggunakan tes genetik untuk diagnosis lanjut kanker usus besar (Burt, 2000: 837).

c. Riwayat Penyakit

Penyakit yang terdapat pada seseorang dapat memicu penyakit-penyakit lainnya, begitu juga dengan kanker kolorektal dapat dipicu oleh beberapa penyakit, yaitu:

1) Penyakit Polip Kolon

Polip adalah pertumbuhan jaringan yang berkembang pada lapisan usus besar atau rektum yang dapat menjadi kanker (Alteri, et al, 2011: 2). Terdapat beberapa jenis polip, yaitu polip *adenomatus* atau adenoma, polip *hiperplastik*, dan polip *inflamasi*. Polip adenoma merupakan polip yang dapat berubah menjadi kanker, sedangkan polip *inflamasi* dan *hiperplastik* bukan prekanker. Namun jika polip *hiperplastik* tumbuh pada kolon sisi sebelah kanan maka dapat menimbulkan kanker (Gontar Alamsyah Siregar, 2007: 4).

2) Penyakit Radang Usus

Penyakit radang usus adalah suatu kondisi dimana usus besar yang meradang selama jangka waktu yang lama. Pasien yang terkena radang usus besar

dalam jangka waktu yang lama akan mengembangkan *dysplasia*. *Dysplasia* istilah yang digunakan untuk menggambarkan sel-sel lapisan usus besar atau rektum yang terlihat normal (tetapi tidak seperti sel kanker sebenarnya) jika dilihat dari mikroskop. Selanjutnya jika radang dibiarkan maka sel-sel ini berubah menjadi kanker (Japerson, et al, 2010: 336). Penyakit radang usus meliputi dua keadaan dengan gejala yang mirip kolitis ulceratif dan crohn (Parker, 2009: 190). Gejala kedua kelainan ini adalah nyeri perut, diare, hilang nafsu makan, demam, pendarahan usus, dan berat badan turun.

3) Penyakit Diabetes

Banyak penelitian yang menemukan hubungan antara diabetes dan peningkatan resiko kanker kolorektal. Ciri-ciri fisik yang ditunjukkan oleh penderita diabetes (tipe 2) hampir sama dengan penderita kanker, seperti aktifitas fisik, indeks massa tubuh, dan lain-lain. Tetapi hal ini lebih banyak ditemukan pada pria daripada wanita (Alteri, et al, 2011: 10).

d. Kelebihan Berat Badan

Kelebihan berat badan atau obesitas dapat mempertinggi resiko terkena kanker kolorektal, hal ini terjadi lebih sering pada pria daripada wanita. Semakin besar ukuran lingkar badan seseorang semakin besar juga resiko kanker kolorektal (Alteri, et al, 2011: 10).

e. Diet

Pola diet yang salah pada seseorang dapat memicu tumbuhnya kanker kolorektal. Banyak orang yang ingin memiliki bentuk badan yang ideal dengan cara diet yang salah yaitu dengan mengurangi kalori yang dimakan secara

besar-besaran. Pola diet yang salah ini dapat menyebabkan tubuh kekurangan vitamin dan mineral.

f. Komsumsi Alkohol

Dampak buruk dari penggunaan alkohol akan mengenai bebagai organ di dalam tubuh, yaitu otak, pencernaan mulai dari mulut sampai usus besar, liver, pankreas, dan otot tulang. Alkohol dapat menyebabkan peradangan kronis pada saluran pencernaan, membentuk erosi sampai tukak usus dan selanjutnya akan menyebabkan perubahan struktur dalam usus sampai berubah menjadi sel ganas atau kanker (Widodo Judarwanto, 2006: 92)

g. Faktor Makanan

Makanan yang dikonsumsi tentunya mempengaruhi kesehatan seseorang. Seseorang yang mengkonsumsi makanan tinggi lemak, rendah serat, dan bahan makanan yang mengandung *karsinogen* (pemicu kanker) dapat mempertinggi resiko terkena kanker kolorektal. Sedangkan seseorang yang rajin mengkonsumsi makanan tinggi serat dan multivitamin serta multimineral akan menurunkan resiko kanker hingga 84 persen.

2. Gejala Kanker Kolorektal

Gejala umum dari kanker kolorektal ditandai oleh perubahan kebiasaan buang air besar. Gejala tersebut meliputi (Alteri, et al, 2011: 13):

- a. Diare atau sembelit.
- b. Merasa bahwa usus tidak kosong.
- c. Ditemukannya darah (baik merah terang atau sangat gelap) di feses.
- d. Feses yang dikeluarkan lebih sedikit dari biasanya.

- e. Sering mengalami sakit perut, kram perut, atau perasaan penuh atau kembung.
- f. Kehilangan berat badan tanpa alasan yang diketahui.
- g. Merasa sangat lelah sepanjang waktu.
- h. Mual atau muntah.

Gejala-gejala ini biasanya bukan merupakan gejala kanker pada umumnya, tetapi merupakan gejala penyakit lainnya dan biasa dirasakan oleh manusia. Jika terdapat seseorang merasakan gejala ini, seharusnya diperlukan diagnosa dokter supaya dapat ditangani sedini mungkin, karena biasanya kanker dini tidak menimbulkan rasa sakit.

3. Deteksi Dini

Deteksi dini penyakit kanker kolorektal sangat diperlukan. Semakin cepat deteksi itu dilakukan, maka semakin cepat dokter melakukan tindakan. Jika penyakit kanker diketahui pada stadium awal maka peluang untuk sembuh dari penyakit kanker semakin besar. Terdapat berbagai macam cara untuk mendeteksi kanker kolorektal pada pasien, diantaranya (American Cancer Society, 2011: 13) :

- a. *Screening*

Screening kanker kolorektal adalah proses mencari sel kanker atau pra-kanker pada orang yang tidak memiliki gejala penyakit. Dari waktu sel-sel abnormal pertama mulai tumbuh menjadi polip, biasanya membutuhkan waktu sekitar 10 sampai 15 tahun polip tersebut berkembang menjadi kanker kolorektal. *Screening* yang dilakukan secara rutin dapat mencegah kanker kolorektal. Hal ini dikarenakan polip yang ditemukan dapat disembuhkan dan dihilangkan sebelum berubah menjadi sel kanker. Proses *screening* juga dapat

digunakan untuk menemukan sel kanker sedini mungkin, sehingga kanker berpeluang besar untuk sembuh. *Screening* dapat dilakukan secara rutin pada usia 50 tahun, pada orang yang memiliki riwayat keluarga terkena kanker, dan orang yang memiliki faktor resiko kanker.

b. *Flexible sigmoidoscopy*

Proses ini dilakukan dengan melihat salah satu bagian dari usus besar dan rektum dengan *sigmoidoscopy* fleksibel, alat ini memiliki lampu pada tabung yang berukuran setebal jari dengan kamera kecil pada ujung alat. Alat ini dimasukkan melalui rektum dan bagian bawah usus besar. Gambar itu akan terlihat pada layar monitor. Dengan menggunakan *sigmoidoscopy* maka dokter dapat melihat bagian dalam rektum dan usus besar untuk mendeteksi kelainan apapun. Karena *sigmoidoscopy* berukuran 60 cm, maka dokter dapat melihat seluruh rektum tetapi hanya dapat melihat setengah bagian dari usus besar.

c. *Double Contrast Barium Enema (DCBE)*

Pendeteksi kanker menggunakan DCBE ini menggunakan barium dengan kontras udara. Barium sulfat merupakan cairan berkapur, dan udara digunakan untuk menguraikan bagian dalam usus besar dan rektum untuk mencari daerah yang mengandung sel abnormal. Jika terdapat daerah yang mencurigakan pada tes ini yang dilihat menggunakan sinar X maka dilakukan tes *Colonoscopy* untuk mengetahui penyakit lebih lanjut. Dengan kata lain tes ini hanya dapat membantu dokter untuk mengetahui posisi sel abnormal.

d. CT-Scan (Virtual Colonoscopy)

CT scan adalah sinar X yang menghasilkan gambar penampang rinci tubuh. Jika pada tes sinar X, gambar yang diambil hanya dari satu arah. Pada CT scan, terdapat banyak gambar yang dapat diambil dari berbagai arah. Lalu gambar-gambar irisan bagian tubuh ini akan digabungkan untuk dipelajari kembali oleh dokter. Terdapat dua jenis CT *colonography*, yaitu dengan dua dimensi dan tiga dimensi. Tes ini memungkinkan dokter mencari polip atau kanker.

e. *Colonoscopy*

Pada tes ini, dokter melihat seluruh panjang usus besar dan rektum dengan *colonoscope*. *Colonoscope* adalah versi lama dari *sigmoidoscopy*. Alat ini dimasukkan melalui rektum ke dalam usus besar. *Colonoscope* memiliki kamera video di ujung yang terhubung ke *display* sehingga dokter dapat melihat dan meneliti bagian dalam usus besar. Dengan alat *colonoscopy* dapat dilakukan deteksi dan pembuangan polip serta biopsi kanker selama pemeriksaan.

f. Tes Darah Tinja

Tes ini untuk mencari darah samar (darah yang tidak dapat dilihat dengan mata telanjang) dalam tinja. Tes ini dilakukan karena jika seseorang terkena polip atau kanker kolorektal maka pembuluh darah di permukaan sering rapuh dan mudah rusak oleh berlalunya feses.

g. *Carcinoembryonic Antigen (CEA)*

CEA adalah zat yang ditemukan dalam darah beberapa orang yang sudah terkena kanker kolorektal. Dokter menggunakan tes ini untuk mengetahui perkembangan penyakit sebelum pengobatan dimulai. Tes ini memudahkan dokter untuk mengambil tindakan lanjut dari pengobatan.

4. Klasifikasi Stadium Kanker Kolorektal

Menurut Prytherch (2005: 140) klasifikasi adalah pengelompokan benda secara logis menurut ciri-ciri kesamaannya. Sedangkan klasifikasi penyakit dapat didefinisikan sebagai suatu sistem yang mengkategorikan tingkat keganasan penyakit dalam beberapa kriteria yang ditetapkan (WHO, 1977).

Klasifikasi penyakit kanker kolorektal berarti mengelompokkan kanker kolorektal ke dalam stadium yang tepat, berdasarkan ciri-ciri tertentu. Ciri-ciri itu didapatkan dari hasil *colonoscopy* penderita kanker kolorektal ataupun bukan penderita kanker kolorektal (normal). Klasifikasi stadium kanker kolorektal pada umumnya dibagi menjadi beberapa tingkatan dengan angka 0 sampai dengan IV. Tahap klasifikasi stadium kanker ini merupakan tahap yang penting, karena membantu dokter untuk melakukan tindakan selanjutnya. Terdapat beberapa sistem klasifikasi stadium kanker, diantaranya adalah Astler-Coller, Dukes, FIGO (*Federasi Internasional Ginekologi dan Obstetri*) dan sistem TNM Kanker. Sistem klasifikasi TNM dan FIGO memiliki sistem yang sangat mirip. Kedua sistem ini mengelompokkan kanker meliputi ukuran tumor, tingkat penyebaran ke kelenjar getah bening, dan adanya metastasis.

Menurut National Cancer Institute (2006: 12) klasifikasi stadium kanker kolorektal dapat didefinisikan sebagai berikut :

- a. Stadium 0 (*Carcinoma in Situ*) : kanker hanya pada lapisan terdalam dari kolon atau rektum.
- b. Stadium I : sel kanker telah tumbuh pada dinding dalam kolon atau rektum, tapi belum menembus ke luar dinding.
- c. Stadium II : sel kanker telah menyebar ke dalam lapisan otot dari kolon atau rektum. Tetapi sel kanker di sekitarnya belum menyebar ke kelenjar getah bening.
- d. Stadium III : kanker telah menyebar ke satu atau lebih kelenjar getah bening di daerah tersebut, tetapi tidak ke bagian tubuh yang lain.
- e. Stadium IV : kanker telah menyebar di bagian lain dari tubuh, seperti hati, paru-paru, atau tulang.

C. Pengolahan Citra Digital

Istilah citra pada pengolahan citra dapat diartikan sebagai suatu fungsi kontinu dari intensitas cahaya dalam dua dimensi. Pengolahan citra pada komputer memerlukan citra digital sebagai *input*. Citra digital adalah citra kontinu yang diubah dalam bentuk diskrit. Citra digital terdiri dari titik-titik kecil berbentuk segiempat, yang disebut sebagai *picture element*, atau piksel. Banyaknya piksel per satuan panjang akan menentukan resolusi citra tersebut. Semakin banyak piksel, maka semakin tinggi resolusinya dan makin halus gambarnya, sehingga titik-titik kecil pada tampilan citra digital tidak terlihat dengan kasat mata. Citra

yang dimaksud dalam penelitian ini adalah citra hasil *colonoscopy* yang merupakan representasi dari berbagai jenis stadium kanker kolorektal.

Pengolahan citra biasa dikenal dengan *image processing* adalah proses mengolah piksel-piksel dalam citra digital untuk suatu tujuan. Salah satunya adalah untuk memperoleh citra digital baru dengan karakteristik tertentu dan cocok secara visual yang dibutuhkan untuk tahap yang lebih lanjut dalam pemrosesan analisis citra.

1. Jenis Citra Digital

Menurut Sianipar (2013: 200-202) terdapat tiga macam jenis citra digital. Diantaranya adalah citra biner, citra abu-abu (*greyscale*), dan citra warna (RGB).

a. Citra Biner

Citra biner adalah citra digital yang setiap piksel hitam atau putih. Citra biner memiliki piksel yang bernilai nol untuk hitam dan satu untuk putih. Citra seperti ini biasanya dimanfaatkan dalam representasi biner dari teks, sidik jari, hasil foto rongseng, atau rancangan arsitektur.

b. Citra Abu-abu (*Greyscale*)

Citra Abu-abu adalah citra yang nilai pikselnya merepresentasikan derajat keabuan atau intensitas warna putih. Setiap piksel merupakan bayangan abu-abu yang memiliki intensitas 0 (hitam) sampai 255 (putih).

c. Citra Warna (RGB)

Citra Warna adalah citra yang nilai pikselnya merepresentasikan suatu warna tertentu. Warna tersebut dideskripsikan oleh banyak warna merah (R, *red*), hijau (G, *green*), dan biru (B, *blue*). Visual citra warna lebih kaya

dibandingkan dengan citra biner, karena citra warna menampilkan objek seperti warna aslinya. Setiap komponen dari citra warna memiliki rentang 0-255, sehingga terdapat $255^3 = 16.777.216$ kemungkinan jenis warna.

2. Ekstraksi Fitur

Ekstraksi fitur (*feature extraction*) atau juga dikenal dengan sebutan *indexing*, adalah teknik pengambilan sifat-sifat khusus dari sebuah gambar. Ekstraksi fitur merupakan langkah awal klasifikasi. Ekstraksi fitur citra berdasarkan tekstur pada orde pertama menggunakan metode statistik, yaitu dengan melihat distribusi derajat keabuan pada histogram citra tersebut.

Terdapat banyak teknik-teknik dalam mengekstrak gambar. Salah satunya adalah dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Proses GLCM akan menghasilkan parameter yang digunakan dalam analisis citra. Parameter dasar itu berupa energi, kontras, korelasi, *Sum of Square*, *Inverse Different Moment* (IDM), *sum average*, *sum variance*, *sum entropy*, *entropy*, *differential variance*, *differential entropy*, maksimum probabilitas, homogenitas, *dissimilarity*, dan lainnya.

a. Energi

Energi disebut juga *Angular Second Moment* (ASM) menunjukkan ukuran sifat homogenitas pada citra digital (Aditya Angga Kusuma, R. Rizal Isnanto, & Imam Santoso, 2012: 3). Rumus untuk mencari energi (ASM) adalah (Anami & Vishwanath, 2009: 11) :

$$ASM = \sum_{i,j=1}^{N_g} g(i,j)^2 \quad (2.1)$$

dengan,

N_g adalah banyak tingkat abu-abu dari citra,

$g(i,j)$ adalah piksel di lokasi baris ke- i dan kolom ke- j .

b. Kontras

Kontras adalah perhitungan perbedaan antara piksel satu dan piksel yang berdekatan diseluruh gambar. Kontras (C) bernilai nol untuk gambar yang konstan (Sharma & Sourabh, 2013: 331). Rumus untuk mencari kontras adalah (Anami & Vishwanath, 2009: 11) :

$$C = \sum_{i,j=1}^{N_g} g(i,j) (i - j)^2 \quad (2.2)$$

c. Korelasi

Korelasi mengukur ketergantungan linear derajat keabuan dari suatu piksel (Mohanaiah, Sathyanarayana, & GuruKumar, 2013: 2). *Digital Image Correlation* adalah metode yang menekankan pada pelacakan dan teknik keakuratan 2D dan 3D pada gambar. Metode ini sering digunakan untuk mengukur perubahan bentuk, perpindahan, dan gerakan pada *mouse* serta bidang ilmu pengetahuan lainnya. Rumus untuk mencari korelasi (Cr) adalah (Anami & Vishwanath, 2009: 11):

$$Cr = \sum_{i,j=1}^{N_g} \frac{(ij)g(i,j) - \mu_i \mu_j}{\sigma_i \sigma_j} \quad (2.3)$$

dengan,

μ_i, μ_j adalah rata-rata dari g_i, g_j

σ_i, σ_j adalah standar deviasi dari g_i, g_j

d. *Sum of Squares (Variance)*

Menunjukkan variasi elemen-elemen matriks *coocurance*. Rumus untuk menghitung *variance* (VAR) pada citra digital adalah (Haralick, Shanmugam & Dinstein, 1973: 619) :

$$VAR = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i,j) (i - \mu)^2 \quad (2.4)$$

μ adalah rata-rata dari $\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i,j)$

e. *Inverse Difference Moment (IDM)*

Inverse Difference Moment (IDM) atau homogenitas lokal adalah tingkat kehomogenan citra yang berderajat keabuan sejenis. Rumus IDM suatu gambar adalah (Mohanaiah, Sathyanarayana, & GuruKumar, 2013: 2) :

$$IDM = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i,j)}{1 + (i - j)^2} \quad (2.5)$$

f. *Sum Average*

Rumus untuk menghitung *sum average* (SA) pada citra digital adalah (Haralick, Shanmugam & Dinstein, 1973: 619) :

$$SA = \sum_{i=2}^{2N_g} k g_{x+y}(k) \quad (2.6)$$

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

g. *Sum Entropy*

Rumus untuk menghitung *sum entropy* (SE) pada citra digital adalah (Haralick, Shanmugam & Dinstein, 1973: 619) :

$$SE = - \sum_{i=2}^{2N_g} \sum_{j=2}^{2N_g} g_{x+y}(k) \log\{g_{x+y}(k)\} \quad (2.8)$$

h. Sum Varians

Sum varians menunjukkan seberapa derajat keabuan yang beragam dari rata-rata. Rumus untuk menghitung *sum varians* (SV) pada citra digital adalah (Haralick, Shanmugam & Dinstein, 1973: 619) :

$$SV = \sum_{i=2}^{2N_g} \sum_{j=2}^{2N_g} (i - SA)^2 g_{x+y}(k) \quad (2.7)$$

i. Entropy

Entropy menunjukkan ukuran ketidakteraturan intensitas gambar. Nilai *entropy* akan besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur. Rumus *Entropy* (EN) dari suatu gambar adalah (Sharma & Sourabh, 2013: 331) :

$$EN = - \sum_{i,j=1}^{N_g} g(i,j) \log_2\{g(i,j)\} \quad (2.9)$$

j. Differential Variance

Rumus untuk menghitung *differential variance* (DV) pada citra digital adalah (Haralick, Shanmugam & Dinstein, 1973: 619) :

$$DV = \text{variance of } (g_{x-y}) \quad (2.10)$$

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

k. Differential Entropy

Rumus untuk menghitung *differential variance* (DE) pada citra digital adalah (Haralick, Shanmugam & Dinstein, 1973: 619) :

$$DE = - \sum_{i,j=0}^{N_g-1} g(i,j) \log_2 \{g(i,j)\} \quad (2.11)$$

I. Maksimum Probabilitas

Maksimum probabilitas menunjukkan derajat keabuan yang memenuhi relasi pada persamaan *entropy*. Rumus untuk menghitung maksimum probabilitas (MP) adalah (Anami & Vishwanath, 2009: 11) :

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

m. Homogenitas

Secara matematis, homogenitas pada GLCM adalah invers dari kontras pada GLCM, yaitu keseragaman intensitas keabuan pada citra. Rumus untuk mencari homogenitas (H) suatu gambar adalah (Sharma & Sourabh, 2013: 331):

$$H = \sum_{i,j=1}^{N_g} \frac{g(i,j)}{1 + (i - j)^2} \quad (2.13)$$

n. Dissimilarity

Dissimilarity menunjukkan perbedaan tiap piksel. Rumus untuk menghitung *dissimilarity* (D) adalah (Anami & Vishwanath, 2009: 11) :

$$D = \sum_{i,j=0}^{N_g-1} g(i,j) |i - j| \quad (2.14)$$

D. Artificial Neural Network

Artificial Neural Network (ANN) atau jaringan syaraf tiruan sudah berkembang sejak tahun 1943 ketika Mc Culloch seorang ahli neuro biologi dan Walter Pitts seorang ahli statistika menerbitkan makalah yang berjudul “*A Logical*

Calculus of Ideas Imminent in Nervous Activity” di salah satu buletin matematika. Makalah tersebut terinspirasi dari perkembangan komputer digital modern. Pada tahun 1958, Frank Rosenblatt terinspirasi dari makalah ini dan kemudian melakukan suatu penelitian yang menyebabkan tercetusnya pengetahuan mengenai ANN generasi pertama yaitu *Perceptron* (Hu & Hwang, 2002:13).

Penelitian-penelitian yang sudah ada, menunjukkan bahwa ANN memiliki kemampuan prediksi yang tepat dan pola klasifikasi yang kuat (Zhang, 2004). *Artificial Neural Network* (ANN) banyak digunakan dalam menyelesaikan permasalahan di berbagai bidang industri, bisnis dan ilmu. Beberapa aplikasi ANN adalah sebagai berikut (Zhang, 2004: 160):

a. Klasifikasi (*Classification*)

b. Pengenalan Pola (*Pattern Recognition*)

Artificial Neural Network dapat digunakan untuk mengenali pola (misalkan huruf, angka, suara, dan tanda tangan) yang sudah sedikit berubah. Hal ini sama dengan otak manusia yang masih mampu mengenali orang yang beberapa waktu tidak dijumpai.

c. *Signal Processing*

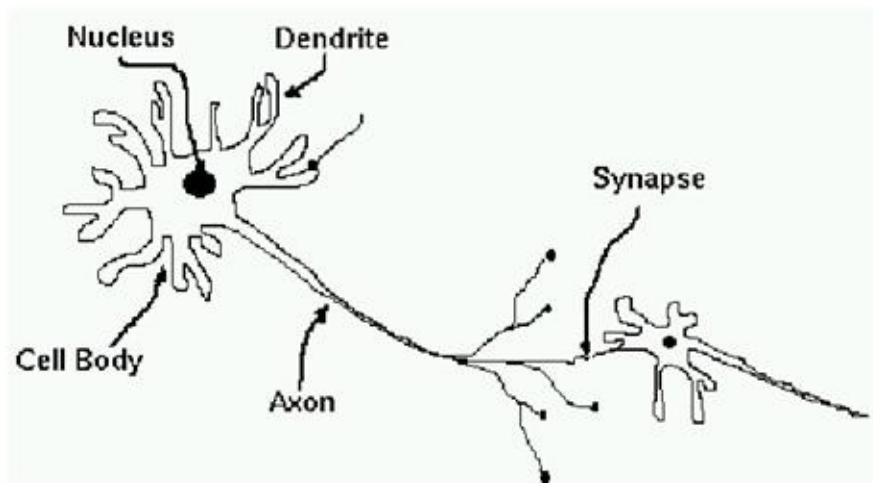
Artificial Neural Network dapat digunakan untuk menekan *error* atau *noise* dalam saluran telepon, agar gangguan pada telepon dapat diminimalkan.

d. Peramalan (*Forecasting*)

Artificial Neural Network juga dapat digunakan untuk meramal apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang sudah ada di masa lampau.

1. Pengertian *Artificial Neural Network*

Menurut Fausett (1994: 3) ANN adalah suatu sistem pemrosesan yang mempunyai karakteristik kinerja tertentu yang mengambil analogi dari jaringan syaraf biologi. Struktur ANN terinspirasi dari jaringan biologi manusia, khususnya otak manusia yang dapat memproses suatu informasi. Otak manusia terdiri dari beberapa neuron yang saling terhubung. Neuron yang saling terhubung menangkap, mengelola, dan menyalurkan informasi. Neuron memiliki tiga komponen yang penting, yaitu dendrit, badan sel (soma), dan axon. Neuron menerima sinyal yang berupa implus dari neuron-neuron lain melalui dendrit yang dikirim melalui celah sinapsis. Selanjutnya soma menjumlahkan semua sinyal-sinyal yang masuk, jika jumlahnya melebihi ambang batas (*threshold*) maka sinyal akan diteruskan ke neuron lain melalui axon. Gambar 2.3 jaringan syaraf pada manusia.



Gambar 2.3 Neuron Jaringan Syaraf Manusia

Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian ANN (Jong Jek Siang, 2005: 23). Seperti otak manusia, ANN

juga terdiri dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut mentransformasikan informasi (input) yang diterima melalui sambungan keluarnya menuju neuron yang lain. Hubungan ini disebut bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tertentu. Bobot ini akan digunakan dalam proses pembelajaran.

Artificial Neural Network dibentuk sebagai generalisasi model matematika dari jaringan syaraf manusia, dengan asumsi (Fausett, 1994: 3):

- a. informasi diproses oleh elemen-elemen sederhana yang disebut neuron,
- b. sinyal dikirim antar neuron yang saling terhubung,
- c. setiap penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal,
- d. setiap neuron memiliki fungsi aktivasi yang dikenakan pada *input* yang diterima untuk menentukan besarnya *output*.

Artificial Neural Network ditentukan oleh 3 hal yaitu:

- a. fungsi aktivasi,
- b. pola hubungan antar neuron (disebut arsitektur jaringan),
- c. metode untuk menentukan bobot penghubung (disebut metode *training* atau *learning*).

2. Arsitektur *Artificial Neural Network*

Kerangka kerja *Artificial Neural Network* (ANN) dapat dilihat dari banyak lapisan (*layer*) dan banyak simpul (*node*) pada setiap lapisan. Lapisan-lapisan penyusun ANN dibagi menjadi tiga (Samarasinghe, 2007) yaitu:

a. Lapisan *input*

Data yang akan dianalisa dimasukkan pada lapisan *input*. Fungsi lapisan *input* adalah menampung dan menyebarkan sinyal yang ada melalui neuron ke lapisan berikutnya, oleh karena itu lapisan *input* memiliki fungsi yang sama dengan dendrit pada otak manusia. Neuron yang terdapat dalam lapisan *input* disebut neuron *input*.

b. Lapisan tersembunyi

Lapisan tersembunyi terletak diantara lapisan *input* dan lapisan *output*. Lapisan ini berfungsi mengelola sinyal yang ada dalam bentuk fungsi transformasi neuron. Neuron yang terdapat ada lapisan tersembunyi disebut neuron tersembunyi.

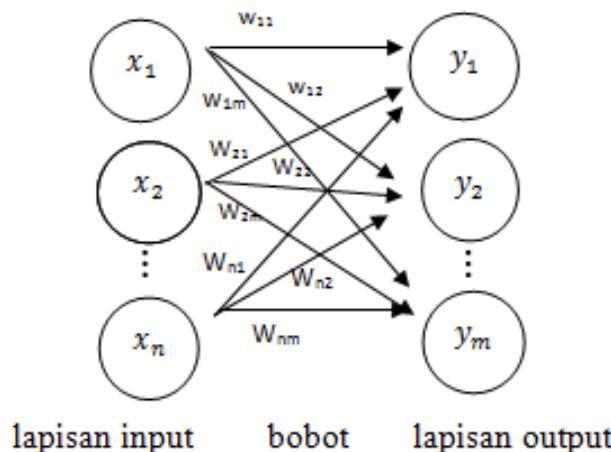
c. Lapisan *output*

Lapisan *output* merupakan tempat hasil dari proses selama dalam jaringan atau hasil operasi ANN. Hal ini juga dapat diartikan bahwa *output* adalah suatu nilai yang merupakan hasil dari fungsi aktivasi. *Output* dapat berupa keluaran dari jaringan atau menjadi *input* bagi node lain. Lapisan *output* memiliki fungsi yang sama dengan axon pada otak manusia.

Pada ANN neuron-neuron tersusun dalam lapisan. Pengaturan neuron dalam lapisan dan hubungan neuron satu dengan neuron-neuron lain pada lapisan terdekat disebut dengan arsitektur jaringan. Pada umumnya terdapat tiga arsitektur ANN (Haykin, 1999: 43-44). Arsitektur ANN tersebut, adalah:

a. Jaringan Syaraf dengan Lapisan Tunggal (*Single Layer Network*)

Jaringan syaraf ini hanya terdiri dari lapisan *input* yang langsung terhubung ke lapisan *output* dan tidak ada lapisan tersembunyi. Jaringan syaraf dengan lapisan tunggal adalah jaringan umpan maju (*feedforward*). Jaringan ini memiliki satu lapisan dengan bobot-bobot terhubung. Berikut adalah arsitektur jaringan lapisan tunggal yang diperlihatkan pada gambar 2.4.

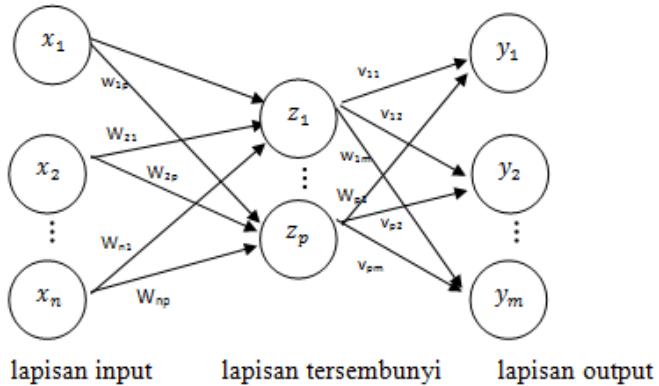


Gambar 2.4 Arsitektur Jaringan Lapisan Tunggal

Pada gambar 2.4 diperlihatkan bahwa arsitektur jaringan lapisan tunggal terdiri dari n buah *input* (x_1, x_2, \dots, x_n) dan m buah *output* (y_1, y_2, \dots, y_m) serta bobot w . Bobot-bobot ini saling independen. Selama pelatihan bobot-bobot tersebut akan dimodifikasi untuk mendapatkan hasil yang akurat.

b. Jaringan Syaraf dengan Lapisan Jamak (*Multi Layer Network*)

Jaringan syaraf ini terdiri dari lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Jaringan ini adalah pengembangan dari *single layer* dengan menambahkan satu atau lebih lapisan tersembunyi. Berikut adalah arsitektur jaringan lapisan jamak yang diperlihatkan pada gambar 2.5.



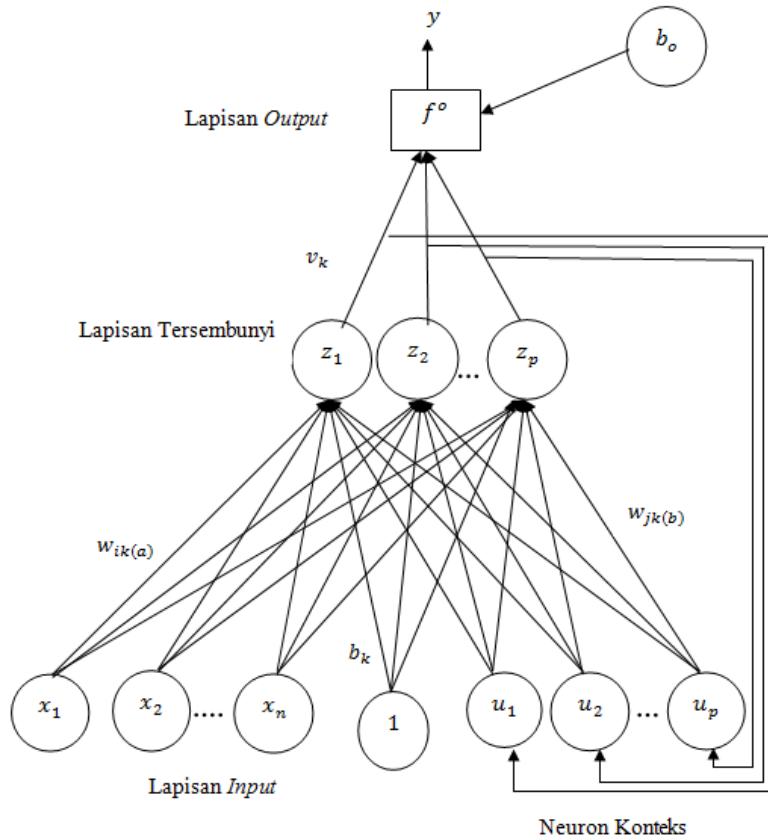
Gambar 2.5 Arsitektur Jaringan Lapisan Jamak

Pada Gambar 2.5 diperlihatkan bahwa arsitektur jaringan lapisan jamak terdiri dari n buah *input* (x_1, x_2, \dots, x_n), sebuah lapisan tersembunyi yang terdiri dari p buah neuron (z_1, z_2, \dots, z_p) dan m buah *output* (y_1, y_2, \dots, y_m).

c. Jaringan *Recurrent*

Jaringan *Recurrent* berbeda dengan jaringan syaraf lainnya, yaitu jaringan dengan lapisan tunggal dan jaringan syaraf dengan lapisan jamak.

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi neuron aktif. Model jaringan ini mirip dengan *multilayer* namun terdapat neuron *output* yang memberikan sinyal pada neuron *input* (*feedback loop*). Berikut adalah arsitektur jaringan *recurrent* yang diperlihatkan pada gambar 2.6.



Gambar 2.6 Arsitektur Jaringan *Recurrent*

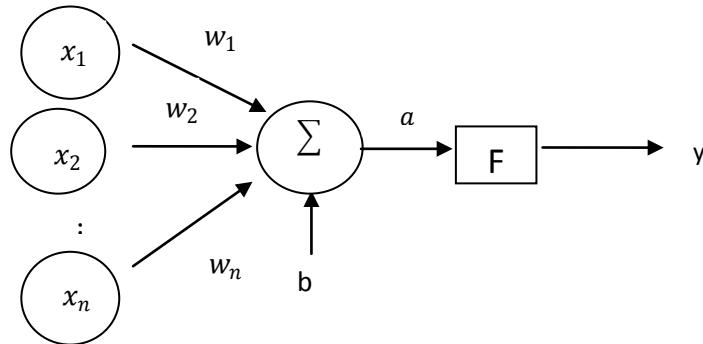
Pada gambar 2.6 diperlihatkan bahwa arsitektur jaringan *recurrent* terdiri dari n buah *input* (x_1, x_2, \dots, x_n), sebuah lapisan tersembunyi yang terdiri dari p buah neuron (z_1, \dots, z_p) dan *output* (y).

3. Fungsi Aktivasi

Fungsi aktivasi akan menentukan *output* suatu neuron (mengubah sinyal *input* menjadi sinyal *output*) yang akan dikirim ke neuron lain (Fausett, 1994:17). Fungsi aktivasi berfungsi untuk mengaktifkan neuron yang dipakai ada jaringan tersebut. Misalkan terdapat sebuah neuron yang akan mengolah n buah *input* (x_1, x_2, \dots, x_n) yang masing-masing memiliki bobot (w_1, w_2, \dots, w_n) dan bobot bias b , dengan rumus:

$$a = \sum_{i=1}^n x_i w_i + b \quad (2.15)$$

kemudian fungsi aktivasi F akan mengaktifasi a menjadi *output* jaringan y .



Gambar 2.7 Fungsi Aktivasi Sederhana (Sri Kusumadewi, 2004)

Terdapat beberapa fungsi aktivasi yang sering digunakan dalam *ANN*.

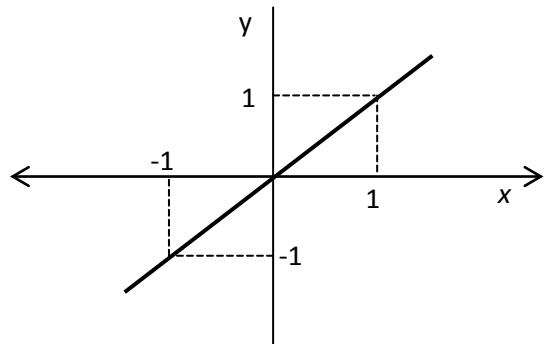
Menurut Sri Kusumadewi (2004: 51) fungsi aktivasi yang disediakan dalam *toolbox* Matlab, antara lain: fungsi undak biner (*hardlim*), fungsi bipolar (*hardlims*), fungsi linear (*purelin*), fungsi *saturating linear* (*satlin*), fungsi *symmetric saturating linear* (*satlins*), fungsi sigmoid biner (*logsig*), dan fungsi sigmoid bipolar (*tansig*). Fungsi aktivasi yang akan digunakan dalam tugas akhir ini dengan pembelajaran jaringan *recurrent* adalah:

a. Fungsi Linear (Identitas)

Fungsi ini memiliki nilai *output* yang sama dengan *inputnya*. Pada *toolbox* Matlab dikenal dengan nama *purelin*, dengan syntax: $y = \text{purelin}(x)$ dan dirumuskan sebagai berikut:

$$y = x \quad (2.16)$$

Gambar 2.8 grafik fungsi linear adalah sebagai berikut :



Gambar 2.8 Fungsi Linier (Sri Kusumadewi, 2004)

b. Fungsi Sigmoid Bipolar

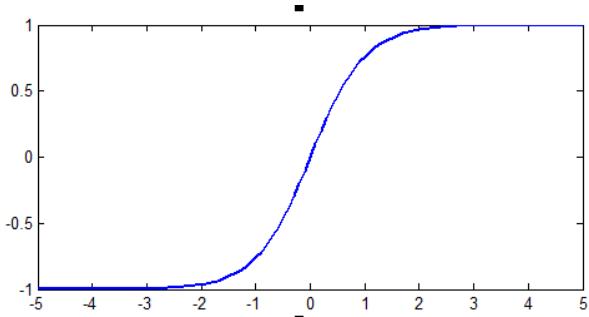
Fungsi ini memiliki *range* antara -1 sampai 1. Pada toolbox Matlab dikenal dengan nama *tansig* dan syntaxnya adalah $y = \text{tansig}(x)$, dirumuskan sebagai berikut:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.17)$$

dengan turunan fungsi tersebut adalah:

$$f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)] \quad (2.18)$$

Gambar 2.9 grafik fungsi sigmoid bipolar adalah sebagai berikut :



Gambar 2.9 Fungsi Sigmoid Bipolar (Sri Kusumadewi, 2004)

4. Aturan Pembelajaran

Proses pembelajaran bertujuan untuk melakukan pengaturan terhadap bobot pada jaringan syaraf, sehingga diperoleh bobot akhir yang sesuai dengan pola data yang dilatih. Pada proses ini akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu.

Rosenblatt memperkenalkan dua konsep pembelajaran, yaitu pembelajaran tak terawasi dan pembelajaran terawasi untuk melatih jaringan preceptron (Samarasinghe, 2007: 26).

a. Pembelajaran Terawasi (*Supervised Learning*)

Metode pembelajaran disebut terawasi jika *output* yang diharapkan sudah diketahui sebelumnya dan ini akan dijadikan target, sehingga masing-masing *input* memiliki pasangan *output* yang bersesuaian. Pada metode pembelajaran ini, satu pola *input* yang akan diberikan neuron pada pola *input*. Selanjutnya pola dirambatkan sepanjang syaraf sampai ke neuron pada lapisan *output*. Lapisan *output* akan membangkitkan pola *output* yang akan dicocokkan dengan pola target. *Error* akan terjadi jika ada perbedaan antara pola *output* pembelajaran dengan pola target.

b. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

Metode pembelajaran tak terawasi tidak membutuhkan target untuk keluarannya, sehingga tidak ada perbandingan untuk menentukan respon yang ideal. Proses ini hanya terdiri dari vektor-vektor masukan, dan berfungsi sebagai pengubah bobot jaringan untuk menghasilkan pola vektor, sehingga

penerapan dua vektor pelatihan suatu vektor lain cukup sejenis menghasilkan pola yang sama.

5. Validasi Silang

Validasi silang (*cross validation*) bertujuan untuk memastikan bahwa perfomansi jaringan tetap berada pada batasan yang diterima dengan berbedanya data *training* dan *testing*. Proses *training* terjadi pada saat pengaturan bobot dan bias. Sedangkan proses *testing* adalah proses ketelitian dari model yang telah diperoleh dari proses *training*. Terdapat beberapa komposisi pembagian data ke dalam data *training*, dan data *testing* yang sering digunakan, yaitu (Hota, et al, 2013: 165) :

- a. 80 % untuk data *training*, dan 20 % untuk data *testing*.
- b. 75 % untuk data *training*, dan 25 % untuk data *testing*.
- c. 60 % untuk data *training*, dan 40 % untuk data *testing*.

E. Pengukuran Ketepatan Klasifikasi

Agar hasil klasifikasi akurat maka dilakukan pengukuran ketepatan klasifikasi. Sensitivitas, spesifisitas, dan akurasi merupakan ukuran statistika ketepatan dari tes klasifikasi. Semua hasil klasifikasi memiliki tingkat kemungkinan, hasil klasifikasi bisa positif (sakit) atau negatif (tidak sakit). Untuk menggambarkan hasil klasifikasi yang mungkin diperlukan aturan, aturan itu adalah sebagai berikut (Lalkhen & McCluskey, 2004):

- a. *True Positive* (TP) : pasien memiliki penyakit dan hasil klasifikasi positif.

- b. *FalsePositive* (FP) : pasien tidak memiliki penyakit tetapi hasil klasifikasi positif.
- c. *True Negative* (TN) : pasien tidak memiliki penyakit dan hasil klasifikasi negatif.
- d. *False Negative* (FN) : pasien memiliki penyakit tetapi hasil klasifikasi negatif.

Setelah mendapatkan keempat tes itu maka sensitivitas, spesifisitas, dan akurasi dapat dihitung.

1. Sensitivitas

Sensitivitas didefinisikan sebagai proporsi orang dengan penyakit yang memiliki hasil klasifikasi positif. Tes dengan sensitivitas tinggi dapat dianggap indikator yang dapat diandalkan ketika hasil menunjukkan negatif. Misalnya, tes memiliki sensitivitas 100% berarti tes menunjukkan bahwa semua hasil klasifikasi sesuai dengan data orang yang memiliki penyakit. Sensitivitas memiliki rumus (Sharma & Sourabh, 2013: 332) :

$$sensitivitas = \frac{TP}{TP + FN} \times 100\% \quad (2.19)$$

2. Spesifisitas

Spesifisitas didefinisikan sebagai proporsi orang tanpa penyakit yang memiliki hasil klasifikasi negatif. Kebalikan dari sensitivitas tinggi dapat dianggap indikator yang diandalkan ketika hasilnya adalah positif. Spesifisitas memiliki rumus (Sharma & Sourabh, 2013: 332) :

$$spesifisitas = \frac{TN}{TN + FP} \times 100\% \quad (2.20)$$

3. Akurasi

Akurasi menunjukkan kedekatan hasil pengukuran dengan nilai sesungguhnya.

Akurasi 100% berarti bahwa hasil klasifikasi sama persis dengan data sesungguhnya. Semakin tinggi akurasi maka tes semakin baik. Akurasi dirumuskan dengan (Sharma & Sourabh, 2013: 332) :

$$\text{akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.21)$$

F. *Mean Square Error (MSE)*

Mean Square Error (MSE) adalah suatu nilai yang didapat dengan mengkuadratkan masing-masing *error* untuk masing-masing pengamatan dalam sebuah susunan data dan kemudian memperoleh rata-rata jumlah kuadrat tersebut.

Error adalah nilai yang diperoleh dari mengurangkan hasil klasifikasi terhadap nilai aktual. *Error* dari sebuah peramalan dapat dihitung dengan persamaan (Hanke dan Wichern, 2005:79):

$$e_l = Y_l - \hat{Y}_l \quad (2.22)$$

dengan

$l = 1, 2, \dots, r$

r : banyaknya data,

e_l : *error* klasifikasi pada saat data ke- l ,

Y_l : nilai aktual data ke- l ,

\hat{Y}_l : nilai klasifikasi dari Y_l .

Error yang diperoleh dari setiap data dapat digunakan untuk menghitung nilai MSE. Nilai MSE dapat dihitung dengan rumus:

$$MSE = \sum_{l=1}^r \frac{e_l^2}{r} \quad (2.23)$$

dengan,

e_l : *error* ke- l ,

r : banyaknya data.

BAB III

PEMBAHASAN

A. *Recurrent Neural Network* untuk Klasifikasi

1. Model *Recurrent Neural Network*

Jaringan Syaraf Tiruan merupakan model yang dapat digunakan untuk menyelesaikan permasalahan klasifikasi karena tidak memerlukan asumsi seperti yang terdapat pada metode parametrik, asumsi itu adalah asumsi normal dan homogenitas matrik kovarian. *Recurrent Neural Network* (RNN) merupakan salah satu model *artificial neural network* (ANN). RNN adalah jenis jaringan pada *neural network* dimana terdapat *loop* sebagai koneksi umpan balik dalam jaringan (Lin & Lee, 1996: 340). Jaringan *recurrent* adalah jaringan yang mengakomodasikan *output* jaringan untuk menjadi *input* pada jaringan tersebut dan selanjutnya menghasilkan output jaringan berikutnya.

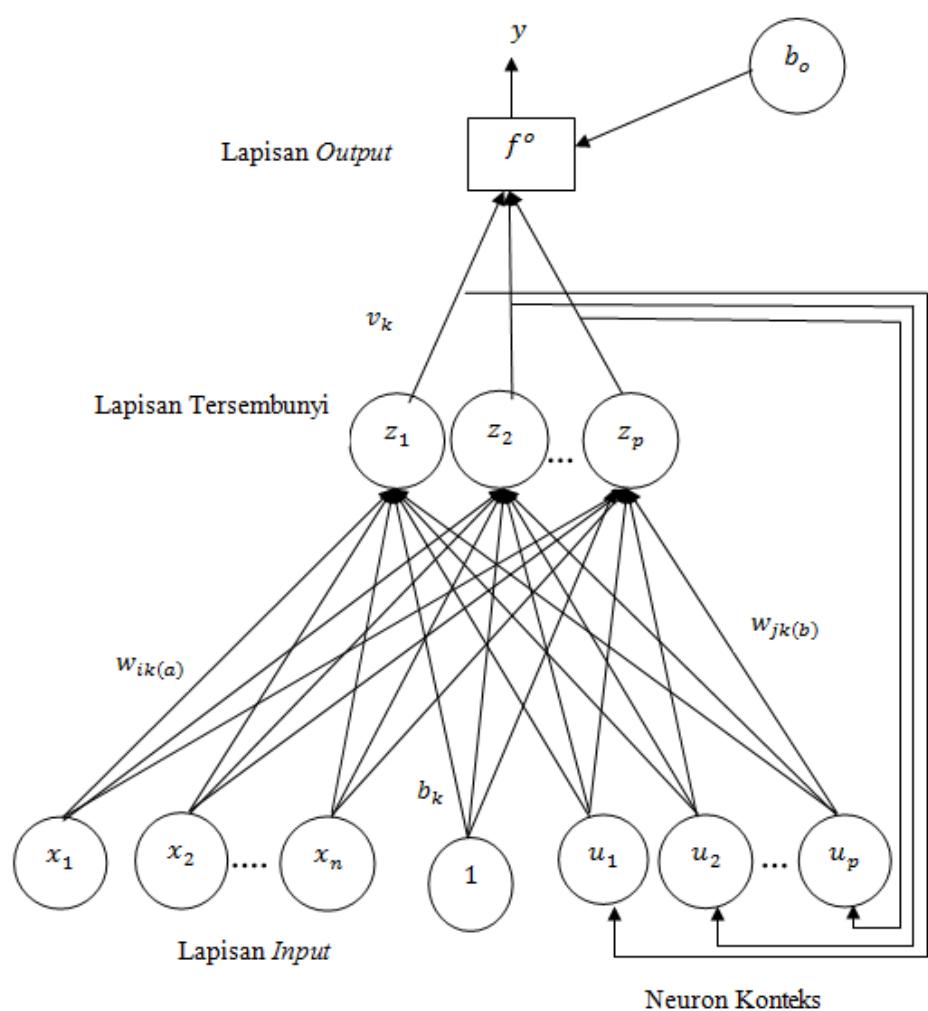
Model jaringan *recurrent* mirip dengan jaringan *feedforward* namun terdapat tambahan neuron pada lapisan *input* yang hanya menerima sinyal internal (umpan balik dari lapisan tersembunyi atau *output* dari lapisan *input*). Model RNN merupakan perbaikan dari model jaringan sebelumnya, yaitu *feedforward*. Keunikan dari jaringan *recurrent* adalah adanya koneksi umpan balik.

Terdapat dua macam jaringan RNN, yaitu jaringan Elman dan jaringan *Hopfield*. Pada skripsi ini akan dibahas mengenai jaringan

recurrent pada jaringan Elman. Jaringan Elman dapat digolongkan menjadi jaringan *recurrent* karena jaringan Elman memiliki koneksi umpan balik dari lapisan tersembunyi kembali pada lapisan *input*.

Jaringan Elman adalah jaringan *backpropogation* dua lapisan dengan neuron tambahan pada lapisan *input* yang berasal dari lapisan tersembunyi. Seperti penjelasan sebelumnya, bahwa jaringan Elman merupakan jaringan *recurrent* yang sederhana sehingga terdapat satu lapisan tersembunyi.

Gambar 3.1 arsitektur jaringan Elman adalah sebagai berikut :



Gambar 3.1 Arsitektur Model *Reccurent Neural Network*
Jaringan Elman

Model dari RNN jaringan Elman dengan menggunakan fungsi aktivasi sigmoid bipolar (2.12), secara matematis dapat dirumuskan dengan persamaan :

$$y = \sum_{k=1}^p v_k \frac{1 - \exp\left(-\left(\sum_{i=1}^n x_i w_{ik}(a) + \sum_{j=1}^p u_j w_{jk}(b) + b_k\right)\right)}{1 + \exp\left(-\left(\sum_{i=1}^n x_i w_{ik}(a) + \sum_{j=1}^p u_j w_{jk}(b) + b_k\right)\right)} + b_o \quad (3.1)$$

dengan,

- y = variabel *output*
- x_i = variabel *input* sebanyak n dengan $i = 1, 2, \dots, n$
- z_k = neuron lapisan tersembunyi dengan $k = 1, 2, \dots, p$
- u_j = variabel *input* yang berada pada neuron tambahan (*neuron konteks*) dengan $j = 1, 2, \dots, p$
- b_k = bobot bias pada neuron ke-k pada lapisan tersembunyi
- b_o = bobot bias pada neuron lapisan *output*
- $w_{ik}(a)$ = bobot dari lapisan *input* ke-i menuju neuron ke-k pada lapisan tersembunyi
- $w_{jk}(b)$ = bobot dari lapisan *input* (neuron tambahan) ke-j menuju neuron ke-k pada lapisan tersembunyi
- v_k = bobot dari neuron ke-k pada lapisan tersembunyi yang menuju lapisan *output* dengan $k = 1, 2, \dots, p$
- ε = *error*

2. Algoritma Pembelajaran *Recurrent Neural Network*

Dalam Jaringan Syaraf Tiruan terdapat proses pembelajaran sehingga jaringan-jaringan yang dibentuk dapat melakukan generalisasi karakteristik tingkah laku objek. Algoritma pembelajaran adalah pembentukan pemetaan (fungsi) yang menggambarkan hubungan antara vektor *input* ke vektor *output*.

Pembelajaran pada RNN menggunakan algoritma *backpropagation* sehingga fungsi aktivasi yang digunakan dalam proses pembelajaran harus sama dengan fungsi yang digunakan pada algoritma *backpropagation*. Tidak seperti *backpropagation*, pada jaringan Elman, fungsi aktivasi yang dapat digunakan adalah fungsi aktivasi linear, sigmoid biner, dan sigmoid bipolar.

Salah satu langkah awal pembelajaran RNN menggunakan algoritma *backpropagation* adalah menentukan parameter. Parameter-parameter yang harus diatur untuk pelatihan ini beserta kriteria penghentian iterasi ada delapan parameter, yaitu (Kusumadewi, 2004):

1) Maksimum epoch

Maksimum epoch adalah jumlah epoch maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai epoch melebihi maksimum epoch.

Instruksi: `net.trainParam.epochs = MaxEpoch`

Nilai default untuk maksimum epoch adalah 1500.

2) Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

Instruksi: `net.trainParam.goal = TargetError`

Nilai default untuk kinerja tujuan adalah 0.7.

3) Maksimum kenaikan kinerja

Maksimum kenaikan kinerja adalah nilai maksimum kenaikan *error* yang diijinkan, antara *error* saat ini dan *error* sebelumnya.

Instruksi: `net.trainParam.max_perf_inc = MaxPerfInc`

Nilai default untuk maksimum kenaikan kinerja adalah 1.06.

4) *Learning rate* (α)

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*.

Instruksi: `net.trainParam.lr = LearningRate`

Nilai default untuk *learning rate* adalah 0.1.

5) Rasio kenaikan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menaikkan *learning rate* apabila *learning rate* yang ada terlalu rendah untuk mencapai kekonvergenan.

Instruksi: `net.trainParam.lr_inc`

Nilai default untuk kenaikan *learning rate* adalah 1.2.

6) Rasio penurunan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menurunkan learning rate apabila learning rate yang ada terlalu tinggi untuk menuju ketidakstabilan.

Instruksi: `net.trainParam.lr_dec`

Nilai default untuk rasio untuk menurunkan learning rate adalah 0.6.

7) Jumlah epoch yang akan ditunjukkan kemajuannya

Menunjukkan jumlah epoch yang akan ditampilkan.

Instruksi: `net.trainParam.show = EpochShow`

Nilai default untuk jumlah epoch yang akan ditunjukkan adalah 100.

8) 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.

Instruksi: `net.trainParam.mc = Momentum`

Nilai default momentum adalah 0.75

Berikut adalah algoritma *backpropagation* yang digunakan dalam RNN pada jaringan Elman (Fausett, 1994: 294-296) :

Langkah 0 : Inisiasi bobot dengan mengambil bobot awal menggunakan nilai random terkecil

Langkah 1 : Menetapkan Parameter

: Inisiasi epoch=0, MSE=1

Fase I : *Feedforward*

Langkah 2 : Tiap-tiap neuron *input* ($x_i, i = 1, 2, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua neuron yang ada pada lapisan tersembunyi.

Langkah 3 : Tiap-tiap neuron pada lapisan tersembunyi ($z_k, k = 1, 2, \dots, p$) menjumlahkan sinyal-sinyal *input* berbobot :

$$z_in_k = \sum_{i=1}^n x_i w_{ik}(a) + b_{1k} \quad (3.2)$$

b_{1k} = bias pada unit tersembunyi k.

Fungsi aktivasi untuk menghitung sinyal *output* :

$$z_k = f(z_in_k) \quad (3.3)$$

z_k = sinyal *output* lapisan tersembunyi pada neuron k.

Sinyal dikirim ke semua neuron pada lapisan *output* dan neuron tambahan di lapisan *input*. Langkah ini dilakukan sebanyak lapisan tersembunyi.

Langkah 4 : Tiap-tiap neuron pada lapisan tersembunyi meneruskan sinyal pada neuron tambahan pada lapisan *input* $u_j, j = 1, 2, \dots, p$. Sinyal *output* neuron tambahan ke lapisan tersembunyi :

$$U_net_j = \sum_{j=1}^p u_j w_{jk}(b) + b_k \quad (3.4)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$U_j = f(U_net_j) \quad (3.5)$$

dan kirim sinyal tersebut ke lapisan tersembunyi.

Langkah 5 : Tiap-tiap neuron pada lapisan tersembunyi (z_k , $k = 1, 2, \dots, p$) menjumlahkan sinyal-sinyal *input* berbobot :

Sinyal *output* neuron tersembunyi ke lapisan *output* :

$$T_{\text{net}_k} = \sum_{i=1}^n x_i w_{ik(a)} + \sum_{j=1}^p u_j w_{jk(b)} + b_k \quad (3.6)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$T_k = f(T_{\text{net}_p}) \quad (3.7)$$

dan kirim sinyal tersebut ke semua neuron pada lapisan *output*.

Langkah 6 : Tiap-tiap neuron *output* (y) menjumlahkan sinyal-sinyal *input* berbobot, dengan rumus :

$$y_{\text{in}} = b_o + \sum_{k=1}^p (T_k) (v_{k1}) \quad (3.8)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* :

$$y = f(y_{\text{in}}) \quad (3.9)$$

kemudian sinyal dikirim ke semua unit di lapisan *output*.

Fase II : *Backpropagation*

Langkah 7 : Tiap-tiap *output* (y) menerima target pola yang berhubungan dengan pola *input* pembelajaran dengan *error*

$$\delta_k = (t - y) f'(y_{\text{in}}) \quad (3.10)$$

δ_k merupakan neuron *error* yang akan dipakai dalam perubahan bobot v_k . Hitung suku perubahan bobot v_k dengan laju pembelajaran α .

$$\Delta v_k = \alpha \cdot \delta_k \cdot z_k \quad (3.11)$$

Hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki b_0) :

$$\Delta b_0 = \alpha \delta_k \quad (3.12)$$

Langkah 8 : Tiap-tiap neuron tersembunyi ($z_k, j = 1, 2, \dots, p$) menjumlakan hasil perubahan (delta) *inputnya* dari neuron-neuron di lapisan atasnya

$$\delta_{in_k} = \sum_{k=1}^p \delta_k v_k \quad (3.13)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error* :

$$\delta_k = \delta_{in_k} f'(z_{in_k}) \quad (3.14)$$

kemudian hitung koreksi bobot (yang nantinya digunakan untuk memperbaiki nilai $w_{ik(a)}$ dan $w_{jk(b)}$) :

$$\Delta w_{ik(a)} = \alpha \delta_k x_i \quad (3.15)$$

$$\Delta w_{jk(b)} = \alpha \delta_k U_j \quad (3.16)$$

Hitung suku koreksi bias (yang akan dipakai nanti untuk merubah bobot b_k)

$$\Delta b_k = \alpha \delta_k \quad (3.17)$$

Fase III : Perubahan bobot

Langkah 9 : Hitung semua perubahan bobot. Perubahan bobot yang menuju ke *output* neuron:

$$v_{k(baru)} = v_{k(lama)} + \Delta v_k \quad (3.18)$$

$$b_{0(baru)} = b_{0(lama)} + \Delta b_0 \quad (3.19)$$

Tiap-tiap neuron tersembunyi ($z_k, k = 1, 2, \dots, p$)

memperbaiki bias dan bobotnya sebagai berikut :

$$w_{ik(a)}(\text{baru}) = w_{ik(a)}(\text{lama}) + \Delta w_{ik(a)} \quad (3.20)$$

$$w_{jk(b)}(\text{baru}) = w_{jk(b)}(\text{lama}) + \Delta w_{jk(b)} \quad (3.21)$$

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

Langkah 10 : Selesai

3. Membangun Model *Recurrent Neural Network* pada Jaringan Elman

Pada dasarnya membangun sebuah jaringan baik, jaringan Elman ataupun jaringan lainnya diawali dengan menentukan *input* jaringan. Selanjutnya mengestimasi neuron yang terletak pada lapisan tersembunyi. Pada Matlab, untuk membangun jaringan Elman dengan menggunakan instruksi newelm.

Fungsi yang digunakan adalah (Kusumadewi: 2004: 327) :

net = newelm(Pn, [S1 S2 ... Sn], {TF1 TF2 ... TFn1}, BTF, BLF, PF)

dengan

Pn : matriks berukuran $nx2$ yang berisi nilai minimum dan maksimum, dengan n adalah banyak variabel *input*.

Si : banyak neuron pada lapisan ke-i, dengan $i = 1, 2, \dots, n$.

TF : fungsi aktivasi pada lapisan ke-r, dengan $t = 1, 2, \dots, N1$ (default: *tansig*).

BTF : fungsi pelatihan jaringan (default: *trainlm*).

BLF : fungsi pelatihan untuk bobot (default: *learngdm*).

PF : fungsi kinerja (default: *mse*).

4. Prosedur Pemodelan *Recurrent Neural Network* pada Jaringan Elman

Langkah dalam mendapatkan model menggunakan jaringan Elman adalah menentukan *input* yang sesuai yang didapat dari ekstraksi fitur hasil *colonoscopy*. Setelah *input* didapat maka data dibagi menjadi data pembelajaran (*training*) dan data pengujian (*testing*). Proses pembelajaran yang dilakukan data *training* berdasarkan nilai MSE dan akurasi data. Selanjutnya dilakukan pembelajaran jaringan dengan langkah sebagai berikut :

a. Normalisasi Data

Sebelum dilakukan proses pembelajaran, data *input* dan target *output* harus dinormalisasi terlebih dahulu. Normalisasi adalah penskalaan terhadap nilai-nilai *input* dan target *output* sedemikian hingga data-data *input* dan target *output* tersebut masuk dalam suatu *range* tertentu. Hal ini dilakukan agar nilai *input* dan target *output* sesuai dengan *range* dari fungsi aktivasi yang digunakan dalam jaringan. Proses normalisasi dapat dilakukan dengan bantuan *mean* dan standar deviasi (Samarasinghe, 2007: 253).

1) Perhitungan nilai rata-rata:

$$\bar{x} = \frac{1}{r} \sum_{l=1}^r x_l \quad (3.23)$$

2) Perhitungan nilai varians:

$$s^2 = \frac{1}{r-1} \sum_{l=1}^r (x_l - \bar{x})^2 \quad (3.24)$$

3) Perhitungan normalisasi

$$x_{\text{normalisasi}} = \frac{x_l - \bar{x}}{s} \quad (3.25)$$

dengan,

x_l : data *input* , $l = 1, 2, \dots, r$

r : banyaknya data,

Matlab menyediakan fungsi *prestd* untuk melakukan normalisasi, dengan syntax:

[pn,meanp,stdp,tn,meant,stdt]=prestd(P,T)

dengan

P : matriks *input*

T : matriks target

pn : matriks *input* yang ternormalisasi

tn : matriks target yang ternormalisasi

meanp : *mean* pada matriks *input* asli

stdp : standar deviasi pada matriks *input* asli

meant : *mean* pada matriks target asli

stdt : standar deviasi pada matriks target asli

b. Pembentukan Model Terbaik

Pembentukan model terbaik meliputi dua langkah penting yaitu penentuan banyak neuron lapisan tersembunyi dan melakukan eliminasi variabel *input*. Penentuan banyak neuron lapisan tersembunyi dilakukan dengan cara *trial* dan *error* beberapa model yang mungkin, kemudian membandingkan nilai *Mean Square Error* (MSE) dan akurasi pada beberapa model. Setelah mendapatkan banyak neuron tersembunyi yang sesuai, yaitu model yang memiliki nilai MSE terkecil dengan prinsip parsimoni (model terbaik adalah model dengan MSE terkecil dengan jaringan paling sederhana) dan akurasi tinggi maka langkah selanjutnya adalah eliminasi *input*. Masing-masing *input* dieliminasi dari model kemudian didapat model

dengan nilai MSE terkecil dan akurasi tinggi, maka model tersebut adalah model terbaik. Pemilihan model terbaik dan eliminasi *input* dilakukan dengan melakukan pembelajaran terhadap model-model yang mungkin menggunakan algoritma pembelajaran *backpropagation*.

Seperti halnya pada jaringan *feedforward*, pembelajaran dilakukan dalam rangka melakukan pengaturan bobot. Pengaturan ini dimaksudkan untuk memperoleh bobot-bobot, MSE, dan nilai akurasi yang baik pada akhir pembelajaran. Selama proses pembelajaran, bobot-bobot diatur secara iteratif untuk meminimumkan fungsi kinerja jaringan. Fungsi kinerja yang sering digunakan untuk *backpropagation* adalah MSE, nilai MSE dihitung dari rata-rata kuadrat *error* yang terjadi antara *output* jaringan dan target.

Dalam rangka meminimumkan kinerja untuk mengatur bobot-bobot digunakan algoritma *gradient descent*. Pada Matlab, ada dua cara untuk mengimplementasikan algoritma *gradient descent*, yaitu dengan *incremental mode* dan *batch mode*. Dalam penelitian ini akan dijelaskan lebih lanjut mengenai *batch mode*. Pada *batch mode*, penghitungan gradien penghitungan gradien dan perbaikan nilai bobot-bobot dilakukan setelah pengoperasian semua *input* data.

Untuk menggunakan pelatihan *backpropagation* dengan *batch mode*, digunakan fungsi *train*.

Syntax: [net,tr] = train(net,P,T,Pi,Ai)

dengan,

net : jaringan syaraf

tr : informasi pelatihan (epoch dan fungsi kinerja)

P : matriks data input

T : matriks data target (default: 0)
Pi : kondisi awal delay input (default: 0)
Ai : kondisi awal delay lapisan (default: 0)

Fungsi ini akan menggunakan objek jaringan, kumpulan data input, dan target sebagai input pembelajaran, dan akan menghasilkan objek jaringan terlatih, bobot-bobot akhir, dan informasi selama pembelajaran (*epoch* dan fungsi kinerja) sebagai nilai output.

Ada beberapa fungsi pembelajaran untuk bobot-bobot yang menggunakan *gradient descent*, yaitu *gradient descent* (*traingd*) dan *gradient descent* dengan *momentum* (*traingdm*). Proses pembelajaran *traind* dan *traingdm*, biasanya berjalan cukup lambat. Untuk lebih mempercepat proses pembelajaran kedua fungsi gradien diperbaiki menjadi fungsi *gradient decent* dengan *momentum* dan *adaptive learning rate* (*traingdx*), sehingga dalam penelitian ini akan digunakan fungsi *traingdx*. Fungsi ini akan memperbaiki bobot-bobot berdasarkan *gradient descent* dengan *learning rate* yang bersifat *adaptive*. Hal ini berarti nilai *learning rate* akan diubah selama proses pelatihan untuk menjaga agar algoritma senantiasa stabil selama proses pelatihan. Selain itu, fungsi *traingdx* juga menggunakan momentum selama proses pelatihan.

c. Denormalisasi

Setelah proses pembelajaran selesai, nilai ternormalisasi output jaringan harus dikembalikan (denormalisasi) ke nilai sebenarnya. Proses ini disebut denormalisasi atau *postprocessing*. Proses ini penting karena pada awal pembelajaran telah dilakukan normalisasi data asli, maka *output*

jaringan memiliki *mean*=0 dan standar deviasi=1. Matlab menyediakan fungsi *poststd* untuk melakukan denormalisasi, dengan *syntax*:

```
[P,T]=poststd(pn,meanp,stdp,tn,meant,stdt)
```

P dan T adalah matriks yang telah didenormalisasi.

Sebelum dilakukan proses denormalisasi, perlu dilakukan simulasi pada jaringan syaraf dengan *syntax* :

```
an = sim(net,pn);
```

```
a = poststd(an,meant,stdt).
```

B. Penerapan Model *Recurrent Neural Network* pada Jaringan Elman untuk Klasifikasi Stadium Kanker Kolorektal

Recurrent Neural Network (RNN) digunakan untuk menentukan klasifikasi stadium pengelompokan penyakit kanker kolorektal, dengan kata lain untuk menentukan apakah data termasuk normal, kanker kolorektal stadium I, stadium II, stadium III, atau stadium IV. Data yang digunakan adalah data sekunder berupa foto *colonoscopy* kolon atau rektum. Foto hasil *colonoscopy* tersebut sudah digolongkan berdasarkan stadiumnya. Diperoleh 92 data foto hasil *colonoscopy*, data ini didapat dari dokumentasi kesehatan yang ada pada :

- a. *Vision and Image Analysis Group*, Cornell University (Murra-Saca),
- b. *Endoskopie Atlas*, Albertinen-Krankenhaus (Guntau, 2008).

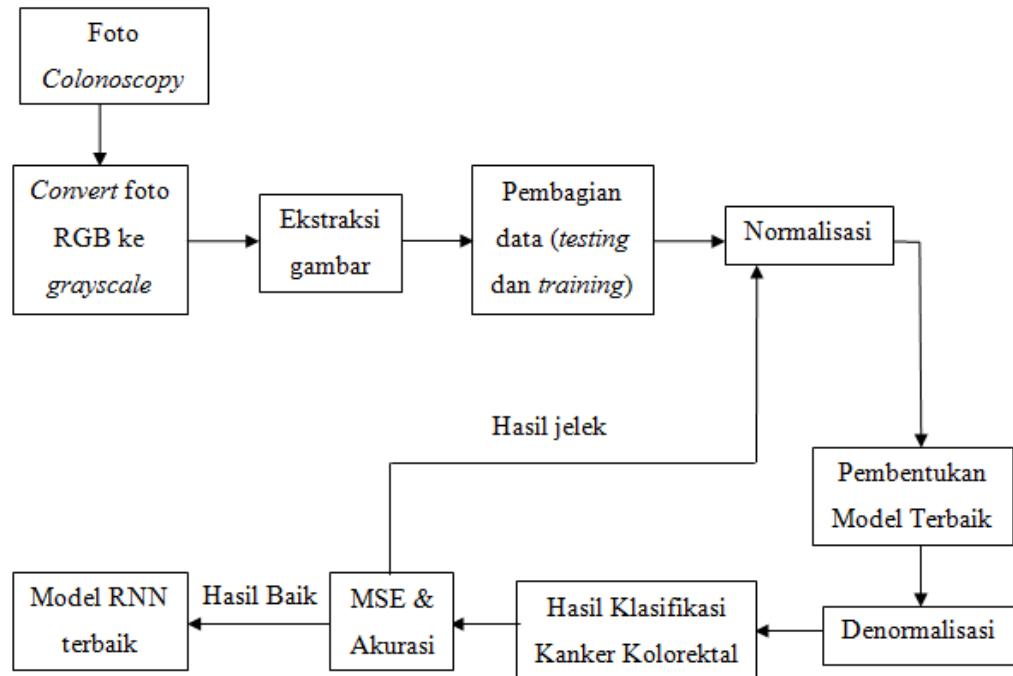
Data yang didapatkan adalah data dengan citra warna atau *rgb* (*red green blue*). Sebelum diklasifikasikan menggunakan model RNN data foto *colonoscopy* tersebut akan diubah dahulu dari *rgb* ke citra *grayscale* yang selanjutnya

dilakukan ekstraksi gambar (*feature extraction*), sehingga diperoleh variabel *input* yang akan digunakan untuk dalam klasifikasi. Proses ekstraksi gambar dilakukan dengan menggunakan MATLAB dengan Image *Processing Toolbox* dan sistem *prototype* yang didesain menggunakan *Graphical User Interface* (GUI).

Setelah variabel *input* didapat maka dapat dilakukan pembentukan model RNN dengan membentuk struktur jaringan. Langkah pertama dalam pembentukan struktur jaringan Elman pada RNN adalah membagi data menjadi data *training* dan data *testing*. Setelah pembagian data dilakukan pembentukan model terbaik. Pembentukan model terbaik diawali dengan menentukan banyaknya lapisan tersembunyi serta banyak neuron yang ada pada lapisan *input* dan lapisan tersembunyi.

Struktur jaringan Elman pada *Recurrent Neural Network* pada penelitian ini menggunakan 1 lapisan tersembunyi. Banyak neuron *input* dan neuron tersembunyi ditentukan dengan cara *trial* dan *error* beberapa model yang mungkin, kemudian membandingkan nilai akurasi pada beberapa model tersebut. Setelah mendapatkan banyak neuron tersembunyi yang sesuai, yaitu model yang memiliki nilai akurasi tinggi maka langkah selanjutnya adalah eliminasi *input*. *Input* dengan nilai akurasi tinggi adalah input yang optimal.

Secara singkat klasifikasi penyakit kanker kolorektal menggunakan model *Recurrent Neural Network* digambarkan pada Gambar 3.2.



Gambar 3.2. Tahap-Tahap Pemodelan *Recurrent Neural Network*

C. Ekstraksi Gambar

Data yang digunakan dalam penelitian ini adalah data dalam bentuk gambar foto *colonoscopy*. Maka untuk mendapatkan variabel *input* perlu dilakukan ekstraksi gambar. Sebelum dilakukan ekstraksi gambar perlu dilakukan pengubahan (*convert*) citra gambar dari *rgb* menjadi *grayscale*, setelah itu proses ekstraksi gambar dapat dilakukan dengan *Gray Level Cooccurrence Matrix* (GLCM).

1. Mengubah Tipe Gambar

Gambar foto *colonoscopy* tersebut adalah gambar dengan citra warna atau *rgb* (*red green blue*). Warna dalam gambar dideskripsikan oleh banyaknya warna

merah, hijau, dan biru sehingga citra warna dipandang sebagai penumpukan tiga matriks yang mewakili masing-masing warna setiap piksel.

Dalam penelitian ini, untuk memperoleh sifat-sifat dari gambar maka gambar harus di konversi ke dalam dua matrik saja, sehingga gambar citra rgb akan diubah menjadi *grayscale* menggunakan MATLAB. Perintah pada MATLAB yang digunakan adalah sebagai berikut :

```
k=imread('F:/kanker/stage 4/e2.jpg'); { digunakan untuk membaca gambar}  
l=rgb2gray(k); {digunakan untuk mengubah citra gambar rgb ke grayscale}  
imshow(k),figure,imshow(l) {untuk memunculkan gambar k dan l}
```

Berikut Gambar 3.3 mengenai fitur RGB yang dikonversi ke *Grayscale*



Gambar 3.3 Gambar Asli (RGB) dan Tipe *Grayscale*

Seluruh gambar foto *colonoscopy* dalam citra rgb beserta hasil konversi dalam citra *grayscale* data *training* dapat dilihat pada lampiran 1 dan untuk data *testing* dapat dilihat pada lampiran 2.

2. *Gray Level Cooccurrence Matrix (GLCM)*

Setelah dilakukan pengubahan gambar dengan citra rgb ke citra *grayscale* maka dilakukan ekstraksi gambar menggunakan *Gray Level Cooccurrence Matrix* (GLCM). Proses GLCM adalah metode pengekstrak gambar pda orde kedua dalam analisis statistika gambar, hal ini berarti GLCM dapat bekerja pada matriks

dua dimensi. Proses GLCM mengkonversi gambar ke fitur *Grayscale*, fitur ini didapat dari rata-rata intensitas warna pada fitur RGB. Proses GLCM akan menghasilkan variabel yang nantinya digunakan sebagai *input* dalam proses klasifikasi. Hanya terdapat beberapa fungsi GLCM yang pada MATLAB untuk mendapatkan variabel, diantaranya adalah energi, kontras, korelasi, *entropy*, dan homogenitas. Perintah untuk mendapatkan sifat-sifat tersebut adalah sebagai berikut :

```
l=rgb2gray(k);
Entropy=entropy(l)
glcm=graycomatrix(l);
stats=graycoprops(glcm,'all')
maka keluaran dari perintah di atas, yaitu
```

Entropy: 7.6222

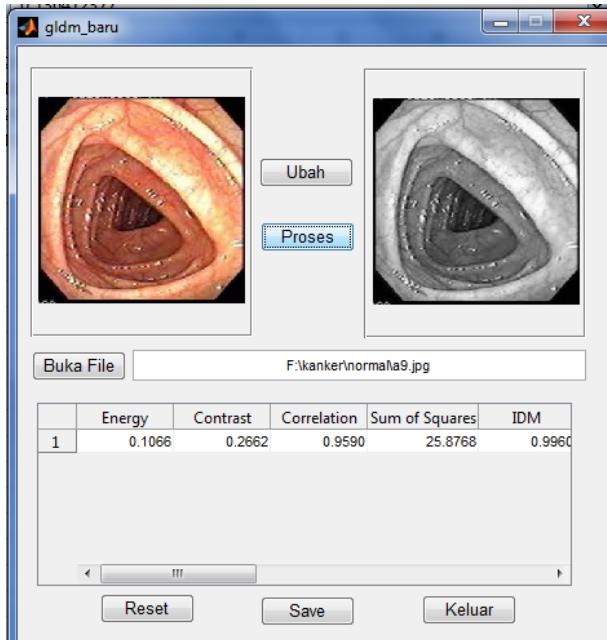
Contrast: 0.3141

Correlation: 0.9287

Energy: 0.1137

Homogeneity: 0.8710

Terdapat 28 fitur pada yang dapat diperoleh pada ekstraksi gambar (Harralick, 1973: 18). Dalam penelitian ini, akan digunakan 14 fitur dengan 5 fitur yang sudah terdapat pada perintah MATLAB, sedangkan 9 fitur lainnya didapat dengan membuat Mfile pada MATLAB (lampiran 3). Untuk memudahkan dan mempercepat proses pengubahan (*convert*) gambar serta ekstraksi gambar maka dibangun *Graphical User Interface* (GUI) pada MATLAB atau biasa dikenal sebagai GUIDE (GUI Builder). Program GUI dibuat dengan menggunakan kode Mfile yang terdapat pada lampiran 4, dan program GUI tersebut akan menghasilkan tampilan seperti gambar 3.4.



Gambar 3.4 Tampilan GUI dalam Mengubah Gambar & Ekstraksi Gambar

D. Klasifikasi *Recurrent Neural Network*

Data foto hasil *colonoscopy* yang diperoleh digunakan untuk proses klasifikasi, terdapat 92 data yang sudah diketahui terlebih dahulu stadiumnya. Stadium pada gambar tersebut nantinya akan dijadikan target, sedangkan hasil ekstraksi gambar dijadikan sebagai variabel *input*. Terdapat 14 fitur hasil ekstraksi yang akan dijadikan sebagai variabel *input*. Data 14 fitur hasil ekstraksi beserta targetnya dapat dilihat pada lampiran 5 (untuk data *training*) dan lampiran 6 (untuk data *testing*) yang nantinya digunakan dalam proses klasifikasi. Klasifikasi stadium kanker kolorektal tersebut menggunakan model Elman *Recurrent Neural Network* (RNN). Adapun langkah-langkah untuk klasifikasi menggunakan model Elman RNN, yaitu :

1. Menentukan Variabel

a. Variabel *Input*

Variabel *input* terdiri dari 14 fitur hasil ekstraksi yang merupakan variabel yang dominan dalam klasifikasi menggunakan foto hasil *colonoscopy*, yaitu:

- 1) x_1 (Energi)
- 2) x_2 (Kontras)
- 3) x_3 (Korelasi)
- 4) x_4 (*Sum of Squares*)
- 5) x_5 (*Inverse Difference Moment* (IDM))
- 6) x_6 (*Sum Average*)
- 7) x_7 (*Sum Variance*)
- 8) x_8 (*Sum Entropy*)
- 9) x_9 (*Entropy*)
- 10) x_{10} (*Differential Variance*)
- 11) x_{11} (*Differential Entropy*)
- 12) x_{12} (Maksimum Probabilitas)
- 13) x_{13} (Homogenitas)
- 14) x_{14} (*Dissimilarity*)

Setelah dilakukan ekstraksi gambar, terdapat 14 fitur hasil ekstraksi yang dijadikan variabel *input*. Tetapi terdapat fitur yang menghasilkan nilai yang sama, yaitu X2 (kontras) dan X10 (*differential variance*). Hal ini menyebabkan salah satu dari fitur tersebut harus dieliminasi terlebih dahulu, maka X10 dieliminasi sehingga tersisa 13 fitur yang dijadikan *input*.

b. Variabel *Output*

Variabel *output* berupa data numerik, dan *output* yang diharapkan berupa angka 1 untuk normal (tidak terkena kanker kolorektal), 2 untuk kanker kolorektal stadium I, 3 untuk kanker kolorektal stadium II, 4 untuk kanker kolorektal stadium III, dan 5 untuk kanker kolorektal stadium IV.

Dalam mengklasifikasi stadium kanker kolorektal, terdapat beberapa kriteria pengelompokan, yaitu:

- 1) Jika *output* (y) bernilai $0,5 \leq y < 1,5$ maka dianggap 1, sehingga hasil klasifikasi normal.
- 2) Jika *output* (y) bernilai $1,5 \leq y < 2,5$ maka dianggap 2, sehingga hasil klasifikasi stadium I.
- 3) Jika *output* (y) bernilai $2,5 \leq y < 3,5$ maka dianggap 3, sehingga hasil klasifikasi stadium II.
- 4) Jika *output* (y) bernilai $3,5 \leq y < 4,5$ maka dianggap 4, sehingga hasil klasifikasi stadium III.
- 5) Jika *output* (y) bernilai $4,5 \leq y < 5,5$ maka dianggap 5, sehingga hasil klasifikasi stadium IV.

2. Pembagian Data

Data dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Pada penelitian ini digunakan komposisi 80% (72 data) untuk data *training* dan 20% (18 data) untuk data *testing*. Pembagian data tersebut tidak bulat dikarenakan total keseluruhan data adalah 92 data.

Komposisi pembagian data ini didasarkan pada nilai MSE dan akurasi antara data *training* dan data *testing*. Komposisi pembagian data ini menghasilkan nilai MSE dan akurasi yang lebih kecil.

3. Perancangan Model

Perancangan model diperoleh melalui hasil pembelajaran pada data *training* yang akan menghasilkan model terbaik. Sebelum perancangan model, data perlu dinormalisasi agar data berdistribusi normal. Proses normalisasi data menggunakan bantuan *Matlab R2009a*. Setelah data berdistribusi normal, dilakukan perancangan model terbaik. Pemilihan model terbaik dengan melihat nilai MSE dan akurasi hasil pembelajaran. Proses pembelajaran dilakukan menggunakan algoritma *gradient descent* dengan fungsi pembelajaran bobot-bobot *trainidx*, variabel 13 *input*, 1 lapisan tersembunyi, dan 1 *output*. Fungsi aktivasi yang digunakan adalah *sigmoid bipolar* (*tansig*) pada lapisan tersembunyi, dan fungsi linear (*purelin*) pada lapisan *output*. Prosedur penentuan model terbaik, adalah sebagai berikut:

a. Menentukan Neuron Terbaik

Proses pembelajaran (lampiran 7) diawali dengan pencarian banyak neuron pada lapisan tersembunyi berdasarkan nilai MSE dan akurasi dengan variabel *input* $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{11}, x_{12}, x_{13}, x_{14}$ serta target *output*. Proses pembelajaran RNN dengan algoritma *backpropagation* . Hasil nilai MSE dan akurasi dari proses pembelajaran terdapat pada tabel 3.1.

Tabel 3.1. Nilai MSE dan Akurasi Hasil Pembelajaran

Banyak Neuron	MSE		Akurasi	
	Training	Testing	Training	Testing
1	1,2074*	2,2709*	77%*	94%*
2	1,1513	1,8410	78%	83%
3	0,5970	3,4842	90,5%	67%
4	0,3493	3,5664	91,9%	83%
5	0,4274	2,9374	89,3%	67,7%
6	0,2094	3,5204	95,9%	61,1%
7	0,2029	4,9088	94,59%	56%
8	0,1464	2,8101	98,65%	84,2%
9	0,1842	3,1821	94,5%	83,3%

Keterangan : *) Model terbaik

Pada Tabel 3.1 terlihat bahwa model terbaik adalah model dengan banyak neuron 1. Hal ini sesuai dengan prinsip parsimoni dan dengan membandingkan akurasi pada data *testing*. Prinsip parsimoni adalah memilih model yang mempunyai nilai MSE kecil dan akurasi tinggi dengan jaringan yang sederhana. Hal ini berarti, jika terdapat dua model yang menghasilkan nilai MSE dan akurasi yang hampir sama, maka model yang dipilih adalah model dengan arsitektur yang paling sederhana.

b. Eliminasi *Input* Jaringan

Setelah didapat neuron terbaik untuk model, maka tahap selanjutnya adalah eliminasi *input* untuk mendapatkan model yang lebih sederhana. Eliminasi *input* dilakukan dengan cara melakukan pembelajaran dengan mengeliminasi *input* jaringan (dengan banyak neuron tersembunyi 1) dan membandingkan dan MSE serta akurasi. Proses pembelajaran dilakukan dengan bantuan *Matlab R2009a*. Tabel 3.2 berikut adalah nilai MSE dan akurasi hasil eliminasi *input*.

Tabel 3.2. Nilai MSE dan Akurasi Hasil Eliminasi

No	Eliminasi	MSE		Akurasi	
		Training	Testing	Training	Testing
1	-	1,207	2,271	77%	94%
2	x_{11}	1,113	1,232	79%	94%
2	x_{11}, x_6	0,802	0,494	81%	94%
3	x_{11}, x_6, x_8	0,598	0,154	85%	94%
7	x_{11}, x_6, x_8, x_4^*	0,282*	0,0931*	92%*	94%*

Keterangan : *) Model terbaik

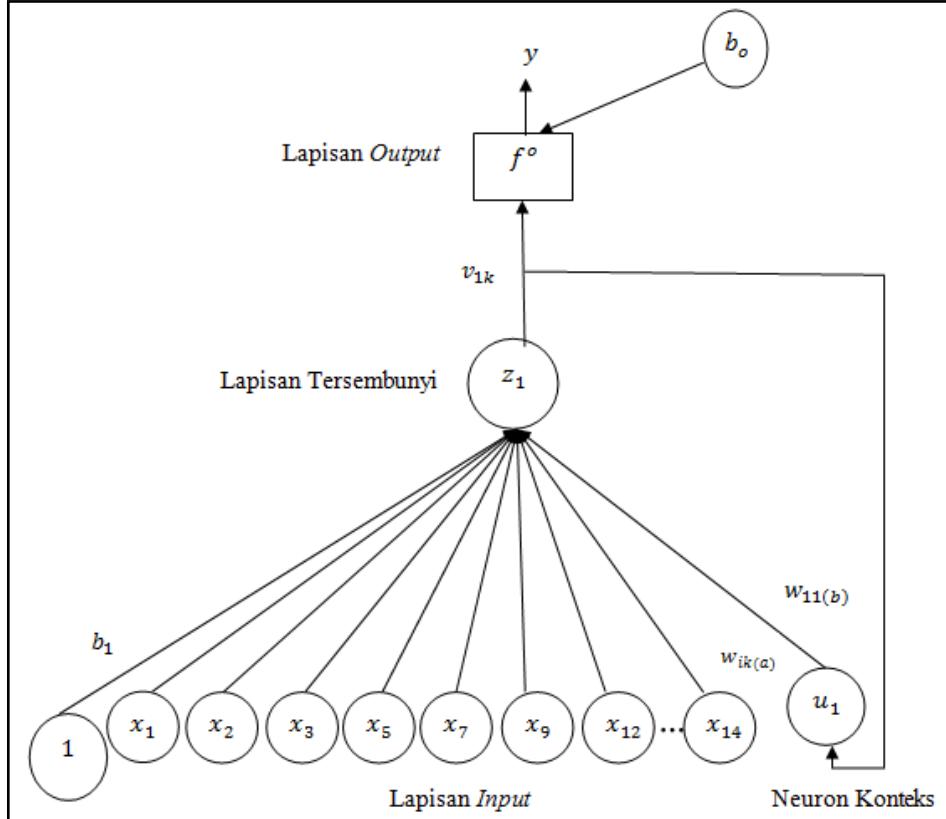
Berdarkan Tabel 3.2, dengan 1 neuron pada lapisan tersembunyi didapatkan model terbaik yaitu model jaringan yang memiliki 9 *input* $x_1, x_2, x_3, x_5, x_7, x_9, x_{12}, x_{13}, x_{14}$. Tabel 3.2 merupakan perbandingan nilai MSE dan akurasi dari eliminasi *input* yang memiliki nilai MSE dan akurasi yang terbaik. Untuk memastikan bahwa 9 *input* tersebut merupakan model terbaik, maka dilakukan lagi tahapan eliminasi *input*. Hasil MSE dan akurasi pada tahapan eliminasi *input* dapat dilihat pada lampiran 8.

Dari hasil pembelajaran model terbaik, diperoleh model terbaik dengan 1 neuron pada lapisan tersembunyi dan 9 *input* $x_1, x_2, x_3, x_5, x_7, x_9, x_{12}, x_{13}, x_{14}$. Proses pembelajaran ini akan menghasilkan *output* jaringan. *Output* tersebut masih dalam bentuk normal, sehingga untuk membawa *output* ke dalam bentuk semula *output* harus didenormalisasi dengan fungsi *poststd* pada MATLAB. Nilai target (stadium asli), *output*, *error*, dan klasifikasi data *training* terdapat pada lampiran 9, sedangkan untuk data *testing* terdapat pada lampiran 10.

Jadi jaringan RNN yang terpilih dan digunakan sebagai model estimasi memiliki arsitektur jaringan dengan 9 *input* $x_1, x_2, x_3, x_5, x_7, x_9, x_{12}, x_{13}, x_{14}$, 1 lapisan tersembunyi dengan 1 *neuron* dan 1 unit *output*. Struktur jaringan ini terpilih karena pada proses pembelajaran terhadap data *training* maupun data *testing*

menghasilkan MSE paling kecil pada saat jaringan menggunakan 1 neuron pada lapisan tersembunyi.

Gambar 3.5 arsitektur jaringan Elman RNN yang dibangun adalah sebagai berikut



Gambar 3.5 Arsitektur Jaringan Elman RNN Model Terbaik

Persamaan matematis model tersebut adalah

$$y = v_1 \frac{1 - \exp\left(-(\sum x w_{1(a)} + u_1 w_{11(b)} + b_1)\right)}{1 + \exp\left(-(\sum x w_{1(a)} + u_1 w_{11(b)} + b_1)\right)} + b_0 \quad (3.26)$$

dengan,

$$\begin{aligned} \sum x w_{1(a)} = & x_1 w_{11(a)} + x_2 w_{21(a)} + x_3 w_{31(a)} + x_5 w_{51(a)} + x_7 w_{71(a)} + x_9 w_{91(a)} + \\ & x_{12} w_{12,1(a)} + x_{13} w_{13,1(a)} + x_{14} w_{14,1(a)} \end{aligned}$$

Model RNN merupakan model dengan koneksi umpan balik dari neuron tersembunyi menuju neuron tambahan pada lapisan *input*. Dari neuron tambahan

itu, diperoleh dua bobot tambahan yaitu bobot dari neuron tambahan menuju neuron tersembunyi dan dari neuron tambahan menuju neuron pada lapisan *output*. Operasi keluaran lapisan tambahan ke lapisan tersembunyi dapat dihitung dengan persamaan berikut:

$$u_k = \frac{1 - \exp\left(-\left(\sum x w_{1(a)} + b_1\right)\right)}{1 + \exp\left(-\left(\sum x w_{1(a)} + b_1\right)\right)} \quad (3.27)$$

dan operasi keluaran dari neuron tambahan menuju ke lapisan *output* dapat dihitung dengan persamaan:

$$T_k = \frac{1 - \exp\left(-\left(\sum x w_{1(a)} + u_1 w_{11(b)} + b_1\right)\right)}{1 + \exp\left(-\left(\sum x w_{1(a)} + u_1 w_{11(b)} + b_1\right)\right)} \quad (3.28)$$

Selanjutnya dari persamaan (3.27) dan persamaan (3.28), dapat digunakan untuk menghitung *output* jaringan. *Output* jaringan dapat dihitung dengan persamaan:

$$y = v_{11} T_1 + b_0 \quad (3.29)$$

Setelah mendapatkan model terbaik, akan diklasifikasikan stadium kanker kolorektal menggunakan persamaan (3.27). Menggunakan salah satu data *training* yang ada pada nomor 71. Berdasarkan struktur jaringan Elman RNN dengan algoritma *backpropagation* menggunakan program MATLAB, diperoleh bobot-bobot sebagai berikut:

$$b_1 = 0,742$$

$$b_0 = 0,49456$$

$$w_{11(b)} = 0,5869$$

$$w_{i1(a)}$$

$$= [1,182 \quad -1.783 \quad -0.333 \quad -1.402 \quad -0.092 \quad 1.657 \quad -0.084 \quad 0.349 \quad 1.193]$$

$$v_k = [-1,488]$$

data *input* yang digunakan adalah yang sudah dinormalisasi (lampiran 11), yaitu

$x_{normalisasi}$

$$= [1,7334 \quad -1.0991 \quad 0.2511 \quad 1.4143 \quad -2.1819 \quad -1.7306 \quad 1,5283 \quad 1,2318 \quad 1.2199]$$

Operasi keluaran lapisan *input* ke lapisan tersembunyi

$$\begin{aligned} u_{net_1} &= x_1 w_{11(a)} + x_2 w_{21(a)} + x_3 w_{31(a)} + x_5 w_{41(a)} + x_7 w_{51(a)} + x_9 w_{61(a)} \\ &\quad + x_{12} w_{71(a)} + x_{13} w_{81(a)} + x_{14} w_{91(a)} + b_1 \\ &= (1,7334)(1,182) + (-1.0991)(-1.783) + (0.2511)(-0.333) \\ &\quad + (1.414)(-1.402) + (-2.182)(-0.092) + (-1.731)(1.657) \\ &\quad + (1,528)(-0.084) + (1,232)(0.349) + (1.219)(1.193) + 0,742 \\ &= -0,71431 \end{aligned}$$

Sinyal terbobot u_{net_1} kemudian diaktivasi dengan fungsi aktivasi

$$u_1 = f(u_{net_1}) = \frac{1 - e^{-(0,71431)}}{1 + e^{-(0,71431)}} = -0,34271$$

Kemudian dengan menggunakan persamaan (3.28), yaitu operasi keluaran neuron pada lapisan tersembunyi dengan neuron tambahan menuju ke lapisan *output* :

$$T_{net_1} = u_{net_1} + u_1 w_{11(b)} = (-0,7143) + (-0,34271)(0,5869) = -0,65691$$

$$T_1 = f(T_{net_1}) = \frac{1 - e^{(-0,65691)}}{1 + e^{(-0,65691)}} = 0,67964$$

Selanjutnya dapat dicari nilai *output* berdasarkan persamaan (3.29). Operasi pada lapisan *output*:

$$y = v_{11} T_1 + b_0 = (-1,488)(0,67964) + (0,49456) = 0,505$$

Diperoleh nilai $y^* = 0,141$. Nilai tersebut masih dalam bentuk normal. Untuk mengembalikan dalam bentuk data semula, digunakan fungsi *poststd* pada *matlab*. Setelah dilakukan denormalisasi, didapat nilai $y = 4,6528$ dan dibulatkan menjadi

5. Sehingga hasil klasifikasi menunjukkan penderita terkena kanker kolorektal stadium V. Untuk *output* dalam bentuk normal atau belum didenormalisasi terdapat pada lampiran 11.

4. Pengukuran Ketepatan Klasifikasi

Hasil klasifikasi pasti memiliki tingkat kesalahan dan terkadang salah mengidentifikasi kolon normal menjadi kolon yang terkena penyakit, atau sebaliknya. Berdasarkan hal itu perlu dilakukan perhitungan akurasi dimana harus didefinisikan ke dalam *Perfomance Measure* (PM) yaitu, *true positive* (TP), *false positive* (FP), *true negative* (TN) dan *false negative* (FN).

Tabel 3.3 memberikan hasil klasifikasi dan identifikasi kanker kolorektal pada data *training*, yang selanjutnya hasil tersebut akan digunakan dalam perhitungan *sensitivitas*, *spesificitas*, dan akurasi. Berikut hasil perhitungannya:

- a. $sensitivitas = \frac{TP}{TP+FN} \times 100\% = \frac{53}{53+2} \times 100\% = 96\%$
- b. $spesificitas = \frac{TN}{TN+FP} \times 100\% = \frac{15}{15+4} \times 100\% = 79\%$
- c. $akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% = \frac{53+15}{53+15+4+2} \times 100\% = 92\%$

Tabel 3.3 Hasil Klasifikasi Data *Training* Menggunakan RNN

No	Target	Output	Klasifikasi	PM	No	Target	Output	Klasifikasi	PM
1	1	1,34	1	TN	39	2	1,61	2	TP
2	1	1,48	1	TN	40	2	1,84	2	TP
3	1	1,48	1	TN	41	2	1,59	2	TP
4	1	1,39	1	TN	42	2	1,54	2	TP
5	1	1,36	1	TN	43	3	1,67	2	TP
6	1	1,78	2	FP	44	3	1,52	2	TP
7	1	1,89	2	FP	45	3	1,77	2	TP
8	1	1,39	1	TN	46	3	1,35	1	FN
9	1	1,42	1	TN	47	3	3,04	3	TP
10	1	1,51	2	FP	48	3	2,60	3	TP
11	1	1,46	1	TN	49	3	2,38	2	TP
12	1	1,35	1	TN	50	3	3,09	3	TP
13	1	1,36	1	TN	51	3	2,93	3	TP
14	1	1,36	1	TN	52	3	2,05	2	TP
15	1	1,36	1	TN	53	3	2,38	2	TP
16	1	1,49	1	TN	54	3	2,74	3	TP
17	1	1,51	2	FP	55	3	2,96	3	TP
18	1	1,35	1	TN	56	3	2,80	3	TP
19	1	1,34	1	TN	57	3	2,72	3	TP
20	2	2,74	3	TP	58	3	2,85	3	TP
21	2	1,53	2	TP	59	4	4,18	4	TP
22	2	1,67	2	TP	60	4	4,44	4	TP
23	2	1,49	1	FN	61	4	4,27	4	TP
24	2	2,45	2	TP	62	5	4,43	5	TP
25	2	2,54	3	TP	63	5	4,59	5	TP
26	2	1,98	2	TP	64	5	5,22	5	TP
27	2	2,58	3	TP	65	5	4,84	5	TP
28	2	1,57	2	TP	66	5	5,32	5	TP
29	2	2,34	2	TP	67	5	5,08	5	TP
30	2	2,13	2	TP	68	5	4,44	5	TP
31	2	3,05	3	TP	69	5	5,25	5	TP
32	2	2,35	2	TP	70	5	4,81	5	TP
33	2	2,28	2	TP	71	5	4,65	5	TP
34	2	2,03	2	TP	72	5	4,91	5	TP
35	2	1,94	2	TP	73	5	4,54	5	TP
36	2	1,51	2	TP	74	5	5,19	5	TP
37	2	2,48	2	TP					
38	2	2,08	2	TP					

Terlihat pada tabel 3.3 dan hasil perhitungan bahwa akurasi pada data *training* 90%, hal ini berarti proses pembelajaran pada data *training* mendapatkan model yang baik. Untuk itu dapat dilakukan proses *testing* untuk melihat ketelitian dari model yang telah diperoleh dari proses pembelajaran pada data *training*. Berikut adalah tabel 3.4 yang merupakan hasil klasifikasi dan identifikasi kanker kolorektal pada data *testing*.

Tabel 3.4 Hasil Klasifikasi Data *Testing* Menggunakan RNN

No	Target	Output	Klasifikasi	PM	No	Target	Output	Klasifikasi	PM
1	1	1,35	1	TN	10	3	2,74	3	TP
2	1	1,51	2	FP	11	3	3,16	3	TP
3	1	1,34	1	TN	12	3	2,43	2	TP
4	1	1,34	1	TN	13	3	2,70	3	TP
5	1	1,36	1	TN	14	4	4,28	4	TP
6	2	1,79	2	TP	15	4	4,18	4	TP
7	2	1,91	2	TP	16	5	5,17	5	TP
8	2	1,99	2	TP	17	5	4,98	5	TP
9	2	1,53	2	TP	18	5	4,95	5	TP

Perhitungan *sensitivitas*, *spesificitas*, dan akurasi pada data *testing* :

- $sensitivitas = \frac{TP}{TP+FN} \times 100\% = \frac{13}{13+0} \times 100\% = 100\%$
- $spesificitas = \frac{TN}{TN+FP} \times 100\% = \frac{4}{4+1} \times 100\% = 80\%$
- $akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% = \frac{13+4}{13+4+0+1} \times 100\% = 94\%$

Tingkat sensitifitas, spesifikasi, dan akurasi pada data *testing* memiliki selisih yang sedikit. Tingkat sensitifitas adalah 100% berarti indikator terpercaya saat hasil menunjukkan *negative* (terkena kanker kolorektal). Begitu pula untuk tingkat spesifikasi yaitu 80% berarti indikator juga terpercaya saat hasil menunjukkan *positive* (kolorektal normal).

BAB IV

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pembahasan klasifikasi stadium kanker kolorektal menggunakan model *Recurrent Neural Network* (RNN) dapat disimpulkan hal-hal sebagai berikut:

1. Proses penentuan stadium kanker kolorektal menggunakan model RNN adalah sebagai berikut:
 - a. Mengubah citra gambar *red green blue* (rgb) menjadi citra *grayscale* dari data hasil foto *colonoscopy*.
 - b. Melakukan ekstraksi gambar yang telah diubah dalam citra *grayscale* untuk mengambil sifat-sifat (fitur) dari gambar tersebut. Untuk mengubah tipe gambar dan mengekstrak gambar digunakan proses *Gray Level Cooccurrence Matrix* (GLCM) dengan bantuan *Matlab R2009a* dan untuk mempermudah proses pengekstrakan maka didesain pada *Graphical User Interface* (GUI). Proses pengekstrakan gambar didapat 14 sifat (energi, kontras, korelasi, *Sum of Square*, *Inverse Different Moment* (IDM), *sum average*, *sum variance*, *sum entropy*, *entropy*, *differential variance*, *differential entropy*, maksimum probabilitas, homogenitas, *dissimilarity*) yang selanjutnya akan menjadi variabel *input* dalam pemodelan.
 - c. Membagi data menjadi dua, yaitu data *training* dan data *testing*. Data *training* digunakan untuk pelatihan yang menghasilkan bobot-bobot

jaringan sedangkan data *training* digunakan untuk pengujian atau validasi data. Terdapat 92 data hasil gambar *colonoscopy* kanker kolorektal. Ukuran pembagian data adalah 80% (74 data) untuk data *training* dan 20% (18 data) untuk data *testing*.

- d. Normalisasi *input* dan target data dengan menggunakan *mean* dan *standart deviasi*. Proses normalisasi dilakukan dengan cara mengubah data asli menjadi data yang memiliki *mean*=0 dan *standart deviasi*=1.
 - e. Pembentukan model terbaik dengan cara menentukan neuron pada lapisan tersembunyi dan eliminasi *input*. Pencarian banyaknya neuron tersembunyi dan eliminasi *input* dilakukan dengan cara *trial* dan *error* terhadap beberapa jaringan yang mungkin dengan memilih model yang memiliki nilai MSE dan akurasi terkecil.
 - f. Proses denormalisasi dengan menggunakan bantuan *Matlab R2009a*. Data pada proses pembelajaran dalam bentuk normal, yaitu memiliki *mean*=0 dan *standart deviasi*=1. Setelah pembelajaran, data dikembalikan menjadi data asli atau semula sebelum proses normalisasi.
 - g. Pengukuran ketepatan klasifikasi digunakan untuk menguji apakah model yang digunakan sudah sesuai atau belum. Pengukuran ketepatan klasifikasi dilakukan dengan cara melihat nilai *sensitivitas*, *spesificitas*, dan akurasi.
2. Didapat model terbaik dengan 9 *input* $x_1, x_2, x_3, x_5, x_7, x_9, x_{12}, x_{13}, x_{14}$, dan 1 neuron pada lapisan tersembunyi dengan fungsi aktivasi *sigmoid bipolar* (*tansig*) pada lapisan tersembunyi, dan fungsi linear (*purelin*) pada lapisan *output*. Setelah model terbentuk maka dapat dilihat tingkat keakuratan

penentuan stadium kanker kolorektal dengan melihat nilai *sensitivitas*, *spesifitas*, dan akurasi. Pada 74 data *training* yang menunjukkan 53 data *true positive* (TP), 4 data *false positive* (FP), 15 data *true negative* (TN) dan 2 data *false negative* (FN). Sehingga didapat 96% tingkat *sensitivitas*, 79% tingkat *spesifitas*, 92% tingkat akurasi kebenaran model RNN. Sedangkan pada 18 data *testing* menunjukkan 100%, tingkat *sensitivitas*, 80% tingkat *spesifitas*, 94% tingkat akurasi.

B. Saran

Skripsi ini menggunakan jaringan Elman pada model *Recurrent Neural Network* (RNN) dalam menentukan klasifikasi stadium kanker kolorektal. Masih terdapat banyak cara yang dapat digunakan untuk menentukan stadium kanker kolorektal yang mungkin akan memberikan klasifikasi lebih mendekati pada kondisi sebenarnya, antara lain :

1. Menambahkan *input* atau mereduksi *input* penentuan stadium kanker kolorektal agar hasil yang diperoleh lebih akurat.
2. Menggunakan probabilitas dalam penentuan hasil klasifikasi, karena dalam skripsi ini jika *output* (y) bernilai 2,6 maka dianggap 3 sehingga hasil klasifikasi adalah stadium II, pada umumnya $y = 2,6$ memiliki probabilitas masuk dalam stadium I dan masuk dalam stadium II.
3. Menggunakan jaringan Hopfield yang terdapat juga pada RNN.
4. Menggunakan model lainnya, misalkan *Fuzzy*, *Neuro-Fuzzy*, *ANFIS*, dan lain-lain.

DAFTAR PUSTAKA

- Aditya Angga Kusuma, R. Rizal Isnanto, & Imam Santoso. (2012). Pengenalan Iris Mata Menggunakan Pencirian Matriks Ko-Okurenси Aras Keabuan. *Penelitian Universitas Diponegoro*. Semarang: Universitas Diponegoro.
- Agung Radistya Putra, Achmad Rizal, & M. Syahrul Mubarok. (2012). *Klasifikasi Kanker Usus Besar Berbasis Pengolahan Citra dengan Metode Jaringan Syaraf Tiruan Backpropagation*. Bandung: Institut Teknologi Telkom.
- Alteri, R. et al. (2011). *Colorectal Cancer Facts & Figure 2011-2013*. Atlanta: American Cancer Society.
- American Cancer Society. (2011). *Colorectal Cancer*. Atlanta: American Cancer Society.
- American Joint Committee on Cancer (AJCC). (2010). *Colon and Rectum Cancer Staging*. 7th Edition. Springer-Verlag New York: American Cancer Society.
- Anami, B. S., & Vishwanath, C. B. (2009). Texture Based Identification and Classification of Bulk Sugary Food Object. *ICGST*. Vol. 9. Page 9-14.
- Biglarian, A. et al. (2012). Artificial Neural Network for Prediction of Distant Metastasis in Colorectal. *Asian Pasific Journal of Cancer Prevention*. Vol. 13. Page 927-930.
- Bottaci, L. et. al. (1997). Artificial Neural Network Applied to Outcome Prediction for Colorectal Cancer Patient in Separate Institutions. *The Lancet Journal*. Page 469-472.
- Burt, W. R. (2000). Colon Cancer Screening. *Journal from Gastroenterology*. Vol. 119. Issue 4. Page 837-853.
- Chen, J., & Chaudhari, N. S. (2003). Protein Family Classification Using Second-Order Recurrent Neural Networks. *Genome Informatics*. Vol. 14.
- Fausett, L. (1994). *Fundamental of Neural Network (Architectures, Algorithm, and Application)*. Upper Saddle River, New Jersey: Prentice Hall Inc.
- Gontar Alamsyah Siregar. (2007). *Deteksi Dini dan Penatalaksanaan Kanker Usus Besar*. Medan: Repostory Universitas Sumatera Utara.
- Guntau, J. *Endoskopie Atlas*. Hamburg, Jerman: Albertinen-Krankenhaus. Diakses dari http://endoskopiebilder.de/kolon_normalbefunde.html pada tanggal 21 Maret 2013, Jam 17.00 WIB.

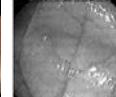
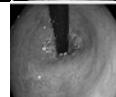
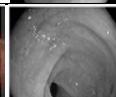
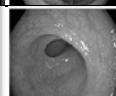
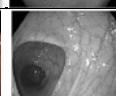
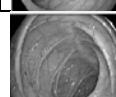
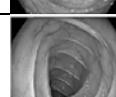
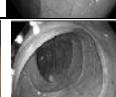
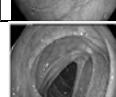
- Hanke, J.E. & Wichern, D.W. (2005). *Bussiness Forecasting, 8th Edition*. Upper Saddle River, New Jersey:Prentice Hall.
- Haralick, R. M., Shanmugam, M., & Dinstein, I. (1973). Features for Image Classification. *IEEE Transaction System, Man and Cybernetics*. Vol. SMC-3. No. 6. Page 610-621.
- Haykin, S. (1999). *Neural Network: A Comprehensive Foundation*. Singapore: Prentice Hall.
- Hota, H. S., Shrivastava, 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 (IJRTE)*. Vol. 1. Issue 6. Page 164-169.
- Hu, H. Y., & Hwang, Jenq-Neng. (2002). *Handbook of Neural Network Signal Processing*. CRC Press.
- International Agency for Research on Cancer (IARC). (2013). *Lastest World Cancer Statistic*. Switzerland: World Health Organization.
- Japerson, K. W. et al. (2010). Hereditary and Familial Colon Cancer. *Gastroenterology*. Vol. 110. Issue 4. Page 335-339.
- Jong Jek Siang. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- Kim ,Kyung-Joong & Cho, Sung-Bae. (2003). Prediction of Colon Cancer Using an Evolutionary Neural Network. *Journal from Elsevier*. Page 361-379.
- Lalkhen, G. A., & McCluskey, A. (2008). Clinical Test: Sensitivity and Specificity. *Oxford Journals*. Vol.8, Issue 6.
- Lin, C. T., & Lee, G. (1996). *Neuro Fuzzy Systems*. Upper Saddle River, New Jersey: Prentice-Hall.
- Medika Risnasari. (2011). *Klasifikasi Disritmia Menggunakan Recurrent Neural Network*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Mohanaiah, P., Sathyanarayana, P., & GuruKumar, L. (2013). Image Texture Feature Extraction Using GLCM Approach. *International Journal of Scientific and Research Publication*. Vol. 3.
- Murra-Saca, J. *Vision and Image Analysis Group*. El Salvador: Cornell University. Diakses dari <http://www.gastrointestinalatlas.com/English/> pada tanggal 21 Maret 2013, Jam 16.10 WIB.

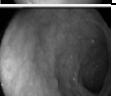
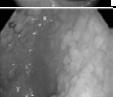
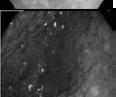
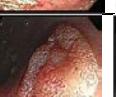
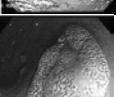
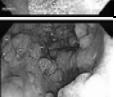
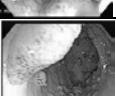
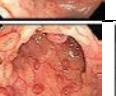
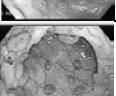
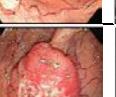
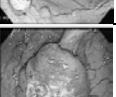
- National Cancer Institute. (2006). *Cancer of the Colon and Rectum*. Diakses dari <http://www.cancer.gov/cancertopics/types/colon-and-rectal> pada tanggal 22 Desember 2013, Jam 11.00 WIB.
- Nwoye, E. et al. (2005). *Spectral and Statistical Features in Fuzzy Neural Expert Machine for Colorectal Adenomas and Adenocarcinoma Classification*. United Kingdom: University of Newcastle Upon Tyne.
- Parker, S. (2009). *The Human Body Book (Ensiklopedia Tubuh Manusia)*. Penerjemah: Winardini. Indonesia: Penerbit Erlangga.
- Prytherch, R. (2005). *Harrod's Librarians' Glossary and Reference Book*. England : Ashgate Publishing Company.
- R.H. Sianipar. 2013. *Pemrograman Matlab Dalam Contoh dan Penerapan*. Informatika: Bandung.
- Sri Kusumadewi. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*. Yogyakarta: Graha Ilmu.
- Samarasinghe, S. (2007). *Neural Networks for Applied Sciences and Engineering*. New York: Taylor & Francis Group.
- Sharma, M. & Sourabh, M. (2013). Artificial Neural Network Fuzzy Inference System (ANFIS) for Brain Tumor Detection. *Advances in Intelligent Systems and Computing*. Vol. 177. Page 329-339.
- Tortora, G. J., & Grabowski, S. R. (2002). *Principles of Anatomy and Physiology*. 10th Edition. Columbia: John Wiley & Sons Inc .
- Uppuluri, A. (2008). *GLCM Texture Features*. Diakses dari http://www.mathworks.com/matlabcentral/fileexchange/22187-glcm-textures/content/GLCM_Features1.m pada tanggal 20 Desember 2013, Jam 16.00 WIB.
- Widodo Jadarwanto. (2006). Reaksi Simpang Makanan. *Jurnal Dexa Media*. Vol. 19. Hlm. 88-93.
- World Health Organization (WHO). (1977). *Manual of The International Statistical Classification of Diseases, Injuries, and Cause of Death*. Switzerland: World Health Organization.
- Zhang, G. P. (2004). *Neural Networks in Business Forcasting*. Amerika: Idea Group Publishing Inc.

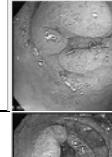
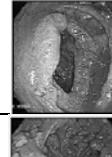
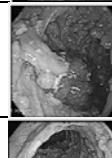
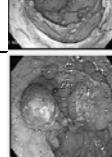
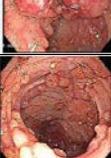
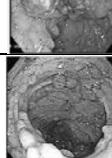
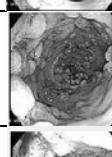
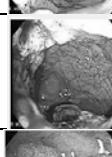
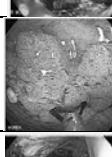
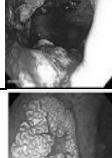
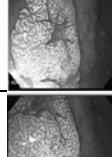
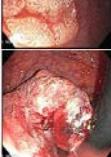
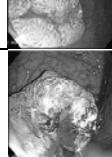
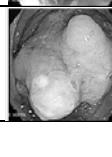
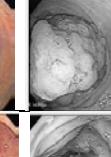
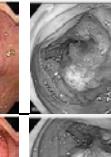
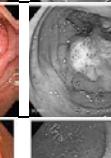
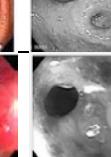
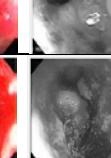
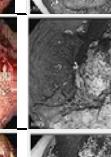
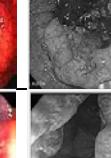
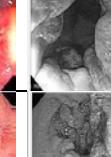
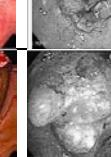
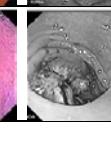
LAMPIRAN

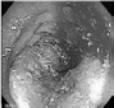
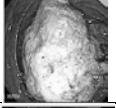
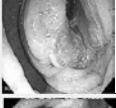
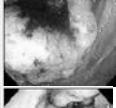
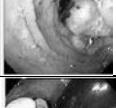
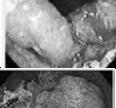
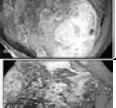
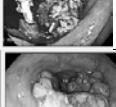
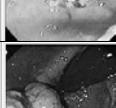
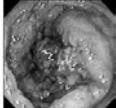
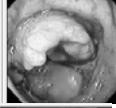
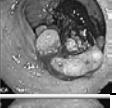
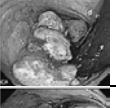
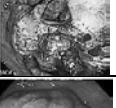
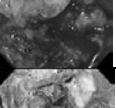
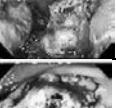
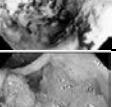
Lampiran 1

Gambar Colonoscopy Data Training

No	Target	Gambar Asli dan Grayscale	
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		

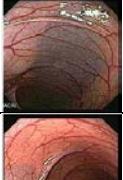
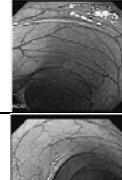
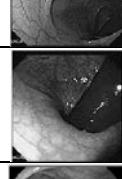
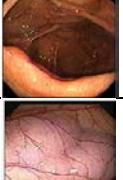
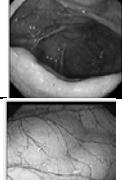
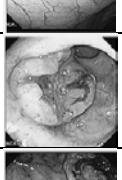
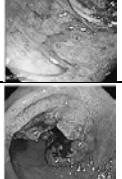
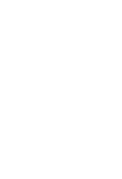
14	Normal		
15	Normal		
16	Normal		
17	Normal		
18	Normal		
19	Normal		
20	Stadium1		
21	Stadium1		
22	Stadium1		
23	Stadium1		
24	Stadium1		
25	Stadium1		
26	Stadium1		
27	Stadium1		

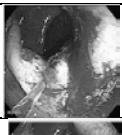
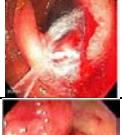
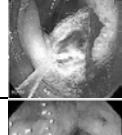
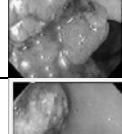
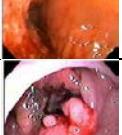
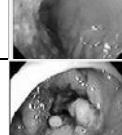
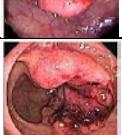
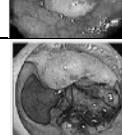
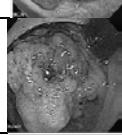
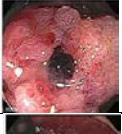
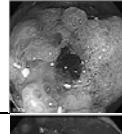
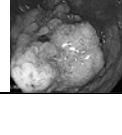
28	Stadium1		
29	Stadium1		
30	Stadium1		
31	Stadium1		
32	Stadium1		
33	Stadium1		
34	Stadium1		
35	Stadium1		
36	Stadium1		
37	Stadium1		
38	Stadium1		
39	Stadium1		
40	Stadium1		
41	Stadium1		
42	Stadium1		
43	Stadium2		
44	Stadium2		
45	Stadium2		
46	Stadium2		
47	Stadium2		
48	Stadium2		
49	Stadium2		
50	Stadium2		
51	Stadium2		
52	Stadium2		
53	Stadium2		
54	Stadium2		
55	Stadium2		

56	Stadium2		
57	Stadium2		
58	Stadium2		
59	Stadium3		
60	Stadium3		
61	Stadium3		
62	Stadium4		
63	Stadium4		
64	Stadium4		
65	Stadium4		
66	Stadium4		
67	Stadium4		
68	Stadium4		
69	Stadium4		
70	Stadium4		
71	Stadium4		
72	Stadium4		
73	Stadium4		
74	Stadium4		

Lampiran 2

Gambar *Colonoscopy Data Testing*

No	Target	Gambar Asli dan Grayscale	
1	Normal		
2	Normal		
3	Normal		
4	Normal		
5	Normal		
6	Stadium1		
7	Stadium1		
8	Stadium1		
9	Stadium1		

10	Stadium2		
11	Stadium2		
12	Stadium2		
13	Stadium2		
14	Stadium3		
15	Stadium3		
16	Stadium4		
17	Stadium4		
18	Stadium4		

Lampiran 3 Mfile GLCM Ekstraksi 14 Fitur

(dapat diunduh di http://www.mathworks.com/matlabcentral/fileexchange/22187-glcm-textures/content/GLCM_Features1.m)

```
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_glcmin_1 = size(glcmin,1);
size_glcmin_2 = size(glcmin,2);
size_glcmin_3 = size(glcmin,3);
% checked
out.contr = zeros(1,size_glcmin_3); % Contrast: matlab/[1,2]
out.corr = zeros(1,size_glcmin_3); % Correlation: matlab
out.dissi = zeros(1,size_glcmin_3); % Dissimilarity: [2]
out.energ = zeros(1,size_glcmin_3); % Energy: matlab / [1,2]
out.entro = zeros(1,size_glcmin_3); % Entropy: [2]
out.homom = zeros(1,size_glcmin_3); % Homogeneity: matlab
out.maxpr = zeros(1,size_glcmin_3); % Maximum probability: [2]
out.sosvh = zeros(1,size_glcmin_3); % Sum of squares: Variance [1]
out.savgh = zeros(1,size_glcmin_3); % Sum average [1]
out.svarh = zeros(1,size_glcmin_3); % Sum variance [1]
out.senth = zeros(1,size_glcmin_3); % Sum entropy [1]
out.dvarh = zeros(1,size_glcmin_3); % Difference variance [4]
out.denth = zeros(1,size_glcmin_3); % Difference entropy [1]
```

```

out.idmnc = zeros(1,size_glcmb_3); % Inverse difference moment
normalized [3]

glcm_sum = zeros(size_glcmb_3,1);
glcm_mean = zeros(size_glcmb_3,1);
glcm_var = zeros(size_glcmb_3,1);

% the values of correlation that are provided by matlab.
u_x = zeros(size_glcmb_3,1);
u_y = zeros(size_glcmb_3,1);
s_x = zeros(size_glcmb_3,1);
s_y = zeros(size_glcmb_3,1);
% checked p_x p_y p_xplusy p_xminusy
p_x = zeros(size_glcmb_1,size_glcmb_3); % Ng x #glcms[1]
p_y = zeros(size_glcmb_2,size_glcmb_3); % Ng x #glcms[1]
p_xplusy = zeros((size_glcmb_1*2 - 1),size_glcmb_3); %[1]
p_xminusy = zeros((size_glcmb_1),size_glcmb_3); %[1]
% checked hxy hxy1 hxy2 hx hy
hxy = zeros(size_glcmb_3,1);
hxy1 = zeros(size_glcmb_3,1);
hx = zeros(size_glcmb_3,1);
hy = zeros(size_glcmb_3,1);
hxy2 = zeros(size_glcmb_3,1);

for k = 1:size_glcmb_3 % number glcms
    glcm_sum(k) = sum(sum(glcm(:,:,:k)));
    glcm(:,:,:k) = glcm(:,:,:k)./glcm_sum(k); % Normalize each glcm
    glcm_mean(k) = mean2(glcm(:,:,:k)); % compute mean after norm
    glcm_var(k) = (std2(glcm(:,:,:k)))^2;

    for i = 1:size_glcmb_1
        for j = 1:size_glcmb_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_glcmb_1) ));
            out.idmnc(k) = out.idmnc(k) + (glcm(i,j,k)/( 1 + ((i - j)/size_glcmb_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
for k = 1:size_glcmb_3

    for i = 1:size_glcmb_1

        for j = 1:size_glcmb_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_glcmb_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_glcmb_1-1)]))
                p_xminusy((abs(i-j))+1,k) = p_xminusy((abs(i-
j))+1,k) +...
                    glcm(i,j,k);
            end
        end
    end
end
% computing sum average, sum variance and sum entropy:
for k = 1:(size_glcmb_3)

    for i = 1:(2*(size_glcmb_1)-1)
        out.savgh(k) = out.savgh(k) + (i+1)*p_xplusy(i,k);
        % the summation for savgh is for i from 2 to 2*Ng hence
(i+1)
        out.senth(k) = out.senth(k) -
(p_xplusy(i,k)*log(p_xplusy(i,k) + eps));
    end
end
% compute sum variance with the help of sum entropy
for k = 1:(size_glcmb_3)

    for i = 1:(2*(size_glcmb_1)-1)
        out.svarh(k) = out.svarh(k) + (((i+1) -
out.senth(k))^2)*p_xplusy(i,k);
    end
end
% compute difference variance, difference entropy,
for k = 1:size_glcmb_3
    for i = 0:(size_glcmb_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
% compute information measure of correlation(1,2) [1]
for k = 1:size_glcmb_3
    hxy(k) = out.entro(k);
    for i = 1:size_glcmb_1

        for j = 1:size_glcmb_2
            hxy1(k) = hxy1(k) - (glcm(i,j,k)*log(p_x(i,k)*p_y(j,k)
+ eps));
        end
    end
end

```

```

        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.inf1h(k) = ( hxy(k) - hxy1(k) ) / ( max([hx(k),hy(k)]) );
out.inf2h(k) = ( 1 - exp( -2*( hxy2(k) - hxy(k) ) ) )^0.5;
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
    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.corrm(k) = corm(k) / (s_x(k)*s_y(k));
end

```

Lampiran 4 Mfile Program *Graphical User Interface* (GUI)

```
function varargout = gldm_baru(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name', '', 'filename', ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @gldm_baru_OpeningFcn, ...
    'gui_OutputFcn', @gldm_baru_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% Kode diatas tidak boleh diedit

function gldm_baru_OpeningFcn(hObject, eventdata, handles,
varargin)
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_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function pushbutton_keluar_Callback(hObject, eventdata, handles)
close;

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]);
axes(handles.axes_warna);
imshow(I);
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

function pushbutton_ubah_Callback(hObject, eventdata, handles)
global I;
global J;
J=rgb2gray(I);
axes(handles.axes_gray);
imshow(J);
set(handles.pushbutton_proses, 'enable', 'on');
set(handles.uitable_fitur, 'enable', 'on');

% Tombol proses pada GUI
function pushbutton_proses_Callback(hObject, eventdata, handles)
global J;
global P;
GLCM=graycomatrix(J);
stats_1 = GLCM_Features1(GLCM,0);
P(1,:)=stats_1.energ;% Energy
P(2,:)=stats_1.contr;% Contrast
P(3,:)=stats_1.corrn;% 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
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(handlesuitable_fitur,'enable','off');
set(handles.edit_alamat,'String','');

% Tombol save pada GUI
function pushbutton_save_Callback(hObject, eventdata, handles)
global P;
filename = 'Fitur.xlsx';
xlswrite(filename,[P]); %pindah ke excel
```

Lampiran 5 Tabel Hasil Ekstraksi Gambar Data *Training*

No.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	
No.	Energi	Kontras	Korelasi	VAR	IDM	SA	SV	SE	Entropy	DV	DE	MP	Hom	D	Target
1	0,1708	0,1605	0,9656	13,1833	0,9976	6,6182	29,9486	2,0610	2,1897	0,1605	0,4324	0,2833	0,9283	0,1474	1
2	0,1009	0,2486	0,9613	20,6479	0,9962	8,3804	48,7894	2,3660	2,5657	0,2486	0,5563	0,1583	0,8959	0,2177	1
3	0,2100	0,1479	0,9566	20,3182	0,9977	8,6592	52,0586	1,9221	2,0430	0,1479	0,4112	0,3526	0,9336	0,1364	1
4	0,1066	0,2662	0,9590	25,8768	0,9960	9,5456	64,0710	2,3794	2,6034	0,2662	0,5727	0,2038	0,8949	0,2238	1
5	0,1460	0,2463	0,9298	15,6307	0,9964	7,4850	35,9666	2,0816	2,2630	0,2463	0,5209	0,2413	0,9083	0,1957	1
6	0,2120	0,2230	0,9180	15,6260	0,9967	7,5847	38,5543	1,8107	1,9730	0,2230	0,4862	0,3603	0,9192	0,1741	1
7	0,2340	0,2749	0,8957	15,4763	0,9960	7,5559	38,3588	1,7799	1,9678	0,2749	0,5230	0,4182	0,9136	0,1932	1
8	0,1310	0,3557	0,9154	19,2722	0,9948	8,3166	45,8975	2,1648	2,4261	0,3557	0,6264	0,2464	0,8851	0,2558	1
9	0,1671	0,4153	0,8676	17,9520	0,9939	8,1268	43,6242	1,9815	2,2895	0,4153	0,6895	0,2964	0,8651	0,3004	1
10	0,1754	0,2409	0,9338	19,2149	0,9964	8,3727	47,2646	2,0300	2,2234	0,2409	0,5285	0,3215	0,9076	0,1971	1
11	0,1878	0,1691	0,9523	15,0209	0,9975	7,3122	36,4974	1,8737	1,9951	0,1691	0,4328	0,2813	0,9289	0,1479	1
12	0,1415	0,3070	0,9205	16,5913	0,9954	7,6812	37,7925	2,1698	2,4068	0,3070	0,6040	0,2675	0,8858	0,2453	1
13	0,1907	0,4017	0,8618	14,5910	0,9942	7,2800	33,7221	1,9594	2,2406	0,4017	0,6524	0,3846	0,8796	0,2730	1
14	0,2292	0,2486	0,8901	12,9700	0,9963	6,9060	30,5602	1,7790	1,9596	0,2486	0,5136	0,3759	0,9135	0,1884	1
15	0,1360	0,1666	0,9671	18,3121	0,9975	7,9736	43,2820	2,2014	2,3316	0,1666	0,4130	0,2286	0,9358	0,1363	1
16	0,1697	0,1214	0,9673	13,9804	0,9982	6,9949	32,2979	1,9945	2,0816	0,1214	0,3438	0,2786	0,9498	0,1048	1
17	0,1724	0,1380	0,9597	11,9164	0,9979	6,4175	26,4159	1,9781	2,0826	0,1380	0,3730	0,2926	0,9434	0,1183	1
18	0,1685	0,2160	0,9411	19,7459	0,9968	8,4978	49,9975	1,9495	2,1107	0,2160	0,4781	0,2826	0,9212	0,1695	1
19	0,1983	0,3709	0,7913	8,0703	0,9947	5,3919	16,3269	1,7664	2,0008	0,3709	0,6212	0,2974	0,8815	0,2612	1
20	0,0753	0,4943	0,9235	18,9849	0,9926	7,9691	42,2553	2,5080	2,8958	0,4943	0,7776	0,1287	0,8283	0,3774	2
21	0,1196	0,3227	0,9613	17,7940	0,9951	7,4133	41,5817	2,3908	2,6501	0,3227	0,6433	0,2827	0,8694	0,2760	2
22	0,0897	0,3518	0,9549	22,9867	0,9948	8,7692	54,8236	2,4775	2,7437	0,3518	0,6450	0,1768	0,8742	0,2734	2

23	0,0909	0,2917	0,9647	25,8005	0,9957	9,3398	63,4099	2,4730	2,6981	0,2917	0,5804	0,1499	0,8943	0,2287	2
24	0,1215	0,3758	0,9243	26,1020	0,9943	9,7508	65,9404	2,2420	2,5399	0,3758	0,6889	0,2344	0,8557	0,3088	2
25	0,1216	0,4226	0,8954	20,4998	0,9936	8,6288	48,5739	2,2320	2,5709	0,4226	0,7287	0,2479	0,8399	0,3437	2
26	0,0927	0,4324	0,9223	21,0839	0,9936	8,5866	48,6801	2,4240	2,7678	0,4324	0,7224	0,1654	0,8513	0,3272	2
27	0,0840	0,4455	0,9339	19,8986	0,9933	8,1624	45,4327	2,4700	2,8252	0,4455	0,7468	0,1740	0,8355	0,3558	2
28	0,1625	0,2620	0,9293	21,9565	0,9960	8,9982	55,3700	2,0543	2,2642	0,2620	0,5758	0,2896	0,8896	0,2306	2
29	0,1115	0,4258	0,9001	16,8653	0,9936	7,7078	37,5546	2,2806	2,6234	0,4258	0,7303	0,2089	0,8413	0,3422	2
30	0,1001	0,3964	0,9164	16,0712	0,9941	7,4335	35,0481	2,3385	2,6568	0,3964	0,7028	0,1992	0,8517	0,3191	2
31	0,0786	0,4916	0,9105	20,9694	0,9926	8,5683	47,5993	2,4774	2,8725	0,4916	0,7828	0,1538	0,8206	0,3897	2
32	0,1331	0,4855	0,8638	18,0155	0,9928	8,0888	41,5190	2,1835	2,5590	0,4855	0,7643	0,2932	0,8361	0,3627	2
33	0,0874	0,4232	0,9284	22,2896	0,9937	8,8253	52,0105	2,4523	2,7927	0,4232	0,7271	0,1800	0,8434	0,3384	2
34	0,0914	0,4148	0,9287	22,0533	0,9938	8,7814	51,5187	2,4339	2,7643	0,4148	0,7126	0,1732	0,8517	0,3229	2
35	0,0950	0,3721	0,9455	21,1419	0,9944	8,4517	49,5133	2,4329	2,7267	0,3721	0,6817	0,1838	0,8568	0,3055	2
36	0,1233	0,2775	0,9757	24,3456	0,9959	8,6633	62,9968	2,3044	2,5097	0,2775	0,5565	0,2106	0,9019	0,2133	2
37	0,1038	0,3861	0,9070	21,1125	0,9942	8,7550	49,4015	2,3142	2,6272	0,3861	0,7041	0,1747	0,8421	0,3323	2
38	0,0908	0,3933	0,9490	20,2109	0,9941	8,1153	46,6871	2,4889	2,7911	0,3933	0,7100	0,2015	0,8460	0,3285	2
39	0,0948	0,3206	0,9530	18,7229	0,9951	7,8566	42,6314	2,4436	2,7034	0,3206	0,6404	0,1751	0,8702	0,2742	2
40	0,0928	0,3937	0,9353	20,9753	0,9941	8,5000	48,2505	2,4611	2,7769	0,3937	0,7020	0,1847	0,8521	0,3182	2
41	0,1081	0,2819	0,9540	23,8985	0,9957	9,1600	58,3535	2,3497	2,5779	0,2819	0,5920	0,2038	0,8865	0,2395	2
42	0,1321	0,2511	0,9482	23,3362	0,9962	9,1769	57,9352	2,2134	2,4161	0,2511	0,5609	0,2312	0,8950	0,2198	2
43	0,1429	0,3364	0,9098	22,1100	0,9950	9,0251	54,4361	2,1467	2,4224	0,3364	0,6442	0,2603	0,8736	0,2719	3
44	0,1591	0,2795	0,9116	21,0342	0,9958	8,8444	51,8059	2,0796	2,3076	0,2795	0,5950	0,2646	0,8833	0,2442	3
45	0,1453	0,2923	0,9076	20,3363	0,9956	8,6919	49,6539	2,0872	2,3241	0,2923	0,6114	0,2601	0,8764	0,2577	3
46	0,1651	0,1572	0,9657	22,1889	0,9976	8,9474	55,6055	2,1210	2,2482	0,1572	0,4077	0,2987	0,9367	0,1334	3
47	0,1820	0,2221	0,9490	18,3639	0,9967	8,0890	44,9982	2,0462	2,2109	0,2221	0,5038	0,3212	0,9102	0,1881	3

48	0,0732	0,8392	0,8302	16,3608	0,9879	7,4684	34,5769	2,4156	2,9871	0,8392	0,9528	0,1374	0,7696	0,5395	3
49	0,0746	0,8076	0,8421	16,1289	0,9882	7,3959	34,2071	2,4171	2,9832	0,8076	0,9510	0,1759	0,7631	0,5467	3
50	0,0533	1,1390	0,8072	19,7919	0,9837	8,2265	43,1197	2,5307	3,2424	1,1390	1,0726	0,1129	0,7232	0,6754	3
51	0,1168	0,5079	0,8074	11,9127	0,9925	6,5372	24,4511	2,1005	2,4794	0,5079	0,7788	0,1882	0,8172	0,3956	3
52	0,1068	0,2333	0,9608	20,7751	0,9965	8,4678	48,6859	2,3850	2,5698	0,2333	0,5404	0,1811	0,8984	0,2101	3
53	0,1350	0,3708	0,8799	21,6048	0,9944	8,9911	52,9327	2,1260	2,4262	0,3708	0,6893	0,2313	0,8477	0,3201	3
54	0,0988	0,3576	0,9290	20,6935	0,9946	8,5573	47,7074	2,3937	2,6799	0,3576	0,6752	0,1946	0,8539	0,3071	3
55	0,1582	0,3980	0,9085	25,5589	0,9942	9,6898	65,2520	2,1428	2,4357	0,3980	0,6633	0,3382	0,8755	0,2801	3
56	0,1113	0,3542	0,9185	21,6237	0,9946	8,8555	51,3260	2,2987	2,5891	0,3542	0,6736	0,2006	0,8568	0,3023	3
57	0,0837	0,4019	0,9498	24,5960	0,9940	9,1076	58,8512	2,5309	2,8454	0,4019	0,7095	0,2010	0,8438	0,3323	3
58	0,0973	0,2952	0,9541	27,1185	0,9955	9,8029	66,8566	2,4337	2,6701	0,2952	0,6099	0,1602	0,8793	0,2537	3
59	0,1019	0,1853	0,9744	23,4446	0,9972	8,9431	57,0707	2,4008	2,5491	0,1853	0,4602	0,1458	0,9232	0,1607	4
60	0,1139	0,2163	0,9640	27,8968	0,9967	10,0084	70,6310	2,3377	2,5171	0,2163	0,5130	0,1683	0,9090	0,1901	4
61	0,0887	0,2885	0,9634	22,7742	0,9957	8,7048	54,1224	2,4870	2,7217	0,2885	0,5930	0,1558	0,8887	0,2375	4
62	0,0764	0,4169	0,9374	21,3042	0,9937	8,5116	48,6876	2,5291	2,8609	0,4169	0,7265	0,1575	0,8352	0,3497	5
63	0,0703	0,5993	0,8991	21,1951	0,9911	8,5721	47,8678	2,5238	2,9833	0,5993	0,8461	0,1314	0,8036	0,4392	5
64	0,1348	0,2580	0,9485	24,6553	0,9961	9,4415	61,6609	2,2366	2,4412	0,2580	0,5482	0,2558	0,9020	0,2094	5
65	0,1500	0,3922	0,9095	11,3344	0,9943	6,0822	23,7194	2,1523	2,4377	0,3922	0,6714	0,2703	0,8653	0,2940	5
66	0,1208	0,2771	0,9341	17,4954	0,9958	7,8760	39,7359	2,2546	2,4772	0,2771	0,5885	0,2112	0,8860	0,2392	5
67	0,1213	0,1920	0,9644	24,0908	0,9971	9,2833	59,5295	2,2877	2,4465	0,1920	0,4818	0,1988	0,9164	0,1732	5
68	0,0974	0,5113	0,8928	19,3852	0,9926	8,2759	43,8132	2,3783	2,7571	0,5113	0,7613	0,1874	0,8408	0,3575	5
69	0,1028	0,4508	0,9025	19,6398	0,9933	8,3571	44,7723	2,3589	2,7098	0,4508	0,7351	0,2189	0,8431	0,3428	5
70	0,0690	0,7055	0,8878	19,5537	0,9897	8,1326	43,4185	2,5163	3,0193	0,7055	0,8917	0,1417	0,7896	0,4805	5
71	0,1978	0,1675	0,9345	10,3502	0,9974	6,0547	22,3353	1,8851	2,0194	0,1675	0,4426	0,3327	0,9262	0,1523	5
72	0,0992	0,3055	0,9415	20,8896	0,9954	8,5876	48,7111	2,3763	2,6267	0,3055	0,6260	0,1768	0,8728	0,2665	5

73	0,0802	0,2778	0,9658	24,7276	0,9958	9,1302	59,2843	2,5482	2,7723	0,2778	0,6004	0,1335	0,8780	0,2524	5
74	0,1625	0,2235	0,9489	25,8820	0,9966	9,7688	67,2752	2,1032	2,2878	0,2235	0,5247	0,2769	0,9054	0,1975	5

Lampiran 6 Tabel Hasil Ekstraksi Gambar Data *Testing*

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	
No.	Energi	Kontras	Korelasi	VAR	IDM	SA	SV	SE	Entropy	DV	DE	MP	Hom	D	Target
1	0,1078	0,2041	0,9782	24,5185	0,9969	8,9275	61,4974	2,3690	2,5387	0,2041	0,4847	0,1740	0,9189	0,1720	0
2	0,1204	0,2365	0,9668	17,7099	0,9964	7,5499	41,4758	2,3100	2,4995	0,2365	0,5355	0,2401	0,9029	0,2038	0
3	0,1378	0,2574	0,9340	12,9173	0,9961	6,6439	27,6048	2,1644	2,3683	0,2574	0,5543	0,2154	0,8992	0,2139	0
4	0,1159	0,2446	0,9541	14,7924	0,9963	6,9949	32,8864	2,2550	2,4540	0,2446	0,5396	0,1853	0,9037	0,2043	0
5	0,1438	0,2092	0,9686	14,1508	0,9969	6,6050	32,4662	2,2059	2,3601	0,2092	0,4595	0,2814	0,9265	0,1598	0
6	0,1171	0,3334	0,9383	29,4780	0,9950	10,3786	75,7444	2,2995	2,5644	0,3334	0,6374	0,2039	0,8765	0,2665	1
7	0,0918	0,3801	0,9345	22,4434	0,9944	8,8696	52,7259	2,4280	2,7189	0,3801	0,6739	0,1661	0,8624	0,2973	1
8	0,1174	0,4193	0,8909	17,0147	0,9938	7,7951	38,1308	2,2396	2,5627	0,4193	0,7031	0,2205	0,8573	0,3140	1
9	0,1560	0,3113	0,9125	19,3848	0,9953	8,4192	46,6995	2,1029	2,3503	0,3113	0,6164	0,2619	0,8805	0,2550	1
10	0,1036	0,3112	0,9557	20,1877	0,9953	8,2024	47,1381	2,4228	2,6722	0,3112	0,6274	0,2266	0,8741	0,2654	2
11	0,1071	0,3216	0,9440	22,9773	0,9951	8,9977	54,9617	2,3801	2,6390	0,3216	0,6386	0,2228	0,8694	0,2751	2
12	0,1875	0,1616	0,9578	19,6631	0,9975	8,4577	48,9591	2,0180	2,1523	0,1616	0,4321	0,3367	0,9293	0,1463	2
13	0,2030	0,1317	0,9611	19,5001	0,9980	8,4706	49,5809	1,9171	2,0260	0,1317	0,3816	0,3634	0,9405	0,1221	2
14	0,1262	0,3434	0,9483	22,4279	0,9950	8,7791	54,8348	2,3057	2,5663	0,3434	0,6021	0,2761	0,8949	0,2379	3
15	0,1066	0,4398	0,8912	17,9382	0,9936	8,0115	40,3064	2,2964	2,6206	0,4398	0,7087	0,1773	0,8524	0,3238	3
16	0,1283	0,3828	0,8756	13,7114	0,9943	7,0088	29,3299	2,1549	2,4514	0,3828	0,6887	0,2273	0,8546	0,3111	4
17	0,1078	0,3622	0,9165	18,2022	0,9945	8,0387	41,1071	2,3123	2,6014	0,3622	0,6773	0,1933	0,8550	0,3063	4
18	0,1145	0,3248	0,9355	15,5207	0,9951	7,2431	33,9628	2,3227	2,5761	0,3248	0,6412	0,2405	0,8650	0,2822	4

Lampiran 7 Proses Pembelajaran Model *Recurrent Neural Network*

```
X1=[ ];%X1 adalah matriks input-output TRAINING
X2=[ ];%X2 adalah matriks input-output TESTING
P=[X1(:,1:7)']; %input TRAINING
Pc=[X2(:,1:7)']; %input TESTING
T=X1(:,8)';%target output TRAINING
Tc=X2(:,8)';%target output Testing
%normalisasi input dan target output training
[Pn,meanp,stdp,Tn,meant,stdt]=prestd(P,T);
%inisialisasi jaringan syaraf
net=newelm(minmax(Pn),[10 1],{'tansig' 'purelin'}, 'traingdx');
%melihat bobot awal input, lapisan, dan bias
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,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};
%menghitung MSE data training
ab = sim(net,Pn);
a=poststd(ab,meant,stdt);
e1 = T-a;
MSE1 = mse(e1)
%menampilkan target, output, dan eror data training
H=[(1:size(P,2))' T' a' e1']
[m1,b1,r1] = postreg(a,T);
%normalisasi input dan target output testing
Qn=trastd(Pc,meanp,stdp);
bn=sim(net,Qn)
b=poststd(bn,meant,stdt)
%menghitung MSE data testing
e2=Tc-b;
MSE2=mse(e2)
```

Lampiran 8 Nilai MSE dan Akurasi Pada Tahapan Eliminasi *Input*

No	Eliminasi	MSE		Akurasi	
		Training	Testing	Training	Testing
1	-	1,207	2,271	77%	94%
2	x_{11}	1,113	1,232	79%	94%
3	x_{11}, x_6	0,802	0,494	81%	94%
4	x_{11}, x_6, x_8	0,598	0,154	85%	94%
5	x_{11}, x_6, x_8, x_4^*	0,282*	0,0931*	92%*	94%*
6	$x_{11}, x_6, x_8, x_4, x_1$	0,2901	0,0983	92%	94%
7	$x_{11}, x_6, x_8, x_4, x_2$	0,2891	0,0899	91%	94%
8	$x_{11}, x_6, x_8, x_4, x_3$	0,2931	0,1082	89,2%	94%
9	$x_{11}, x_6, x_8, x_4, x_5$	2,3336	2,1670	70%	94%
10	$x_{11}, x_6, x_8, x_4, x_7$	0,2890	0,1018	90,5%	94%
11	$x_{11}, x_6, x_8, x_4, x_9$	0,2992	0,0967	89,4%	94%
12	$x_{11}, x_6, x_8, x_4, x_{12}$	0,2835	0,0941	91,8%	94%
13	$x_{11}, x_6, x_8, x_4, x_{13}$	0,2885	0,1004	90,5%	94%
14	$x_{11}, x_6, x_8, x_4, x_{14}$	1,4492	2,4036	81%	94%

Lampiran 9 Data *Target, Output, Error*, dan Hasil Klasifikasi Data *Training*

No	Target	Output	Error	Klasifikasi	Keterangan Kesesuaian
1	1	1,343233	-0,34323	1	Sesuai
2	1	1,477089	-0,47709	1	Sesuai
3	1	1,484735	-0,48473	1	Sesuai
4	1	1,385322	-0,38532	1	Sesuai
5	1	1,358326	-0,35833	1	Sesuai
6	1	1,752589	-0,75259	2	Tidak Sesuai
7	1	1,891559	-0,89156	2	Tidak Sesuai
8	1	1,398039	-0,39804	1	Sesuai
9	1	1,419284	-0,41928	1	Sesuai
10	1	1,511363	-0,51136	2	Tidak Sesuai
11	1	1,45653	-0,45653	1	Sesuai
12	1	1,351989	-0,35199	1	Sesuai
13	1	1,360226	-0,36023	1	Sesuai
14	1	1,354566	-0,35457	1	Sesuai
15	1	1,357743	-0,35774	1	Sesuai
16	1	1,49683	-0,49683	1	Sesuai
17	1	1,505261	-0,50526	2	Tidak Sesuai
18	1	1,345967	-0,34597	1	Sesuai
19	1	1,339573	-0,33957	1	Sesuai
20	2	2,737421	-0,73742	3	Tidak Sesuai
21	2	1,529869	0,470131	2	Sesuai
22	2	1,667153	0,332847	2	Sesuai
23	2	1,485314	0,514686	1	Tidak Sesuai
24	2	2,449397	-0,4494	2	Sesuai
25	2	2,541998	-0,542	3	Tidak Sesuai
26	2	1,983436	0,016564	2	Sesuai
27	2	2,580851	-0,58085	3	Tidak Sesuai
28	2	1,569151	0,430849	2	Sesuai
29	2	2,335641	-0,33564	2	Sesuai
30	2	2,126113	-0,12611	2	Sesuai
31	2	3,050665	-1,05066	3	Tidak Sesuai
32	2	2,353111	-0,35311	2	Sesuai
33	2	2,282031	-0,28203	2	Sesuai
34	2	2,033328	-0,03333	2	Sesuai
35	2	1,940781	0,059219	2	Sesuai
36	2	1,511619	0,488381	2	Sesuai
37	2	2,476183	-0,47618	2	Sesuai
38	2	2,084225	-0,08423	2	Sesuai

39	2	1,612724	0,387276	2	Sesuai
40	2	1,841211	0,158789	2	Sesuai
41	2	1,585851	0,414149	2	Sesuai
42	2	1,535102	0,464898	2	Sesuai
43	3	1,667507	1,332493	2	Tidak Sesuai
44	3	1,517046	1,482954	2	Tidak Sesuai
45	3	1,774512	1,225488	2	Tidak Sesuai
46	3	1,347358	1,652642	1	Tidak Sesuai
47	3	3,039927	-0,03993	3	Sesuai
48	3	2,597499	0,402501	3	Sesuai
49	3	2,379625	0,620375	2	Tidak Sesuai
50	3	3,090629	-0,09063	3	Sesuai
51	3	2,932527	0,067473	3	Sesuai
52	3	2,052773	0,947227	2	Tidak Sesuai
53	3	2,377935	0,622065	2	Tidak Sesuai
54	3	2,739304	0,260696	3	Sesuai
55	3	2,960868	0,039132	3	Sesuai
56	3	2,803605	0,196395	3	Sesuai
57	3	2,717091	0,282909	3	Sesuai
58	3	2,845639	0,154361	3	Sesuai
59	4	4,181612	-0,18161	4	Sesuai
60	4	4,441026	-0,44103	4	Sesuai
61	4	4,272867	-0,27287	4	Sesuai
62	5	4,427061	0,572939	5	Sesuai
63	5	4,593393	0,406607	5	Sesuai
64	5	5,224557	-0,22456	5	Sesuai
65	5	4,839453	0,160547	5	Sesuai
66	5	5,322606	-0,32261	5	Sesuai
67	5	5,08161	-0,08161	5	Sesuai
68	5	4,443636	0,556364	5	Sesuai
69	5	5,251289	-0,25129	5	Sesuai
70	5	4,812229	0,187771	5	Sesuai
71	5	4,652783	0,347217	5	Sesuai
72	5	4,914285	0,085715	5	Sesuai
73	5	4,543614	0,456386	5	Sesuai
74	5	5,185327	-0,18533	5	Sesuai

Lampiran 10 Data *Target, Output, Error*, dan Hasil Klasifikasi Data *Testing*

No	Target	Output	Error	Klasifikasi	Keterangan Kesesuaian
1	1	1,354735	-0,35473	1	Sesuai
2	1	1,512943	-0,51294	2	Tidak Sesuai
3	1	1,340651	-0,34065	1	Sesuai
4	1	1,344766	-0,34477	1	Sesuai
5	1	1,3623	-0,3623	1	Sesuai
6	2	1,791497	0,208503	2	Sesuai
7	2	1,912018	0,087982	2	Sesuai
8	2	1,989464	0,010536	2	Sesuai
9	2	1,529515	0,470485	2	Sesuai
10	3	2,737795	0,262205	3	Sesuai
11	3	3,156291	-0,15629	3	Sesuai
12	3	2,431371	0,568629	2	Tidak Sesuai
13	3	2,697049	0,302951	3	Sesuai
14	4	4,276205	-0,2762	4	Sesuai
15	4	4,181995	-0,182	4	Sesuai
16	5	5,165478	-0,16548	5	Sesuai
17	5	4,980016	0,019984	5	Sesuai
18	5	4,953905	0,046095	5	Sesuai

Lampiran 11 *Input* dan *Output* Data dengan Normalisasi

Pembagian Data	Input									<i>Output</i>
	x_1	x_2	x_3	x_5	x_7	x_9	x_{12}	x_{13}	x_{14}	
Training	2,0283	-1,2171	0,7931	1,2367	0,4164	-1,6490	1,8174	1,4041	-1,5805	-0,884
	-0,4566	-0,5057	0,8511	0,4961	1,4665	0,2907	-0,3488	0,5037	-0,8061	-0,787
	0,4906	-0,6255	0,1353	0,6610	-0,9903	-0,8874	0,1965	0,8149	-1,0545	-0,781
	2,0745	-0,7651	-0,1549	0,8011	-0,7641	-1,8912	1,9308	1,0683	-1,2460	-0,853
	2,6037	-0,4536	-0,7006	0,5110	-0,7812	-1,9091	2,7728	0,9386	-1,0774	-0,873
	0,1303	0,0326	-0,2184	0,0012	-0,1222	-0,3231	0,2717	0,2772	-0,5218	-0,588
	0,9970	0,3907	-1,3899	-0,3712	-0,3209	-0,7956	0,9988	-0,1874	-0,1268	-0,488
	1,1971	-0,6578	0,2328	0,6717	-0,0027	-1,0244	1,3647	0,7989	-1,0421	-0,844
	1,4932	-1,0892	0,6866	1,1146	-0,9439	-1,8146	0,7788	1,2929	-1,4785	-0,829
	0,3812	-0,2602	-0,0925	0,2666	-0,8307	-0,3899	0,5791	0,2923	-0,6150	-0,762
	1,5647	0,3089	-1,5319	-0,2508	-1,1865	-0,9650	2,2846	0,1494	-0,3694	-0,802
	2,4884	-0,6114	-0,8393	0,6494	-1,4629	-1,9376	2,1575	0,9364	-1,1193	-0,877
	0,2499	-1,1044	1,0501	1,1412	-0,3508	-0,6499	0,0113	1,4550	-1,5813	-0,871
	1,0588	-1,3765	1,0542	1,4170	-1,3110	-1,5152	0,7402	1,7797	-1,8607	-0,875
	1,1251	-1,2764	0,8681	1,3156	-1,8252	-1,5117	0,9446	1,6302	-1,7413	-0,873
	1,0306	-0,8076	0,4121	0,8428	0,2362	-1,4147	0,7991	1,1158	-1,2869	-0,773
	1,7465	0,1237	-3,2623	-0,0375	-2,7071	-1,7948	1,0142	0,1929	-0,4741	-0,767
	-0,4292	-0,8791	1,3218	0,8922	1,2415	0,0666	-0,7835	1,0615	-1,7974	-0,882
	-0,1247	-0,6845	1,0434	0,6869	-0,5087	-0,0690	0,1791	0,6899	-1,8699	-0,886
	-1,2090	0,8660	-0,0190	-0,8997	-0,4406	1,3023	-1,4426	-1,0432	0,5560	0,123
	-0,1446	-0,1661	0,9079	0,1355	-0,4995	0,4523	0,7998	-0,0873	-0,3428	-0,749
	-0,8625	0,0090	0,7500	-0,0117	0,6581	0,7761	-0,7422	0,0236	-0,3657	-0,650

	-0,8350	-0,3524	0,9923	0,3656	1,4087	0,6184	-1,1338	0,4891	-0,7623	-0,781
	-0,0988	0,1535	0,0012	-0,1865	1,6299	0,0708	0,0958	-0,4058	-0,0519	-0,085
	-0,0971	0,4348	-0,7094	-0,4753	0,1118	0,1779	0,2937	-0,7739	0,2574	-0,018
	-0,7904	0,4936	-0,0479	-0,5064	0,1211	0,8594	-0,9091	-0,5074	0,1110	-0,422
	-1,0005	0,5724	0,2351	-0,6149	-0,1628	1,0582	-0,7837	-0,8757	0,3650	0,010
	0,8854	-0,5307	0,1234	0,5171	0,7059	-0,8834	0,9000	0,3813	-0,7455	-0,721
	-0,3401	0,4538	-0,5928	-0,4910	-0,8515	0,3598	-0,2750	-0,7405	0,2438	-0,167
	-0,6131	0,2773	-0,1930	-0,3040	-1,0706	0,4752	-0,4164	-0,4985	0,0389	-0,319
	-1,1287	0,8497	-0,3370	-0,9050	0,0266	1,2219	-1,0773	-1,2215	0,6648	0,349
	0,1805	0,8128	-1,4824	-0,8337	-0,5049	0,1368	0,9535	-0,8615	0,4258	-0,155
	-0,9177	0,4385	0,1003	-0,4736	0,4122	0,9458	-0,6956	-0,6928	0,2101	-0,206
	-0,8221	0,3878	0,1092	-0,4070	0,3692	0,8473	-0,7955	-0,4983	0,0725	-0,386
	-0,7345	0,1309	0,5207	-0,1551	0,1939	0,7174	-0,6409	-0,3804	-0,0811	-0,452
	-0,0561	-0,4377	1,2609	0,4636	1,3726	-0,0336	-0,2504	0,6660	-0,8990	-0,762
	-0,5247	0,2153	-0,4233	-0,2641	0,1841	0,3728	-0,7730	-0,7214	0,1563	-0,066
	-0,8362	0,2585	0,6063	-0,3055	-0,0532	0,9400	-0,3832	-0,6317	0,1222	-0,349
	-0,7408	-0,1783	0,7047	0,1500	-0,4077	0,6365	-0,7672	-0,0697	-0,3591	-0,689
	-0,7896	0,2610	0,2707	-0,2919	0,0835	0,8911	-0,6282	-0,4900	0,0312	-0,524
	-0,4202	-0,4114	0,7287	0,4018	0,9667	0,2025	-0,3495	0,3098	-0,6667	-0,708
	0,1548	-0,5965	0,5867	0,5872	0,9301	-0,3578	0,0501	0,5063	-0,8411	-0,745
	0,4145	-0,0839	-0,3544	0,0693	0,6242	-0,3359	0,4738	0,0084	-0,3793	-0,650
	0,8056	-0,4258	-0,3105	0,4089	0,3943	-0,7333	0,5369	0,2347	-0,6252	-0,758
	0,4730	-0,3489	-0,4102	0,3261	0,2062	-0,6761	0,4711	0,0755	-0,5054	-0,572
	0,9491	-1,1610	1,0155	1,1903	0,7264	-0,9386	1,0323	1,4756	-1,6075	-0,881
	1,3555	-0,7709	0,6061	0,7927	-0,2008	-1,0677	1,3603	0,8590	0,5623	0,341

	-1,2590	2,9397	-2,3066	-2,8860	-1,1118	1,6183	-1,3170	-2,4070	0,3974	0,022
	-1,2252	2,7494	-2,0163	-2,7663	-1,1441	1,6050	-0,7556	-2,5581	0,2839	-0,136
	-1,7379	4,7420	-2,8709	-4,6352	-0,3650	2,5018	-1,6733	-3,4844	0,5384	0,377
	-0,2121	0,9473	-2,8657	-0,9688	-1,9969	-0,1386	-0,5766	-1,2997	0,3667	0,263
	-0,4519	-0,7033	0,8966	0,6979	0,1216	0,1742	-0,6802	0,5863	-0,0405	-0,372
	0,2264	0,1232	-1,0887	-0,1660	0,4928	-0,3228	0,0517	-0,5931	0,0477	-0,137
	-0,6438	0,0440	0,1158	-0,0791	0,0360	0,5553	-0,4838	-0,4477	0,3764	0,124
	0,7835	0,2868	-0,3870	-0,2546	1,5697	-0,2899	1,6088	0,0539	0,5797	0,284
	-0,3441	0,0231	-0,1422	-0,0587	0,3523	0,2410	-0,3959	-0,3816	0,3337	0,170
	-1,0068	0,3100	0,6249	-0,3430	1,0102	1,1279	-0,3903	-0,6818	0,4220	0,108
	-0,6795	-0,3315	0,7300	0,3152	1,7100	0,5213	-0,9847	0,1414	0,3463	0,201
	-0,5694	-0,9920	1,2283	1,0096	0,8545	0,1027	-1,1933	1,1609	0,8517	1,165
	-0,2812	-0,8054	0,9731	0,8076	2,0399	-0,0080	-0,8658	0,8318	0,9349	1,352
	-0,8867	-0,3717	0,9585	0,3674	0,5968	0,7001	-1,0490	0,3598	1,0893	1,231
	-1,1835	0,4003	0,3211	-0,4476	0,1217	1,1819	-1,0238	-0,8828	1,1968	1,342
	-1,3279	1,4971	-0,6166	-1,5376	0,0500	1,6053	-1,4031	-1,6175	1,4587	1,462
	0,2199	-0,5547	0,5928	0,5670	1,2558	-0,2709	0,4082	0,6695	1,7266	1,918
	0,5855	0,2522	-0,3629	-0,2225	-2,0609	-0,2827	0,6193	-0,1840	1,5902	1,640
	-0,1156	-0,4398	0,2412	0,4284	-0,6608	-0,1461	-0,2420	0,2971	1,9908	1,988
	-0,1029	-0,9515	0,9833	0,9578	1,0695	-0,2524	-0,4216	1,0029	1,4059	1,814
	-0,6770	0,9681	-0,7713	-0,9245	-0,3044	0,8225	-0,5882	-0,7513	1,2665	1,354
	-0,5482	0,6042	-0,5352	-0,6049	-0,2205	0,6587	-0,1291	-0,6983	2,0229	1,937
	-1,3612	2,1359	-0,8938	-2,1104	-0,3389	1,7301	-1,2543	-1,9411	1,4704	1,620
	1,7334	-1,0991	0,2511	1,1143	-2,1819	-1,7306	1,5283	1,2318	1,2199	1,505
	-0,6340	-0,2693	0,4214	0,2425	0,1238	0,3713	-0,7429	-0,0088	1,3462	1,694

Testing	-1,0918	-0,4357	1,0179	0,4088	1,0480	0,8750	-1,3734	0,1110	1,2209	1,426
	0,8851	-0,7624	0,6025	0,7607	1,7466	-0,8018	0,7150	0,7472	1,6206	1,889
	1,0856	-1,1412	1,0140	1,1574	-1,5164	-1,1413	0,8085	1,2807	-1,4830	-0,87528
	-0,5948	-0,6112	0,9089	0,6045	0,1306	0,1602	-1,0122	0,5266	-0,8601	-0,7611
	0,2921	-0,5586	0,2381	0,5649	-1,7213	-0,5232	-0,1806	0,6034	-1,7799	-0,88544
	-0,2332	-0,6353	0,7317	0,6416	-1,2596	-0,2265	-0,6181	0,7085	-1,8653	-0,88247
	0,4370	-0,8485	1,0869	0,8877	-1,2963	-0,5516	0,7813	1,2387	-1,3732	-0,86982
	-0,2044	-0,1018	0,3440	0,0925	2,4869	0,1557	-0,3473	0,0760	-0,4273	-0,56008
	-0,8120	0,1794	0,2513	-0,1791	0,4747	0,6903	-0,8982	-0,2514	-0,1536	-0,4731
	-0,1973	0,4148	-0,8177	-0,4122	-0,8011	0,1496	-0,1059	-0,3702	-0,0056	-0,41721
	0,7305	-0,2344	-0,2884	0,2289	-0,0521	-0,5854	0,4968	0,1687	-0,5290	-0,74914
	-0,5293	-0,2353	0,7701	0,2148	-0,0137	0,5285	-0,0171	0,0220	0,4500	0,12285
	-0,4442	-0,1723	0,4838	0,1491	0,6702	0,4136	-0,0726	-0,0889	0,5355	0,424871
	1,4857	-1,1343	0,8207	1,1510	0,1454	-1,2707	1,5872	1,3027	0,2806	-0,09829
	1,8597	-1,3142	0,9016	1,3393	0,1998	-1,7077	1,9761	1,5638	0,2432	0,093444
	0,0138	-0,0413	0,5882	0,0844	0,6591	0,1621	0,7038	0,5046	1,0929	1,233093
	-0,4573	0,5378	-0,8113	-0,5040	-0,6109	0,3501	-0,7360	-0,4818	0,9677	1,165104
	0,0649	0,1955	-1,1951	-0,2171	-1,5705	-0,2353	-0,0068	-0,4323	1,7417	1,874866
	-0,4280	0,0715	-0,1905	-0,1026	-0,5409	0,2836	-0,5019	-0,4220	1,4714	1,741021
	-0,2669	-0,1530	0,2756	0,1287	-1,1655	0,1962	0,1846	-0,1904	1,4853	1,722178