

Your complimentary use period has ended.
Thank you for using PDF Complete.

Click Here to upgrade to Unlimited Pages and Expanded Features

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



Oleh
Sinta Deni Ariati
NIM 11305141003

PROGRAM STUDI MATEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2015

BAB I PENDAHULUAN

A. Latar Belakang

Kanker kolorektal atau disebut juga kanker usus besar adalah tumbuhnya sel tumor yang ganas di dalam kolon dan rektum. Kanker ini merupakan salah satu jenis penyakit yang mematikan. Menurut Varalakshmi (2013:1), kanker kolorektal merupakan jenis kanker yang menempati urutan ketiga penyebab kematian terbanyak di dunia yaitu mencapai 1,23 juta kematian per tahun (9,7%) setelah kanker paru yang mencapai 1,61 juta kematian per tahun (12,7%) dan kanker payudara yang mencapai 1,31 juta kematian per tahun (10,9%).

Faktor penyebab kanker kolorektal yang paling utama adalah usia. Menurut Zahari (2010:98), sebanyak 6%-8% dari kasus kanker kolorektal terjadi sebelum usia 40 tahun dan meningkat setelah usia 50 tahun, dengan puncak insidensi pada orang yang berumur 70-80 tahun. Di negara maju hanya 3% dari kasus ini yang ditemukan pada pasien berusia di bawah 40 tahun sedangkan peningkatan tajam terjadi pada pasien berusia di atas 50 tahun. Begitu pula dengan di Indonesia, berdasarkan penelitian Patologi Anatomi FKUI tahun 2003-2007, jumlah pasien kanker kolorektal di atas usia 40 tahun mencapai 71,83% (Muharomah, 2014:2). Namun, selain usia ada pula beberapa hal yang diduga kuat berpotensi memunculkan penyakit ganas ini, yaitu: cara diet yang salah (terlalu banyak mengonsumsi makanan tinggi lemak dan protein, serta rendah serat), obesitas, pernah terkena kanker usus besar, berasal

KLASIFIKASI STADIUM KANKER KOLOREKTAL MENGUNAKAN MODEL *NEURO FUZZY* BERBASIS *GRAPHICAL USER INTERFACE* (GUI)

Oleh:
Sinta Deni Ariati
NIM 11305141003

ABSTRAK

Pada penelitian ini dijelaskan mengenai prosedur, hasil aplikasi, dan ketepatan hasil klasifikasi stadium kanker kolorektal menggunakan model *Neuro Fuzzy* (NF) berbasis *Graphical User Interface* (GUI). Kanker kolorektal adalah tumbuhnya sel tumor yang ganas di dalam kolon dan rektum. Di lain pihak, *Neuro Fuzzy* merupakan salah satu model klasifikasi yang menggabungkan *Neural Network* (NN) dengan logika *fuzzy*. Dalam melakukan klasifikasi stadium kanker kolorektal digunakan program GUI yang ada pada *software Matrix Laboratory* (MATLAB).

Prosedur pemodelan NF untuk klasifikasi stadium kanker kolorektal diawali dengan prosedur pemodelan NN yang meliputi penentuan variabel *input* dan target jaringan, pembagian data menjadi data *training* dan *testing*, normalisasi data, dan perancangan model NN terbaik. Variabel *input* yang digunakan adalah 14 fitur hasil ekstraksi citra foto kolorektal sedangkan target jaringan adalah keterangan kondisi dari citra yaitu normal, stadium I, stadium II, stadium III, dan stadium IV. *Input* optimal yang diperoleh saat perancangan model NN terbaik kemudian digunakan sebagai *input* pemodelan NF. Prosedur pemodelan NF selanjutnya yaitu pengelompokan (*clustering*) data sebanyak 5 *cluster* dengan metode *Fuzzy C-Means*, pembelajaran NN yang berhubungan dengan bagian anteseden pada aturan inferensi *fuzzy*, dan penyederhanaan bagian konsekuen dengan melakukan eliminasi *input* serta mencari nilai koefisien konsekuen masing-masing *cluster* dengan metode *Least Square Estimator*. Algoritma pembelajaran yang digunakan adalah algoritma *backpropagation* dan model yang digunakan dalam aturan inferensi *fuzzy* adalah model Sugeno orde-1.

Berdasarkan prosedur pemodelan NF, diperoleh model NF terbaik dengan 9 variabel *input* pada setiap aturan inferensi *fuzzy*. Selanjutnya model NF terbaik tersebut digunakan dalam membangun sebuah program tampilan GUI untuk memudahkan pengguna dalam melakukan klasifikasi stadium kanker kolorektal. Model NF berbasis GUI untuk klasifikasi stadium kanker kolorektal menghasilkan nilai sensitivitas, spesifisitas, dan akurasi masing masing 98,04%, 94,44%, dan 97,10% untuk data *training* serta 94,12%, 83,33%, dan 60,87% untuk data *testing*.

Kata Kunci : *neuro fuzzy*, klasifikasi, kanker kolorektal, *Graphical User Interface*

vii

dari keluarga yang memiliki riwayat kanker usus besar, pernah memiliki polip di usus, jarang melakukan aktivitas fisik, sering terpapar bahan pengawet makanan maupun pewarna yang bukan untuk makanan, dan merokok (Diananda, 2008:128).

Gejala yang sering dirasakan oleh para pasien kanker kolorektal yaitu terjadinya pendarahan pada usus besar yang ditandai dengan ditemukannya darah pada feses saat buang air besar. Setelah diketahui adanya pendarahan tersebut maka pasien sebaiknya segera melakukan pemeriksaan untuk mendeteksi dari mana sumber darah tersebut, apakah dari rektum, kolon, atau bagian lainnya karena adanya darah pada feses dapat juga disebabkan oleh penyakit lain yang bukan kanker kolorektal. Hal inilah yang membuat banyak pasien kanker kolorektal tidak menyadari bahwa dirinya terkena penyakit kanker kolorektal dan datang ke rumah sakit setelah dalam keadaan stadium lanjut dan mengakibatkan upaya pengobatan menjadi sulit dilakukan. Padahal, seperti dikatakan Ketua Perhimpunan Spesialis Penyakit Dalam Indonesia, dr. Aru Sudoyo, SpPD, KHOM., kunci utama keberhasilan penanganan kanker usus besar adalah ditemukannya kanker pada stadium dini (Diananda, 2008:126).

Deteksi dini memegang peranan penting untuk menurunkan angka kematian akibat kanker kolorektal. Salah satu cara untuk mendeteksi adanya sel abnormal kanker kolorektal adalah dengan melakukan *endoscopy* berupa *colonoscopy*. Proses *colonoscopy* dilakukan dengan memasukkan kamera kecil

Metode klasifikasi yang terus berkembang dewasa ini adalah menggunakan teknik *soft computing*. *Soft computing* merupakan teknik komputasi yang meniru akal manusia dan memiliki kemampuan untuk menalar dan belajar pada lingkungan yang penuh dengan ketidakpastian dan ketidaktepatan. Komponen utama pembentuk *soft computing* adalah *neural network* dan *logika fuzzy* (Sri Kusumadewi & Sri Hartati, 2010:1).

Komponen pembentuk *soft computing* yang pertama yaitu *Neural Network*. *Neural Network* adalah suatu sistem pemrosesan informasi yang memiliki karakteristik kinerja tertentu yang analog dengan jaringan syaraf biologi (Fausett, 1994:3). Seperti halnya syaraf manusia, *neural network* terbentuk dari struktur dasar neuron yang terhubung antara satu dengan yang lain membentuk suatu jaringan. Jaringan pada *neural network* dilatih menggunakan suatu algoritma pembelajaran untuk melakukan fungsi tertentu dengan menyesuaikan nilai-nilai dari koneksi (bobot) antar unsur-unsur. Penelitian terkait *neural network* untuk klasifikasi telah dilakukan, diantaranya oleh Kyung-Joong Kim & Sung-Bae Cho (2003) yang menggunakan model *Evolutionary Artificial Neural Network* (EANN) untuk mengklasifikasi kanker kolon ke dalam kriteria normal dan kanker berdasarkan informasi DNA *microarray* dengan tingkat akurasi 94% dan penelitian oleh Paulin &

□

penalarannya. *Neuro fuzzy* dirancang untuk merealisasikan proses penalaran *fuzzy*, dimana bobot-bobot yang terhubung pada *neural network* terhubung dengan parameter-parameter penalaran *fuzzy*. Penelitian terkait dengan *neuro fuzzy* telah dilakukan, diantaranya penelitian oleh Nwoye, *et al* (2005) yang menggunakan metode *Fuzzy Neural Expert Machine* (F-NEM) untuk mengklasifikasi 204 citra polip kolorektal hasil biopsi ke dalam kriteria normal adenoma dan malignant adenokarsinoma dengan tingkat akurasi 96,5% dan penelitian oleh Nauck, *et al* (2007) yang menggunakan sistem *neuro fuzzy* NEFLCLASS (*NEuro Fuzzy CLASSification*) untuk mengklasifikasi kanker payudara berdasarkan data *Winconsin Breast Cancer* (WBC) ke dalam kriteria *benign* dan *malignant* dengan tingkat akurasi 92,7%.

Perhitungan yang rumit dan kompleks seringkali ditemui dalam proses klasifikasi menggunakan *neuro fuzzy*. Oleh karena itu, perhitungan perlu dilakukan dengan bantuan suatu perangkat lunak (*software*) seperti *Matrix Laboratory* (MATLAB). Pada *software* MATLAB dapat dibangun sebuah program klasifikasi dimana model klasifikasi berasal dari prosedur *neuro fuzzy* dan visualisasinya didesain menggunakan *Graphical User Interface* (GUI). GUI merupakan tampilan grafis hasil dari GUIDE yang ada pada MATLAB yang bertujuan untuk memudahkan pengguna dalam berinteraksi.

Berdasarkan uraian latar belakang sebelumnya, penulis melakukan penelitian yang berjudul "Klasifikasi Stadium Kanker Kolorektal Menggunakan Model *Neuro Fuzzy* Berbasis *Graphical User Interface* (GUI)".

□

Santhakumaran (2010) yang menggunakan metode *backpropagation* untuk klasifikasi kanker payudara berdasarkan data *Winconsin Breast Cancer* (WBC) ke dalam kriteria *benign* dan *malignant* dengan tingkat akurasi 99,28%.

Komponen pembentuk *soft computing* berikutnya yaitu logika *fuzzy*. Logika *fuzzy* adalah suatu cara untuk memetakan suatu ruang *input* ke dalam suatu ruang *output* (Prabowo Pudjo W. & Rahmadya Trias H., 2012:2). Logika *fuzzy* merupakan representasi ketidakpastian, ketidakjelasan, ketidaktepatan yang dikonstruksikan dengan *IF-THEN rules*. Logika *fuzzy* merupakan perkembangan dari logika klasik. Pada logika *fuzzy* nilai kebenaran suatu *item* berada pada interval [0,1], sedangkan pada logika klasik hanya ada dua kemungkinan nilai kebenaran yaitu 0 atau 1. Logika *fuzzy* membangun struktur deskripsi kualitatif penggunaan bahasa sehari-hari sehingga mudah digunakan. Penelitian terkait logika *fuzzy* untuk klasifikasi telah dilakukan, diantaranya oleh Al-Daoud (2010) yang menggunakan jaringan *fuzzy* termodifikasi untuk mengklasifikasi data fitur kanker payudara ke dalam kriteria *benign* dan *malignant* dengan tingkat akurasi 97% dan penelitian oleh Iskandar (2013) yang menggunakan sistem *fuzzy* MADM (*Multi Attribute Decision Making*) untuk mengklasifikasi penyakit kanker payudara berdasarkan data-data laborat dengan tingkat akurasi 90%.

Di lain pihak, *Neuro fuzzy* merupakan penggabungan antara *neural network* dan logika *fuzzy*. *Neuro fuzzy* menggabungkan kelebihan *neural network* dalam melakukan pembelajaran dan logika *fuzzy* dengan kemampuan

□

Penelitian ini diharapkan dapat memberikan alternatif klasifikasi stadium kanker kolorektal dan bermanfaat bagi perkembangan ilmu matematika.

B. Rumusan Masalah

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

1. Bagaimana prosedur pemodelan *Neuro Fuzzy* untuk klasifikasi stadium kanker kolorektal?
2. Bagaimana hasil aplikasi model *Neuro Fuzzy* untuk klasifikasi stadium kanker kolorektal?
3. Bagaimana proses dan ketepatan hasil klasifikasi stadium kanker kolorektal menggunakan model *Neuro Fuzzy* berbasis *Graphical User Interface* (GUI)?

C. Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka tujuan penulisan penelitian ini adalah:

1. Menjelaskan prosedur pemodelan *Neuro Fuzzy* untuk klasifikasi stadium kanker kolorektal.
2. Mendeskripsikan hasil aplikasi model *Neuro Fuzzy* untuk klasifikasi stadium kanker kolorektal.
3. Menjelaskan proses dan ketepatan hasil klasifikasi stadium kanker kolorektal menggunakan model *Neuro Fuzzy* berbasis *Graphical User Interface* (GUI).

□

Sebagai salah satu bahan dalam mempelajari model *neuro fuzzy* dan diharapkan penelitian ini dapat dijadikan sebagai referensi untuk penelitian selanjutnya. Selain itu, penelitian ini diharapkan dapat membantu mempermudah dan mempercepat dalam mendiagnosis penyakit kanker kolorektal yang diderita pasien dengan biaya yang lebih murah.

2. Bagi perpustakaan Universitas Negeri Yogyakarta

Penulisan skripsi ini bermanfaat sebagai koleksi bahan pustaka yang bermanfaat bagi Universitas Negeri Yogyakarta pada umumnya, dan mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam pada khususnya.

□

- a. Usia, umumnya kanker kolorektal menyerang lebih sering pada usia tua. Lebih dari 90 persen penyakit ini menimpa penderita di atas usia 50 tahun. Walaupun pada usia yang lebih muda dari 50 tahun pun dapat saja terkena.
- b. Polip kolorektal, adalah pertumbuhan tumor pada dinding sebelah dalam usus besar dan rektum. Sering terjadi pada usia di atas 50 tahun. Kebanyakan polip ini adalah tumor jinak, tetapi sebagian dapat berubah menjadi kanker. Menemukan dan mengangkat polip ini dapat menurunkan risiko terjadinya kanker kolorektal.
- c. Riwayat kanker kolorektal pada keluarga, bila keluarga dekat yang terkena (orang tua, kakak, adik, atau anak), maka risiko untuk terkena kanker ini lebih besar, terutama bila keluarga yang terkena kanker ini pada usia muda.
- d. Kelainan genetik, perubahan pada gen tertentu akan meningkatkan risiko terkena kanker kolorektal. Menurut Prof. dr. Siti Boedina Kresno, SpPK(K), ada tiga golongan gen suseptibilitas kanker kolorektal. Beberapa di antara gen yang paling penting yaitu APC, MUTYH (bentuk *familial polyposis*), dan gen sindrom Lynch (MLH1, MSH2, MSH6, dan PMS2). Gen-gen ini bertanggungjawab atas >5% semua kasus kanker kolorektal, tetapi sebagian besar yang terkena adalah orang usia muda atau usia produktif.

□

BAB II KAJIAN TEORI

A. Kanker Kolorektal

Kanker kolorektal atau kanker usus besar adalah tumbunya sel tumor yang ganas di dalam kolon dan rektum. Kanker ini merupakan penyakit kanker nomor tiga yang paling banyak terjadi di dunia setelah kanker paru dan kanker payudara. Jika kanker ini ditemukan pada stadium I, peluang penderita untuk hidup hingga lima tahun mencapai 85-95 persen. Sementara bila ditemukan pada stadium II, peluang itu mencapai 60-80 persen, pada stadium III sekitar 30-60 persen, dan stadium IV sekitar 25 persen (Diananda, 2008:127).

Kanker kolorektal seringkali tidak menunjukkan gejala awal yang khas sehingga membuat banyak penderita kanker kolorektal datang ke rumah sakit dalam keadaan stadium lanjut dan upaya pengobatan menjadi sulit. Padahal, seperti dikatakan Ketua Perhimpunan Spesialis Penyakit Dalam Indonesia, dr. Aru Sudoyo, SpPD, KHOM., kunci utama keberhasilan penanganan kanker usus besar adalah ditemukannya kanker dalam stadium dini, sehingga terapi dapat dilaksanakan secara bedah kuratif. Oleh sebab itu, deteksi dini dan diagnosis memegang peranan penting guna menurunkan angka kematian akibat kanker kolorektal.

1. Penyebab

Beberapa faktor risiko penyebab kanker kolorektal antara lain sebagai berikut (Diananda, 2008:128):

□

- e. Pernah menderita penyakit sejenis, dapat terserang kembali dengan penyakit yang sama untuk kedua kalinya. Demikian pula wanita yang memiliki riwayat kanker indung telur, kanker rahim, dan kanker payudara memiliki risiko yang tinggi untuk terkena kanker ini.
- f. Radang usus besar, berupa *colitis ulceratif* atau penyakit Crohn yang menyebabkan inflamasi atau peradangan pada usus untuk jangka waktu lama, akan meningkatkan risiko terserang kanker kolorektal.
- g. Diet, makanan tinggi lemak (khususnya lemak hewan) dan rendah kalsium, folat dan rendah serat, jarang memakan sayuran dan buah-buahan, sering minum alkohol, akan meningkatkan risiko terkena kanker kolorektal.
- h. Merokok, dapat meningkatkan risiko terjadinya kanker ini. Dalam buku Panduan Adenokarsinoma Kolorektal disebutkan bahwa meskipun penelitian awal tidak menunjukkan hubungan merokok dengan kejadian kanker usus besar, namun penelitian terbaru menunjukkan, perokok jangka lama (30-40 tahun) mempunyai risiko sekitar 1,5-3 kali. Diperkirakan, satu dari lima kasus kanker usus besar di Amerika Serikat bisa diatributkan kepada perokok. Penelitian kohort dan kasus-kontrol dengan desain yang baik menunjukkan bahwa merokok berhubungan dengan kenaikan risiko terbentuknya adenoma dan juga kenaikan risiko perubahan adenoma menjadi kanker usus besar.

□

- a. Perut terasa penuh meskipun telah buang air besar, pada feses saat buang air besar,
- b. Perubahan pada fungsi usus (diare atau sembelit) tanpa sebab yang jelas selama lebih dari enam minggu,
- c. Penurunan berat badan tanpa sebab yang jelas,
- d. Rasa sakit di perut,
- e. Perut masih terasa penuh meskipun telah buang air besar,
- f. Rasa lelah yang terus menerus,
- g. Kadang-kadang kanker dapat menjadi penghalang dalam usus besar yang tampak pada beberapa gejala seperti sembelit, rasa sakit, dan rasa kembung di perut.

3. Deteksi

Karena tidak dapat diketahui penyebab pasti dari kanker kolorektal, maka yang paling penting untuk dilakukan adalah deteksi dini. Deteksi dini berupa *screening* yang diperlukan untuk mengetahui ada atau tidaknya kanker kolorektal adalah dengan menjalani (Diananda, 2008:131):

- a. *Fecal occult blood test* (FOBT), kanker maupun polip dapat menyebabkan pendarahan dan FOBT dapat mendeteksi adanya darah pada tinja. Bila tes ini mendeteksi adanya darah, harus dicari dari mana

11

memeriksa bagian rektum. Bila ada tumor di rektum maka akan teraba dan diketahui melalui pemeriksaan ini.

4. Klasifikasi Stadium Kanker Kolorektal

Klasifikasi stadium kanker kolorektal yaitu mengelompokkan kanker kolorektal ke dalam stadium yang tepat, berdasarkan ciri-ciri tertentu. Ciri-ciri itu dapat berupa kedalaman invasi sel kanker pada dinding usus besar, tingkat penyebaran pada kelenjar getah bening, dan adanya metastasis, yang diperoleh dari hasil *colonoscopy* baik pada pasien kanker kolorektal maupun bukan pasien kanker kolorektal.

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

- a. Stadium 0 (*Carcinoma in Situ*) : kanker hanya pada lapisan terdalam dari kolon dan rektum.
- b. Stadium I : sel kanker telah tumbuh pada dinding dalam kolon atau rektum tetapi belum menembus ke luar dinding.
- c. Stadium II : sel kanker telah menyebar ke dalam lapisan otot kolon atau rektum, tetapi sel kanker di sekitarnya belum menyebar ke kelenjar getah bening.
- d. Stadium III : sel 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 ke bagian lain dari usus besar, seperti hati, paru-paru, atau tulang.

12

sumber darah tersebut, apakah dari rektum, kolon, atau bagian anus lainnya dengan pemeriksaan yang lain. Penyakit wasir juga dapat menyebabkan adanya darah dalam tinja.

- b. *Sigmoidoscopy*, yaitu suatu pemeriksaan dengan alat berupa kabel seperti kabel kopling yang ujungnya terpasang alat petunjuk yang diberi cahaya dan bisa meneropong yang disebut *sigmoidoscope*. Alat ini dimasukkan melalui lubang dubur ke dalam rektum sampai kolon sigmoid, sehingga dinding dalam rektum dan kolon sigmoid dapat terlihat. Bila ditemukan adanya polip, dapat sekalian diangkat.
- c. *Colonoscopy*, sama seperti *sigmoidoscopy*, namun menggunakan kabel yang lebih panjang (*colonoscope*), sehingga seluruh bagian usus besar dapat diteropong dan diperiksa. Setelah hasil kanker terdeteksi maka dilakukan penentuan stadium kanker kolorektal.
- d. *Double-contrast barium enema*, yakni pemeriksaan radiologi dengan sinar-X pada kolon dan rektum. Penderita diberikan enema dengan larutan barium dan udara yang dipompakan ke dalam rektum, kemudian difoto. Seluruh lapisan dinding dalam kolon dapat dilihat apakah normal atau terdapat kelainan.
- e. Colok dubur, yakni pemeriksaan yang sangat sederhana dan dapat dilakukan oleh semua dokter, yaitu dengan memasukkan jari yang sudah dilapisi sarung tangan dan zat lubrikasi ke dalam dubur kemudian

12

B. Pengolahan Citra Digital

Citra digital adalah sebuah gambar dalam tampilan dua dimensi yang disimpan dalam bentuk file sehingga dapat diolah menggunakan komputer (Gonzales & Woods, 2002:1). Citra digital terbagi menjadi tiga jenis yakni citra warna RGB (*Red Green Blue*), citra berskala keabuan (*grayscale*), dan citra biner. Citra warna RGB dideskripsikan oleh banyak warna merah (R, *red*), hijau (G, *green*), dan biru (B, *blue*) dengan rentang intensitas setiap warna adalah 0 sampai 255. Citra *grayscale* adalah citra yang nilai pikselnya merupakan bayangan abu-abu yang memiliki nilai intensitas 0 (hitam) sampai 255 (putih). Sedangkan citra biner adalah citra dengan setiap pikselnya hanya dinyatakan dengan sebuah nilai dari dua kemungkinan, yaitu 0 menyatakan hitam dan 1 menyatakan putih (Sianipar, 2013:200). Piksel adalah istilah yang paling banyak digunakan untuk menggambarkan suatu elemen (titik) terkecil dari citra digital. Pengolahan citra digital yang biasa dikenal dengan *image processing* adalah proses mengolah piksel-piksel dalam citra digital untuk suatu tujuan.

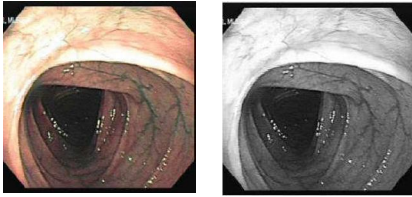
Data *input* yang digunakan pada penelitian ini berupa citra warna RGB yang diperoleh dari proses *colonoscopy*. Citra tersebut dapat diekstraksi sehingga diperoleh beberapa fitur yang akan digunakan sebagai variabel *input* dalam proses klasifikasi. Ekstraksi citra atau juga dikenal dengan sebutan *indexing*, merupakan suatu teknik pengambilan sifat-sifat khusus (fitur) dari sebuah citra.

13

2.1 (a) unakukan menggunakan MATLAB R2010a dengan perintah sebagai berikut:

$a = imread('a1.jpg')$ untuk membaca gambar

$b = rgb2gray(a)$ untuk transformasi dari citra RGB ke citra abu-abu



Gambar 2.1 Citra RGB (a) dan Citra abu-abu (b)

Langkah selanjutnya setelah transformasi citra adalah ekstraksi citra.

Terdapat banyak teknik ekstraksi gambar, salah satunya adalah dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). GLCM adalah suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra (Usman Ahmad, 2005:233). Ekstraksi citra dilakukan dengan metode GLCM untuk menghasilkan 14 fitur ekstraksi. Fungsi bawaan MATLAB R2010a

1. Energy

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

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

dengan,

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

2. Contrast

Nilai *contrast* adalah ukuran intensitas kontras antara suatu piksel dan piksel lainnya dari keseluruhan citra (Sharma & Mukharjee, 2013:331).

Rumus kontras (C) adalah sebagai berikut (Kalas, 2010:20):

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

dengan,

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

3. Correlation

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

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

dengan,

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

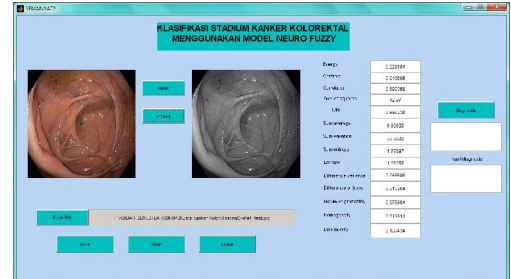
$\mu_x = \sum_i \sum_j i p(i,j)$,

$\mu_y = \sum_i \sum_j j p(i,j)$,

$\sigma_x = \sum_i \sum_j (i - \mu_x)^2 p(i,j)$,

hanya mampu memberikan 5 fitur saja yaitu *entropy*, *contrast*, *correlation*, *energy*, dan *homogeneity*.

Fungsi bawaan MATLABR2010a tersebut dapat dikembangkan untuk mendapatkan 9 fitur lainnya dengan kode tertentu. *Graphical User Interface* (GUI) coba dibangun untuk dapat digunakan dalam proses ekstraksi. Tampilan GUI untuk ekstraksi dari citra RGB menjadi citra *grayscale* dan hasil ekstraksinya adalah sebagai berikut (Gambar 2.2).



Gambar 2.2 Tampilan GUI

Fitur-fitur hasil ekstraksi yang diperoleh adalah *energy*, *contrast*, *correlation*, *sum of square variance*, *Inverse Difference Moment* (IDM), *sum average*, *sum variance*, *sum entropy*, *entropy*, *difference variance*, *difference entropy*, *maximum probability*, *homogeneity* dan *dissimilarity*.

$$\sigma_y = \sum_i \sum_j (i - \mu_y)^2 p(i,j)$$

4. Sum of square variance

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

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

dengan,

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

μ = rata-rata $\sum_i \sum_j p(i,j)$.

5. Inverse Difference Moment (IDM)

IDM adalah ukuran dari homogenitas lokal. Nilai IDM tinggi ketika level abu-abu (*grey level*) lokal seragam dan invers dari GLCM tinggi.

(Mohanaiyah, et al., 2013:2). Rumus IDM adalah sebagai berikut (Harralick, et al., 1973:619):

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

dengan,

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

6. Sum average

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

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

dengan,

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

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

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

dengan,
 $p_{x+y(k)} = \prod_{i=1}^{N_g} \prod_{j=1}^{N_g} p_{i,j} ; i+j = k; k = 2,3,\dots,2N_g,$
 N_g = banyak level abu-abu yang diperoleh dari citra.

8. Sum variance

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

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

dengan,
 SE = jumlah entropi,
 $p_{x+y(k)} = \prod_{i=1}^{N_g} \prod_{j=1}^{N_g} p_{i,j} ; i+j = k; k = 2,3,\dots,2N_g,$
 N_g = banyak level abu-abu yang diperoleh dari citra.

9. Entropy

Entropy adalah suatu ukuran keacakan. Entropy menunjukkan jumlah informasi dari sebuah citra yang dibutuhkan untuk mengkompres citra (Mohanaiyah, et al., 2013:2). Rumus entropy (EN) dari suatu citra adalah sebagai berikut (Harralick, et al., 1973:619):

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

dengan,

13. Homogeneity

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

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

dengan,
 $p_{i,j}$ = piksel di lokasi baris ke-i dan kolom ke-j.

14. Dissimilarity

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

$$D = \sum_i \sum_j p_{i,j} |i-j| \quad (2.14)$$

dengan,
 $p_{i,j}$ = piksel di lokasi baris ke-i dan kolom ke-j.

C. Neural Network (NN)

Neural Network (NN) adalah suatu sistem pemrosesan informasi yang memiliki karakteristik kinerja tertentu yang analog dengan jaringan syaraf biologi (Fausett, 1994:3). Seperti halnya syaraf manusia, neural network terbentuk dari struktur dasar neuron yang terhubung antara satu dengan yang lain. Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian neural network (Siang, 2005:23). Setiap neuron terhubung dengan neuron yang lain melalui suatu pola komunikasi terarah, masing-masing dengan bobot yang bersesuaian. Bobot merupakan informasi yang

$p_{i,j}$ = piksel di lokasi baris ke-i dan kolom ke-j.

10. Difference variance

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

$$DV = var(p_{x-y(k)}) \quad (2.10)$$

dengan,
 $p_{x-y(k)} = \prod_{i=1}^{N_g} \prod_{j=1}^{N_g} p_{i,j} ; i-j = k; k = 0,1,2,\dots,(N_g-1),$
 N_g = banyak level abu-abu yang diperoleh dari citra.

11. Difference entropy

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

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

dengan,
 $p_{x-y(k)} = \prod_{i=1}^{N_g} \prod_{j=1}^{N_g} p_{i,j} ; i-j = k; k = 0,1,2,\dots,(N_g-1),$
 N_g = banyak level abu-abu yang diperoleh dari citra.

12. Maximum probability

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

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

dengan,
 $p_{i,j}$ = piksel di lokasi baris ke-i dan kolom ke-j.

digunakan oleh jaringan untuk menyelesaikan sebuah permasalahan (Fausett, 1994:3).

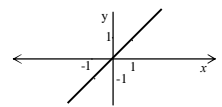
Lapisan-lapisan penyusun neural network dibagi menjadi 3 yaitu lapisan input (input layer), lapisan tersembunyi (hidden layer), dan lapisan output (output layer). Lapisan input menerima input dari luar, input yang diberikan berupa gambaran dari suatu permasalahan. Lapisan tersembunyi menghubungkan lapisan input dan lapisan output, keluaran dari lapisan ini tidak dapat diamati secara langsung. Keluaran dari lapisan output merupakan hasil dari neural network terhadap suatu permasalahan yang diterima lapisan input (Siang, 2005:24).

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

1. Fungsi Aktivasi

Mengaktifkan neural network sama halnya mengaktifkan setiap neuron yang digunakan dalam jaringan. Untuk mengaktifkan neuron, digunakan fungsi tertentu. Fungsi-fungsi tersebut disebut dengan fungsi aktivasi. Fungsi aktivasi akan menentukan output suatu neuron yang akan dikirim ke neuron lain (Fausett, 1994:17). Terdapat banyak fungsi aktivasi yang dapat digunakan sebagai pengaktif neural network, antara lain sebagai berikut (Fausett, 1994: 17-19).

$$y = f(x) = x, x \in \mathbb{R} \quad (2.15)$$



Gambar 2.3 Fungsi Identitas

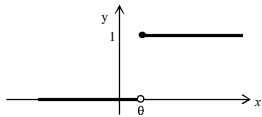
Pada Matlab R2010a, sintaks untuk fungsi aktivasi ini adalah *purelin*.

b. Fungsi Undak Biner (Hard Limit)

Fungsi undak biner digunakan untuk mengkonversikan *input* suatu variabel yang bernilai kontinu ke suatu *output* biner (0 atau 1). Fungsi ini sering digunakan pada jaringan dengan lapisan tunggal. Fungsi undak biner (dengan *threshold* θ) dirumuskan sebagai berikut:

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

Grafik fungsi undak biner tampak pada Gambar 2.4 berikut:



Gambar 2.4 Fungsi Undak Biner

Pada Matlab R2010a, sintaks untuk fungsi aktivasi ini adalah *hardlim*.

output fungsi sigmoid bipolar adalah -1 sampai 1. Fungsi sigmoid bipolar dirumuskan sebagai berikut:

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

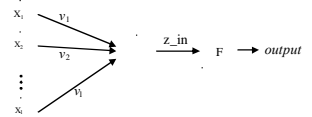
Pada Matlab R2010a, sintaks untuk fungsi aktivasi ini adalah *tansig*.

2. Arsitektur

Pada *neural network* neuron-neuron tersusun dalam lapisan. Hubungan antar neuron dalam lapisan mengikuti pola tertentu tergantung pada arsitektur jaringan syarafnya. Pada dasarnya ada tiga macam arsitektur NN, yaitu:

a. NN dengan Lapisan Tunggal (Single Layered Network)

NN dengan lapisan tunggal hanya terdiri dari *input layer* yang langsung terhubung ke *output layer* dan tidak ada lapisan yang tersembunyi. NN dengan lapisan tunggal adalah jaringan umpan maju atau *feedforward* (Haykin, 1998:21). Gambar 2.6 merupakan salah satu contoh arsitektur NN dengan lapisan tunggal.



Gambar 2.6 Arsitektur NN Lapisan Tunggal

b. NN dengan Banyak Lapisan (Multi Layer Network)

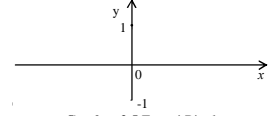
NN dengan banyak lapisan memiliki satu atau lebih lapisan tersembunyi. Lapisan *input* pada NN dengan banyak lapisan akan

c. Fungsi Bipolar (Symmetric Hard Limit)

Fungsi bipolar hampir sama dengan fungsi undak biner, perbedaannya terletak pada nilai *output* yang dihasilkan. Nilai *output* fungsi bipolar berupa 1 dan -1. Fungsi bipolar dirumuskan sebagai berikut (Khalil & Al-Kazzaz, 2009:17):

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

Grafik fungsi bipolar dapat dilihat pada Gambar 2.5 berikut :



Gambar 2.5 Fungsi Bipolar

Pada Matlab R2010a, sintaks untuk fungsi aktivasi ini adalah *hardlims*.

d. Fungsi Sigmoid Biner

Fungsi sigmoid biner sering digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Nilai fungsi ini terletak antara 0 dan 1. Fungsi sigmoid biner dirumuskan sebagai berikut:

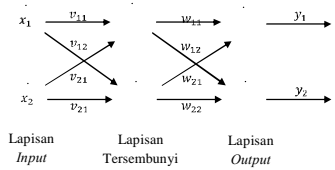
$$y = f(x) = \frac{1}{1 + e^{-x}}, x \in \mathbb{R} \quad (2.18)$$

Pada Matlab R2010a, sintaks untuk fungsi aktivasi ini adalah *logsig*.

e. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, perbedaannya terletak pada rentang nilai *output*nya. Rentang nilai

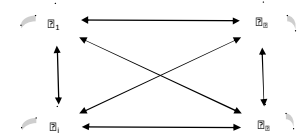
memberikan sinyal pada lapisan tersembunyi dan kemudian lapisan tersembunyi akan meneruskan sinyal tersebut langsung pada lapisan *output* atau lapisan tersembunyi lainnya jika terdapat lebih dari satu lapisan tersembunyi pada jaringan tersebut. Gambar 2.7 merupakan salah satu contoh arsitektur NN dengan banyak lapisan.



Gambar 2.7 Arsitektur NN dengan Banyak Lapisan

c. NN dengan Lapisan Kompetitif

NN dengan lapisan kompetitif memiliki bentuk yang berbeda dengan NN lapisan tunggal dan banyak lapisan. Pada NN dengan lapisan kompetitif, antar neuron dapat saling dihubungkan. Gambar 2.8 merupakan salah satu contoh arsitektur ini.



Gambar 2.8 Arsitektur NN dengan Lapisan Kompetitif

oleh lingkungan jaringan (Suyanto, 2008:74). Perubahan yang terjadi selama proses pembelajaran adalah perubahan nilai bobot. Proses pembelajaran bertujuan untuk melakukan pengaturan terhadap bobot yang ada pada *neural network*, sehingga diperoleh bobot akhir yang tepat sesuai dengan pola data yang dilatih (Sri Kusumadewi & Sri Hartati, 2010: 84). Nilai bobot akan bertambah jika informasi yang diberikan ke suatu neuron mampu tersampaikan ke neuron yang lain. Sebaliknya, nilai bobot akan berkurang jika informasi yang diberikan ke suatu neuron tidak tersampaikan ke neuron lainnya.

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

a. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

Pembelajaran tak terawasi tidak memerlukan target *output* dan jaringan dapat melakukan *training* sendiri untuk mengekstrak fitur dari variabel independen (Sarle, 1994:6). Pada metode ini, tidak dapat ditentukan hasil seperti apa yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu sesuai dengan nilai *input* yang diberikan. Tujuan pembelajaran

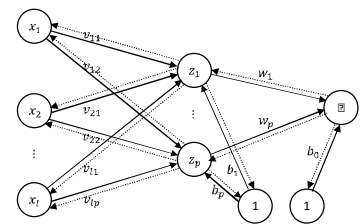
Algoritma pembelajaran *backpropagation* biasanya dirancang untuk operasi pada *multilayer neural network*. Pembelajaran *backpropagation* terdiri dari 3 fase yaitu perambatan maju (*feedforward*), perambatan mundur (*backward*), dan perubahan bobot (Fausett, 1994: 294-295). Fase pertama adalah fase perambatan maju, pola masukan dihitung maju dari lapisan *input* sampai lapisan *output* menggunakan suatu fungsi aktivasi untuk memperoleh *error*. Pada fase perambatan maju, neuron-neuron diaktifkan menggunakan fungsi aktivasi yang dapat didiferensialkan dan monoton naik, seperti *sigmoid biner*, *sigmoid bipolar*, dan *linear*. Fase selanjutnya adalah fase perambatan mundur, *error* yang merupakan selisih antara *output* jaringan dan target yang diinginkan terjadi di perambatan mundur. Fase perambatan mundur dimulai dari sinyal yang berhubungan langsung dengan neuron pada lapisan *output* yang kembali menuju neuron pada lapisan *input*. Fase terakhir yaitu perubahan bobot yang dimaksudkan untuk menurunkan *error* yang terjadi dengan cara memodifikasi bobot. Arsitektur jaringan *backpropagation* terlihat pada Gambar 2.9.

ini adalah untuk mengelompokkan unit-unit yang hampir sama ke dalam suatu area tertentu.

b. Pembelajaran Terawasi (*Supervised Learning*)

Metode pembelajaran pada *neural network* disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Tujuan pembelajaran terawasi adalah untuk memprediksi satu atau lebih variabel target dari satu atau lebih variabel *input* (Sarle, 1994:6). Pada proses pembelajaran, satu pola *input* akan diberikan ke suatu neuron pada lapisan *input*. Selanjutnya pola akan dirambatkan sepanjang *neural network* hingga sampai ke neuron pada lapisan *output*. Lapisan *output* akan membangkitkan pola *output* yang nantinya akan dicocokkan dengan pola *output* targetnya. Apabila terdapat perbedaan antara pola *output* maka akan muncul *error*. Apabila nilai *error* cukup besar maka mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

Terdapat banyak algoritma yang termasuk dalam pembelajaran terawasi, antara lain *Hebb Rule*, *Perceptron*, *Delta Rule*, *Backpropagation*, *Heteroassociative Memory*, *Bidirectional Associative Memory (BAM)*, dan *Learning Vector Quantization (LVQ)*. Namun yang sering digunakan dalam proses klasifikasi adalah algoritma pembelajaran *backpropagation*.



Gambar 2.9 Arsitektur Jaringan *Backpropagation*

Model NN algoritma *backpropagation* secara matematis dapat dirumuskan sebagai berikut:

$$y = f \left(b_0 + \sum_{j=1}^p b_j + \sum_{i=1}^l x_i v_{ij} \right) w_j \quad (2.20)$$

dengan,

- y = nilai *output*,
- w_j = bobot dari neuron ke- j pada lapisan tersembunyi yang menuju neuron pada lapisan *output*,
- x_i = variabel *input* ke- i , $i = 1, 2, \dots, l$,
- v_{ij} = bobot dari lapisan *input* ke- i menuju neuron ke- j pada lapisan tersembunyi,
- b_j = bobot bias pada neuron ke- j pada lapisan tersembunyi, $j = 1, 2, \dots, p$,
- b_0 = bobot bias pada neuron lapisan *output*,
- f = fungsi aktivasi.

Algoritma pembelajaran *backpropagation* adalah sebagai berikut (Sri Kusumadewi, 2004: 95-97):

a. Menetapkan parameter pembelajaran

Sebelum melakukan pembelajaran, parameter pembelajaran harus diatur terlebih dahulu. Sebagian besar aplikasi algoritma

pembelajaran yang menggunakan *gradient descent* yaitu *gradient descent (traingd)* dan *gradient descent* dengan momentum (*traingdm*). Namun kedua fungsi pembelajaran tersebut memiliki kelemahan, yaitu proses pembelajaran yang berjalan cukup lambat. Oleh sebab itu, perbaikan dengan teknik heuristik dilakukan untuk mempercepat proses pembelajaran. Perbaikan ini menghasilkan *gradient descent* dengan *adaptive learning rate (traingda)* dan *gradient descent* dengan momentum dan *adaptive learning rate (traingdx)*.

Gradient descent dengan *adaptive learning rate* memperbaiki bobot-bobot berdasarkan *gradient descent* dengan *learning rate* yang bersifat adaptif, artinya nilai *learning rate* akan diubah selama proses pelatihan untuk menjaga agar algoritma tetap stabil selama proses pelatihan. Sedangkan *gradient descent* dengan momentum dan *adaptive learning rate (traingdx)* memperbaiki bobot-bobot berdasarkan *gradient descent* dengan *learning rate* yang bersifat adaptif seperti *traingda* dan juga menggunakan momentum seperti *traingdm*. Berikut ini adalah parameter-parameter yang harus diatur untuk fungsi pembelajaran *traingdx*.

11

5) Rasio untuk menaikkan *learning rate*

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

6) Rasio untuk menurunkan *learning rate*

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

7) 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.

8) Jumlah *epoch* yang akan ditunjukkan kemajuannya

Menunjukkan jumlah *epoch* yang akan ditampilkan. Nilai *default* untuk jumlah *epoch* yang akan ditunjukkan adalah 25.

b) Kerjakan langkah-langkah berikut selama Epoch<Maksimum Epoch dan MSE>Target Error.

12

1) Maksimum *epoch*

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pembelajaran. Iterasi akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch*. Nilai *default* untuk maksimum *epoch* adalah 10.

2) Target Error

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

3) Maksimum kenaikan kinerja

Maksimum kenaikan kinerja adalah nilai maksimum kenaikan error yang diijinkan, antara error saat ini dan error sebelumnya. Nilai *default* untuk maksimum kenaikan kinerja adalah 1,04.

4) *Learning rate*

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

12

1) Fase 1 (Perambatan Maju)

a) Tiap neuron input ($x_i, i=1,2,3...l$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua neuron pada lapisan setelahnya, yaitu lapisan tersembunyi.

b) Tiap neuron pada lapisan tersembunyi ($z_j, j=1,2,...,p$) menjumlahkan sinyal-sinyal input terbobot:

$$z_j = b_j + \sum_{i=1}^l x_i v_{ij} \quad (2.21)$$

dengan,

b_j = bobot bias pada neuron ke- j pada lapisan tersembunyi dengan $j=1,2,...,p$

x_i = variabel input, $i=1,2,...,l$

v_{ij} = bobot dari lapisan input ke- i menuju neuron ke- j pada lapisan tersembunyi

Sinyal output z_j dihitung menggunakan fungsi

aktivasi sebagai berikut:

$$z_j = f(z_j) \quad (2.22)$$

kemudian sinyal tersebut dikirim ke semua neuron pada lapisan output.

c) Neuron output (y) menjumlahkan sinyal-sinyal input terbobot dari lapisan tersembunyi menggunakan rumus berikut:

$$y = b_0 + \sum_{j=1}^p z_j w_j \quad (2.23)$$

dengan,

b_0 = bias pada neuron output

13

aktivasi sebagai berikut:

$$y = f(y_{in}) \quad (2.24)$$

2) Fase 2 (Perambatan Mundur)

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

$$\delta_0 = t_k - y \quad (2.25)$$

kemudian dihitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai w_j dan koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai b_0 , rumus-rumusny adalah sebagai berikut:

$$\Delta w_j = \alpha \delta_0 z_j \quad (2.26)$$

$$\Delta b_0 = \alpha \delta_0 \quad (2.27)$$

dimana α = laju percepatan.

Langkah ini dilakukan sebanyak jumlah neuron pada lapisan tersembunyi, yaitu menghitung informasi *error* dari suatu neuron lapisan tersembunyi ke neuron pada lapisan tersembunyi sebelumnya.

□

$$b_j \text{ baru} = b_j \text{ lama} + \Delta b_j \quad (2.35)$$

Setelah proses pembelajaran, aplikasi dalam jaringan hanya terdiri dari fase *feedforward*. Pada Matlab R2010a, untuk membangun jaringan *feedforward* digunakan perintah *newff*. Secara lebih lengkap, perintah pada *toolbox* Matlab R2010a adalah sebagai berikut:

$$\text{net} = \text{newff}(\text{PR}, S_1, S_2, \dots, S_q, \text{TF}_1, \text{TF}_2, \dots, \text{TF}_q, \text{BTF})$$

dengan,

- PR : matriks berukuran $R \times 2$ yang berisi nilai minimum dan maksimum, dengan R adalah banyak variabel *input*;
- S_i : banyak neuron pada lapisan ke- i .
- TF_i : fungsi aktivasi pada lapisan ke- i (default: *tansig*).
- BTF : fungsi pembelajaran untuk jaringan (default: *trainlm*).

D. Himpunan Fuzzy

1. Konsep Dasar Himpunan Fuzzy

a. Himpunan Klasik (*Crisp*)

Himpunan klasik adalah himpunan yang membedakan anggota dan bukan anggota dengan batasan yang jelas (Ross, 2010:26). Misalkan pada suatu himpunan A hanya akan terdapat dua kemungkinan keanggotaan yaitu menjadi anggota A atau tidak menjadi anggota A . Suatu nilai yang menunjukkan seberapa besar tingkat keanggotaan suatu elemen x pada suatu himpunan A disebut nilai keanggotaan atau derajat keanggotaan, dinotasikan dengan $\mu_A(x)$. Pada himpunan klasik, hanya ada dua nilai keanggotaan, yaitu $\mu_A(x) = 1$ untuk x merupakan anggota A dan $\mu_A(x) = 0$ untuk x bukan anggota A .

□

b) Tiap-tiap neuron tersembunyi (z_j) menjumlahkan delta *input*-nya dari neuron-neuron yang berada pada lapisan di atasnya sebagai berikut:

$$\delta_{in_j} = \sum_{k=1}^n \delta_0 w_j \quad (2.28)$$

Kalikan nilai δ_{in_j} dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_j = \delta_{in_j} f' z_{in_j} \quad (2.29)$$

Kemudian hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai v_{ij} dengan rumus sebagai berikut:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.30)$$

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

$$\Delta b_j = \alpha \delta_j \quad (2.31)$$

3) Fase Perubahan Bobot dan Bias

a) Neuron *output* (y) memperbaiki bias dan bobotnya sebagai berikut:

$$w_j \text{ baru} = w_j \text{ lama} + \Delta w_j \quad (2.32)$$

$$b_0 \text{ baru} = b_0 \text{ lama} + \Delta b_0 \quad (2.33)$$

b) Tiap-tiap neuron lapisan tersembunyi (z_j) memperbaiki bias dan bobotnya sebagai berikut:

$$v_{ij} \text{ baru} = v_{ij} \text{ lama} + \Delta v_{ij} \quad (2.34)$$

□

Contoh 2.1 Jika $S = \{1,3,5,7,9\}$ adalah suatu himpunan semesta,

$A = \{1,3,5\}$, dan $B = \{5,7,9\}$, maka dapat dikatakan bahwa:

- 1) Nilai keanggotaan 1 pada himpunan $A, \mu_A 1 = 1$, karena $1 \in A$,
- 2) Nilai keanggotaan 9 pada himpunan $A, \mu_A 9 = 0$, karena $9 \notin A$,
- 3) Nilai Keanggotaan 5 pada himpunan $B, \mu_B 5 = 1$, karena $5 \in B$.
- 4) Nilai Keanggotaan 3 pada himpunan $B, \mu_B 3 = 0$, karena $3 \notin B$.

b) Himpunan Fuzzy

Tahun 1965 Lotfi A. Zadeh memperkenalkan teori himpunan *fuzzy* sebagai cara untuk merepresentasikan kesamaran dalam bahasa (Lin & Lee, 1996:3). Pada dasarnya, himpunan *fuzzy* merupakan perluasan dari himpunan klasik. Jika pada himpunan klasik nilai keanggotaan hanya ada dua kemungkinan yaitu 0 atau 1, pada himpunan *fuzzy* nilai keanggotaan berkisar pada rentang 0 sampai 1.

Menurut Zimmermann (1991:11-12) jika X adalah koleksi dari obyek-obyek yang dinotasikan secara generik oleh x , maka suatu himpunan *fuzzy* A , dalam X adalah suatu himpunan pasangan berurutan:

$$A = \{(x, \mu_A x) \mid x \in X\} \quad (2.36)$$

dengan $\mu_A x$ adalah derajat keanggotaan x di A yang memetakan X ke ruang keanggotaan M yang terletak pada rentang $[0, 1]$.

Ada beberapa cara untuk menotasikan himpunan *fuzzy*, antara lain:

□

Contoh 2.2 Misalkan sebuah industri kendaraan bermotor ingin merancang dan memproduksi sebuah mobil yang nyaman untuk digunakan keluarga yang besar. Ada 5 model yang telah dirancang dan ditunjukkan dalam variabel $X = \{1, 2, 3, 4, 5\}$, dengan 1 adalah desain mobil ke-1, dan seterusnya. Himpunan fuzzy \tilde{A} yang merupakan himpunan “mobil yang nyaman digunakan untuk keluarga yang besar” dapat ditulis sebagai:

$$\tilde{A} = \{(1; 0,6); (2; 0,3); (3; 0,8); (4; 0,2); (5; 0,1)\}$$

Yang artinya:

- Mobil pertama memenuhi tingkat kenyamanan sebesar 0,6 dari skala kenyamanan 0 sampai 1.
- Mobil kedua memenuhi kenyamanan sebesar 0,3 dari skala kenyamanan 0 sampai 1.
- Mobil ketiga memenuhi kenyamanan sebesar 0,8 dari skala kenyamanan 0 sampai 1.
- Mobil keempat memenuhi kenyamanan sebesar 0,2 dari tingkat kenyamanan 0 sampai 1.
- Mobil kelima memenuhi kenyamanan sebesar 0,1 dari tingkat kenyamanan 0 sampai 1.

□

- Linguistik, yaitu penamaan suatu himpunan yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami seperti MUDA, PAROBAYA, TUA.
- Numeris, yaitu suatu nilai (bilangan) yang menunjukkan ukuran dari suatu variabel seperti 40, 25, 50, dsb.

2. Fungsi Keanggotaan Fuzzy (Fuzzy Membership Function)

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval $[0,1]$. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan pendekatan fungsi. Terdapat beberapa fungsi keanggotaan yang dapat digunakan untuk merepresentasikan himpunan fuzzy, diantaranya adalah representasi linear, representasi kurva segitiga, representasi bentuk bahu, representasi kurva bentuk lonceng (*Bell Curved*), representasi kurva trapesium, dan representasi kurva-S.

Fungsi keanggotaan yang paling sering digunakan adalah fungsi keanggotaan representasi kurva segitiga (Barua, et al, 2014:1), karena fungsi keanggotaan segitiga dapat dikembangkan menjadi fungsi keanggotaan yang lainnya melalui suatu distribusi pendekatan perubahan fungsi keanggotaan. Pada representasi kurva segitiga (Gambar 2.10) hanya terdapat satu nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika $x = b$.

□

- Dalam beberapa literatur, himpunan fuzzy sering dinotasikan sebagai:

$$\tilde{A} = \mu_{\tilde{A}} u_1 / u_1 + \mu_{\tilde{A}} u_2 / u_2 + \dots + \mu_{\tilde{A}} u_n / u_n = \sum_{i=1}^n \mu_{\tilde{A}} u_i / u_i \quad (2.37)$$

untuk himpunan semesta diskret, dan

$$\tilde{A} = \mu_{\tilde{A}} u_1 / u_1 + \mu_{\tilde{A}} u_2 / u_2 + \dots + \mu_{\tilde{A}} u_n / u_n = \int \mu_{\tilde{A}} u_i / u_i \quad (2.38)$$

untuk himpunan semesta kontinu.

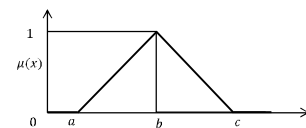
Tanda “ \sum ” bukan menotasikan operasi penjumlahan seperti yang dikenal pada aritmetika, tetapi melambangkan keseluruhan unsur $x \in X$ bersama dengan fungsi keanggotaan $\mu_{\tilde{A}} x$ dalam himpunan fuzzy \tilde{A} . Tanda “ \int ” bukan menotasikan operasi penjumlahan, tetapi melambangkan pemisahan antara keanggotaan elemen himpunan fuzzy \tilde{A} dan fungsi keanggotaan yang lain. Begitu pula tanda “ \int ” bukan merupakan operasi pembagian yang dikenal dalam kalkulus, tetapi melambangkan hubungan antara satu elemen himpunan fuzzy \tilde{A} dan fungsi keanggotaannya.

Menggunakan Contoh 2.2, himpunan fuzzy \tilde{A} dapat direpresentasikan sebagai:

$$\tilde{A} = 0,6/1 + 0,3/2 + 0,8/3 + 0,2/4 + 0,1/5$$

Himpunan fuzzy memiliki dua atribut, yaitu (Sri Kusumadewi: 2003:158):

□



Gambar 2.10 Kurva Segitiga

Fungsi keanggotaannya dirumuskan sebagai berikut:

$$\mu_x = \begin{cases} 0, & x < a \text{ atau } x > c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases} \quad (2.39)$$

3. Operasi Himpunan Fuzzy

Terdapat beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Dalam himpunan fuzzy dikenal beberapa operasi dasar, yaitu (Ibrahim, 2004: 37):

- Komplemen

Komplemen dari suatu himpunan fuzzy A adalah \bar{A} dan fungsi keanggotaannya didefinisikan sebagai:

$$\mu_{\bar{A}} x = 1 - \mu_A x \quad (2.40)$$

- Gabungan

Gabungan dari dua himpunan fuzzy A dan B adalah himpunan fuzzy dengan fungsi keanggotaan yang didefinisikan sebagai:

$$\mu_{A \cup B}(x) = \max \mu_A x, \mu_B(x) \quad (2.41)$$

- Irisan

Irisan dari dua himpunan fuzzy A dan B adalah himpunan fuzzy dengan fungsi keanggotaan yang didefinisikan sebagai:

$$\mu_{A \cap B}(x) = \min \mu_A x, \mu_B(x) \quad (2.42)$$

□

Logika *fuzzy* merupakan representasi ketidakpastian, ketidaktepatan, ketidaktepatan yang dikonstruksikan dengan *IF-THEN rules*. Logika *fuzzy* merupakan perkembangan dari logika klasik. Pada logika *fuzzy* nilai kebenaran suatu *item* berada pada interval [0,1].

Logika *fuzzy* didasarkan pada bahasa alami, artinya dasar logika *fuzzy* adalah bahasa komunikasi manusia (MathWorks, 1999:5). Bahasa alami adalah bahasa yang digunakan dalam komunikasi sehari-hari. Logika *fuzzy* membangun struktur deskripsi kualitatif penggunaan bahasa sehari-hari tersebut sehingga logika *fuzzy* mudah digunakan.

5. Sistem Inferensi Fuzzy (Fuzzy Inference System)

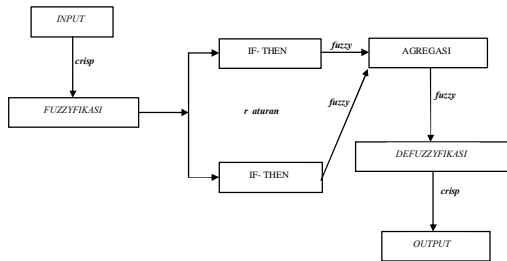
Salah satu aplikasi logika *fuzzy* yang telah berkembang amat luas dewasa ini adalah sistem inferensi *fuzzy*, yaitu suatu kerangka komputasi yang bekerja atas dasar teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *IF-THEN*, dan prinsip penalaran *fuzzy* (Sri Kusumadewi & Sri Hartati, 2010:40). Sistem ini berfungsi untuk mengambil keputusan melalui proses tertentu dengan menggunakan aturan inferensi berdasarkan logika *fuzzy*. Pada dasarnya sistem inferensi fuzzy terdiri dari tiga unit, yaitu :

a. Unit fuzzifikasi (*fuzzification unit*)



c. Unit defuzzifikasi (*defuzzification unit* / unit penegasan)

Pada sistem inferensi *fuzzy*, *input* klasik dikonversikan oleh unit fuzzifikasi ke nilai *fuzzy* yang sesuai. *Input* yang telah difuzzikan selanjutnya dikirim ke unit basis pengetahuan yang berisi *r* aturan *fuzzy*. Setiap aturan akan menghasilkan *fire strength* (*output fuzzy*). Apabila jumlah aturan yang tersedia lebih dari satu, maka akan dilakukan agregasi atau penggabungan dari semua aturan. Selanjutnya pada hasil agregasi akan dilakukan langkah terakhir oleh unit defuzzifikasi untuk mendapatkan nilai klasik sebagai *output* sistem inferensi *fuzzy*. Langkah-langkah tersebut secara skematis disajikan dalam Gambar 2.11.



Gambar 2.11 Diagram Blok Sistem Inferensi Fuzzy (Sri Kusumadewi & Sri Hartati, 2010:2)

Terdapat tiga metode inferensi *fuzzy*, yaitu metode Tsukamoto, metode Mamdani (metode Max-Min), dan metode Sugeno. Pada metode



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

b. Unit basis pengetahuan (*knowledge base unit*), yang terdiri dari dua bagian:

- 1) Basis data (*data base*), yang memuat fungsi-fungsi keanggotaan dari himpunan-himpunan *fuzzy* yang terkait dengan nilai dari variabel-variabel linguistik yang dipakai.
- 2) Basis aturan (*rule base*), yang memuat *r* aturan *fuzzy*. Aturan *fuzzy* merupakan otak dari sistem *fuzzy*. *Input* yang telah diubah menjadi variabel *fuzzy* dikirim ke basis aturan untuk didefinisikan sebelumnya sehingga dihasilkan *output fuzzy*. Aturan *fuzzy* dinyatakan dalam bentuk *IF-THEN rules*. *IF-THEN rules* ditulis sebagai (Wang, 1997:91):

$$R^s: IF x_1 \text{ is } A_1^s \circ \dots \circ x_l \text{ is } A_l^s THEN z \text{ is } B^s \quad (2.43)$$

dengan,

- R^s = aturan inferensi ke-*s*, ($s = 1, 2, \dots, r$, dan *r* adalah banyak aturan inferensi),
- x_i = variabel *input* ke-*i*, $i = 1, 2, \dots, l$,
- A_i^s = himpunan *fuzzy input* pada variabel x_i aturan inferensi ke-*s*,
- z = nilai *output*,
- B^s = himpunan *fuzzy output* pada aturan inferensi ke-*s*,
- \circ = operator *fuzzy*.



Tsukamoto, setiap konsekuensi pada aturan yang berbentuk *IF-THEN* harus direpresentasikan menggunakan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Pada metode Mamdani, untuk mendapatkan output diperlukan 4 tahapan, yaitu pembentukan himpunan *fuzzy* aplikasi fungsi implikasi (aturan), komposisi aturan, dan defuzzifikasi. Metode inferensi *fuzzy* Sugeno atau dikenal Takagi Sugeno Kang (TSK) merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dengan aturan *IF-THEN* dimana *outputnya* tidak berupa himpunan *fuzzy* melainkan berupa konstanta atau persamaan linear (Sri Kusumadewi, 2003: 180-196). Metode inferensi *fuzzy* Sugeno inilah yang akan digunakan dalam tugas akhir ini. Terdapat 2 model untuk metode inferensi *fuzzy* Sugeno, yaitu metode Sugeno orde-0 dan metode Sugeno orde-1 (Lin & Lee, 1996: 18-19).

a. Metode Sugeno orde-0

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

$$IF x_1 \text{ is } A_1 \circ x_2 \text{ is } A_2 \circ \dots \circ (x_l \text{ is } A_l) THEN y = k \quad (2.44)$$

dengan,

- x_i = variabel *input* ke-*i*, $i = 1, 2, \dots, l$,
- A_i = himpunan *fuzzy input* pada variabel x_i ,
- y = nilai *output*,
- k = konstanta (berupa *crisp*),
- \circ = operator *fuzzy*.



$$IF \ x_1 \text{ IS } A_1 \circ \dots \circ (x_i \text{ IS } A_i) \text{ THEN } y = p_1 x_1 + \dots + p_l x_l + q \quad (2.43)$$

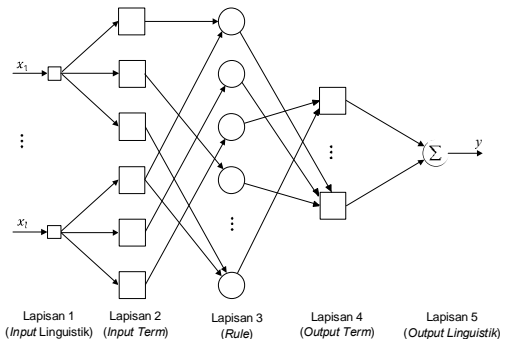
dengan,
 x_i = variabel *input* ke- i , $i = 1, 2, \dots, l$,
 A_i = himpunan *fuzzy* *input* pada variabel x_i ,
 y = nilai *output*,
 p_i = konstanta *crisp* pada variabel x_i ,
 q = konstanta untuk persamaan linier,
 \circ = operator *fuzzy*.

E. Model Neuro Fuzzy (NF)

Model NF merupakan penggabungan antara NN dan logika *fuzzy* ke dalam satu sistem kecerdasan buatan yang terintegrasi. Model NF menggabungkan kelebihan dari model NN dengan melakukan pembelajaran dan logika *fuzzy* dengan kemampuan penalarannya. Model NF dirancang untuk merealisasikan proses penalaran *fuzzy*, dimana bobot-bobot yang terhubung pada NN terhubung dengan parameter-parameter penalaran *fuzzy*.

NN pada model NF digunakan untuk membentuk fungsi keanggotaan pada bagian anteseden dan sebagai pengendali penalaran *fuzzy* (melakukan inferensi pada bagian konsekuen). Pada kebanyakan aplikasi logika *fuzzy*, fungsi keanggotaan biasanya dibentuk menggunakan pendekatan terhadap beberapa fungsi tertentu seperti linear, segitiga, trapesium, bentuk bahu, dan sebagainya. Pada dasarnya, fungsi keanggotaan dapat direalisasikan dengan menggunakan suatu neuron melalui fungsi aktivasinya untuk mendapatkan

aturan *fuzzy*. Neuron pada lapisan fuzzifikasi menerima *input* berupa *crisp* dan menentukan nilai keanggotaan setiap *input*.



Gambar 2.12 Struktur Neuro Fuzzy Secara Umum

3. Lapisan 3 adalah lapisan dengan aturan *fuzzy*. Masing-masing neuron dalam lapisan ini merepresentasikan aturan logika *fuzzy*, artinya setiap neuron pada lapisan ini membentuk aturan dasar *fuzzy*. *Input* yang diterima lapisan ini merupakan himpunan *fuzzy* hasil *fuzzifikasi* (anteseden) pada lapisan 2. *Input* tersebut dimasukkan ke dalam neuron-neuron yang bersesuaian yang selanjutnya dalam tugas akhir ini disebut *cluster*. Teknik pengklasteran yang digunakan adalah *Fuzzy C-Means* (FCM). FCM adalah suatu teknik pengklasteran data berdasarkan derajat keanggotaan data tersebut. Konsep dasar FCM adalah menentukan pusat *cluster* yang

fungsi keanggotaan yang diinginkan. Sedangkan konsep dasar penggunaan NN sebagai pengendali penalaran *fuzzy* adalah menggunakan NN untuk merealisasikan atau membangkitkan sistem inferensi *fuzzy* (Lin & Lee, 1996:498). Hal tersebut dilakukan karena salah satu kelemahan sistem inferensi *fuzzy* adalah dalam penentuan fungsi keanggotaan dan pembangkitan fungsi pembelajaran pada aturan-aturan inferensi.

Lin & Lee (1996:507) mendefinisikan aturan yang dipakai dalam model NF dengan metode inferensi *fuzzy* Sugeno sebagai berikut:

$$R^s: IF \ x = \ x_1, x_2, \dots, x_l \text{ IS } A_s \text{ THEN } z^s = NN_s \ x_1, x_2, \dots, x_l \quad (2.46)$$

dengan,

- R^s = aturan ke- s ($s = 1, 2, \dots, r$, dan r adalah banyak aturan inferensi),
- x_i = variabel *input* ke- i , $i = 1, 2, \dots, l$,
- A_s = himpunan *fuzzy* pada aturan ke- s bagian anteseden,
- z^s = nilai *output* aturan ke- s ,
- NN_s = NN dengan *input* x_1, x_2, \dots, x_l dan *output* z^s .

Struktur NF secara umum ditunjukkan pada Gambar 2.12 (Lin & Lee, 1996: 536). Setiap lapisan pada NF dikaitkan dengan langkah-langkah tertentu dalam proses inferensinya. Dari Gambar 2.12 (Lin & Lee, 1996: 536-537),

- Lapisan 1 adalah lapisan *input* yang merepresentasikan variabel linguistik. Setiap *neuron* dalam lapisan ini mengirimkan sinyal langsung ke lapisan berikutnya .
- Lapisan 2 adalah lapisan *fuzzifikasi*. *Neuron* pada lapisan ini merupakan representasi dari himpunan *fuzzy* yang digunakan sebagai anteseden dalam

menandai lokasi rata-rata untuk tiap-tiap *cluster*. *Output* dari FCM berupa deretan beberapa pusat *cluster* dan derajat keanggotaan tiap data. Informasi ini dapat digunakan untuk membangun suatu sistem inferensi *fuzzy*. Algoritma pengklasteran menggunakan FCM adalah sebagai berikut (Zimmerman, 1991:234):

- Menentukan matriks X yang berukuran $n \times l$ dengan n adalah banyak data yang akan diklaster dan l adalah banyak variabel yang akan diinput,
- Menentukan banyak *cluster* yang akan dibentuk. Banyak *cluster* ditentukan secara bebas asalkan *cluster* yang dibentuk lebih dari 1 $C \geq 2$.
- Menentukan pangkat (pembobot) yang dinotasikan dengan w . Banyaknya pembobot harus lebih dari 1 $w > 1$.
- Menentukan maksimum iterasi.
- Menentukan kriteria penghentian yang dinotasikan dengan ϵ yaitu nilai positif yang sangat kecil.
- Menentukan iterasi awal, $t = 1$ dan $\Delta = 1$
- Membentuk matriks partisi awal yang dinotasikan dengan U yaitu matriks dengan entri-entri yaitu μ_{ct} x_i = nilai keanggotaan setiap data ke- t pada *cluster* ke- c dengan ketentuan sebagai berikut:

$$U = \begin{matrix} \mu_{11} & x_1 & \mu_{12} & x_2 & \dots & \mu_{1n} & x_n \\ \mu_{21} & x_1 & \mu_{22} & x_2 & \dots & \mu_{2n} & x_n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{c1} & x_1 & \mu_{c2} & x_2 & \dots & \mu_{cn} & x_n \end{matrix} \quad (2.47)$$

$$V = \begin{pmatrix} v_{c1} \\ \vdots \\ v_{ci} \\ \vdots \\ v_{ct} \end{pmatrix} \quad (2.48)$$

$$v_{ci} = \frac{\sum_{k=1}^n \mu_{ck} w_{ki} x_{ki}}{\sum_{k=1}^n \mu_{ck} w_{ki}} \quad (2.49)$$

dengan:

- v_{ci} = pusat cluster ke-c pada variabel ke-i
- μ_{ct} = nilai keanggotaan data ke-t pada cluster ke-c
- w = pangkat (pembobot)
- x_{ti} = data ke-t pada variabel ke-i

- i. Memperbaiki derajat keanggotaan setiap data pada setiap cluster yaitu dengan cara memperbaiki matriks partisi. Cara memperbaiki derajat keanggotaan adalah sebagai berikut:

$$\mu_{ct} = \frac{d_{ct}}{d_{ct}^2 + (w-1)} \quad (2.50)$$

$$d_{ct} = d \cdot x_t - v_c = \sum_{i=1}^n (x_{ti} - v_{ci})^{1/2} \quad (2.51)$$

dengan:

- μ_{ct} = Nilai keanggotaan data ke-t pada cluster ke-c.
- d_{ct} = Jarak antara pusat cluster ke-c dengan data ke-t.
- d_{it} = Jarak antara pusat cluster variabel ke-i dengan data ke-t.

- j. Langkah terakhir adalah menentukan kriteria berhenti, yaitu perubahan matriks partisi pada iterasi sekarang dengan iterasi sebelumnya, sebagai berikut

$$\Delta = U^t - U^{t-1} \quad (2.52)$$

1. FMN Tipe I (konsekuen berupa variabel fuzzy)

Pada FMN tipe I, aturan fuzzy memiliki konsekuen berupa variabel fuzzy. Format aturan untuk FMN tipe I adalah:

$$R_k^s: \text{IF } x_{1t}^s, x_{2t}^s, \dots, x_{it}^s \text{ is } A_s \text{ THEN } g_t^s \text{ is } B_k \text{ is } \tau_{R_k^s} \quad (2.53)$$

$s = 1, 2, \dots, r$;
dengan

- R_k^s = aturan inferensi fuzzy ke-s konsekuen ke-k
- x_{it}^s = variabel input ke-i data ke-t pada aturan inferensi ke-s;
- A_s = himpunan fuzzy bagian antesenden pada aturan inferensi ke-s,
- g_t^s = target jaringan data ke-t pada aturan inferensi ke-s bagian konsekuen,
- B_k = himpunan fuzzy pada konsekuen yang memiliki fungsi keanggotaan linear,
- $\tau_{R_k^s}$ = nilai kebenaran linguistik dari aturan R_k^s dalam bentuk konstanta yang terletak pada interval $[0, 1]$.

2. FMN Tipe II (konsekuen berupa suatu nilai tunggal)

Pada FMN tipe II, aturan fuzzy memiliki konsekuen berupa suatu nilai tunggal (konstanta). Format aturan untuk FMN tipe II adalah:

$$R^s: \text{IF } x_{1t}^s, x_{2t}^s, \dots, x_{it}^s \text{ is } A_s \text{ THEN } g_t^s \text{ is } f_s; s = 1, 2, \dots, r \quad (2.54)$$

dengan

- R^s = aturan inferensi fuzzy ke-s
- x_{it}^s = variabel input ke-i data ke-t pada aturan inferensi ke-s;
- A_s = himpunan fuzzy bagian antesenden pada aturan inferensi ke-s,
- g_t^s = target jaringan data ke-t pada aturan inferensi ke-s bagian konsekuen,
- f_s = konstanta.

3. FMN Tipe III (konsekuen berupa persamaan linear orde pertama)

Pada FMN tipe III, aturan fuzzy memiliki konsekuen berupa persamaan linear orde pertama. Format aturan untuk FMN tipe III adalah:

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

4. Lapisan 4 adalah lapisan nilai keanggotaan output. Neuron pada lapisan ini adalah himpunan fuzzy yang digunakan sebagai konsekuen pada aturan fuzzy.
5. Lapisan 5 adalah lapisan defuzzifikasi. Pada lapisan ini diperoleh output dari sistem neuro fuzzy.

Pemodelan fuzzy didasarkan pada sistem yang hendak dibangun menggunakan sistem inferensi fuzzy (Lin, 1996:511). Namun, sangat tidak mudah untuk mengidentifikasi aturan-aturan fuzzy dan membangkitkan fungsi keanggotaan, terutama apabila sistem yang dihadapi adalah sistem non-linear yang sangat kompleks.

Pada tahun 1992, Horikawa memperkenalkan metode pemodelan fuzzy melalui pembelajaran NN algoritma backpropagation. Terdapat 3 tipe Fuzzy Modelling Network (FMN) yang diperkenalkan Horikawa yaitu FMN tipe I, FMN tipe II, FMN tipe III. Pada tugas akhir ini digunakan FMN tipe III. FMN ini akan mengidentifikasi aturan-aturan fuzzy dan fungsi keanggotaan secara otomatis dengan cara memodifikasi bobot-bobot NN melalui algoritma pembelajaran backpropagation (Sri Kusumadewi & Sri Hartati, 2010:334).

$$R^s: \text{IF } x_{1t}^s, x_{2t}^s, \dots, x_{it}^s \text{ is } A_s \text{ THEN } g_t^s = f_s \cdot x_{1t}^s \cdot x_{2t}^s \cdot \dots \cdot x_{it}^s \quad (2.55)$$

dengan

- R^s = aturan inferensi fuzzy ke-s,
- x_{it}^s = variabel input ke-i data ke-t pada aturan inferensi ke-s;
- g_t^s = target jaringan data ke-t pada aturan inferensi ke-s bagian konsekuen,
- A_s = himpunan fuzzy bagian antesenden pada aturan inferensi ke-s,
- $f_s \cdot x_{1t}^s \cdot x_{2t}^s \cdot \dots \cdot x_{it}^s = p_{s0} + p_{s1}x_1 + p_{s2}x_2 + \dots + p_{si}x_i$; p_{si} adalah parameter konsekuen; $i = 1, 2, \dots$.

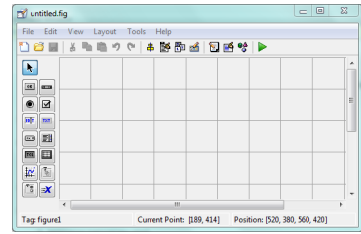
F. Graphical User Interface (GUI)

MATLAB (Matrix Laboratory) merupakan salah satu perangkat lunak yang dirancang khusus sebagai solusi untuk mengerjakan permasalahan yang berkaitan dengan matematika dan lebih khusus berupa matriks. Sebagai perangkat lunak yang dibangun secara kompleks, MATLAB dilengkapi dengan Graphical User Interface Design (GUIDE) atau alat yang berfungsi untuk merancang suatu tampilan berbasis visual (Arif Hermawan, 2006: 106-109).

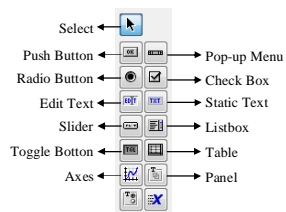
Graphical User Interface Design (GUIDE) merupakan sekumpulan alat MATLAB yang dirancang untuk membangun GUI (Graphical User Interface) dengan mudah dan cepat. GUI merupakan tampilan grafis hasil dari GUIDE yang bertujuan untuk memudahkan pengguna untuk berinteraksi. Program yang menggunakan GUI umumnya lebih mudah dipelajari dan digunakan karena orang yang tidak mengetahui pemrograman MATLAB sekalipun dapat menjalankan program tersebut. Berdasarkan kelebihan inilah, pada tugas akhir ini GUI digunakan untuk menampilkan hasil yang diperoleh dari proses klasifikasi stadium kanker kolorektal.

penyimpanan ini akan GUI terdiri dari dua ekstensi yaitu *figure* dan *menu*.

Berikut adalah tampilan *base design* GUI pada Matlab R2010a.



Gambar 2.13 Base Design GUI



Gambar 2.14 Toolbox GUI

6. Panel

Panel merupakan kotak tertutup yang dapat digunakan untuk mengelompokkan kontrol-kontrol yang berhubungan.

7. Table

Tabel berguna untuk menampilkan nilai, penggunaan tabel hampir sama seperti kolom-kolom pada Ms. Excel.

8. Check Box

Kontrol *check box* digunakan untuk menyediakan beberapa pilihan mandiri atau tidak bergantung dengan pilihan-pilihan lainnya. Contoh aplikasi penggunaan *check box* adalah ketika kita diminta untuk memilih hobi. Karena hobi bisa lebih dari satu, maka kita dapat mengklik *check box* lebih dari satu kali.

9. Slider

Slider berguna untuk menginput nilai tanpa menggunakan *keyboard*, tetapi hanya dengan cara menggeser *slider* secara vertikal maupun horizontal ke nilai yang kita inginkan.

10. Pop-up Menu

Pop-up menu berfungsi menampilkan daftar pilihan atau menu yang didefinisikan pada *String Property*. Ketika tidak dibuka, pop-up menu hanya menampilkan satu item yang menjadi pilihan pertama pada *String Property*.

Alat (*tool*) yang digunakan dalam merancang GUI terdiri atas 1 selektor (*select*) dan 11 jenis *tool*, yaitu *push button*, *toggle button*, *radio button*, *check box*, *edit text*, *static text*, *slider*, *panel*, *listbox*, *pop-up menu*, dan *axes*.

1. Push Button

Push button merupakan jenis kontrol berupa tombol tekan yang akan menghasilkan tindakan jika diklik, misalnya tombol OK, Keluar, Save, Proses, dan sebagainya.

2. Toggle Button

Toggle button memiliki fungsi yang sama dengan *push button*. Perbedaannya adalah saat *push button* ditekan, maka tombol akan kembali pada posisi semula jika tombol mouse dilepas, sedangkan pada *toggle button*, tombol tidak akan kembali ke posisi semula, kecuali *toggle button* ditekan kembali.

3. Radio Button

Radio button digunakan untuk memilih atau menandai satu pilihan dari beberapa pilihan yang ada.

4. Edit Text

Edit text digunakan untuk memasukkan atau memodifikasi suatu text yang diinputkan dari *keyboard*.

5. Static Text

Static text berguna untuk menampilkan teks/tulisan, tetapi tidak bisa dimodifikasi atau diedit.

11. Axes

Axes berguna untuk menampilkan sebuah grafik atau gambar (*image*). *Axes* sebenarnya tidak masuk dalam *UIControl*, tetapi *axes* dapat diprogram agar pemakai dapat berinteraksi dengan *axes* dan obyek grafik yang ditampilkan melalui *axes*.

G. Mean Square Error (MSE)

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

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2 \quad (2.56)$$

dengan,

e_t = *error* ke-*t*, (selisih target ke-*t* dan *output* ke-*t*, $t = 1, 2, \dots, n$)
n = banyak data pengamatan.

H. Ketepatan Hasil Klasifikasi

Hasil pembelajaran bisa saja memberikan hasil klasifikasi yang tidak tepat. Untuk menghitung ketepatan hasil klasifikasi dapat digunakan ukuran statistik kinerja dari klasifikasi, yaitu sensitivitas, spesifisitas, dan akurasi. Ketiganya digunakan untuk mengetahui seberapa bagus dan terpercaya hasil klasifikasi yang telah dilakukan. Kemungkinan yang dapat terjadi dalam hasil klasifikasi ditunjukkan dalam Tabel 2.1 berikut.

Berdasarkan Tabel 2.1, terdapat empat kemungkinan hasil klasifikasi

yaitu :

TP : pasien memiliki penyakit dan hasil klasifikasi menyatakan pasien memiliki penyakit;

FN : pasien memiliki penyakit dan hasil klasifikasi menyatakan pasien tidak memiliki penyakit;

FP : pasien tidak memiliki penyakit dan hasil klasifikasi menyatakan pasien memiliki penyakit;

TN : pasien tidak memiliki penyakit dan hasil klasifikasi menyatakan pasien tidak memiliki penyakit.

1. Sensitivitas

Sensitivitas adalah kemampuan tes untuk mengidentifikasi hasil positif yang tepat (Wong & Lim, 2011: 316). Sensitivitas digunakan untuk mengidentifikasi pasien yang tepat memiliki penyakit dari semua pasien yang memiliki penyakit. Misalnya, jika pada hasil klasifikasi kanker kolorektal diperoleh nilai sensitivitas 95% artinya untuk pasien yang klasifikasi akurat sebesar 95% dalam menyatakan pasien memiliki penyakit kanker kolorektal. Rumus sensitivitas adalah sebagai berikut (Sharma & Mukharje, 2014: 22):

□

$$\text{Sensitivitas} = \frac{TP}{TP+FN} \times 100\% \quad (2.57)$$

2. Spesifisitas

Spesifisitas adalah kemampuan tes untuk mengidentifikasi hasil negatif yang tepat (Wong & Lim, 2011: 316). Spesifisitas digunakan untuk mengidentifikasi pasien yang tidak memiliki penyakit dari semua pasien yang tidak memiliki penyakit. Misalnya, jika pada hasil klasifikasi kanker kolorektal diperoleh nilai spesifisitas 95% artinya untuk pasien yang tidak memiliki penyakit kanker kolorektal, klasifikasi akurat sebesar 95% dalam menyatakan pasien tidak memiliki penyakit kanker kolorektal. Rumus spesifisitas adalah sebagai berikut (Sharma & Mukharje, 2014: 22):

$$\text{Spesifisitas} = \frac{TN}{TN+FP} \times 100\% \quad (2.58)$$

3. Akurasi

Akurasi adalah kemampuan tes untuk mengidentifikasi hasil positif maupun hasil negatif secara tepat. Misalnya, pada hasil klasifikasi kanker kolorektal diperoleh nilai akurasi 95% artinya klasifikasi akurat sebesar 95%, baik untuk pasien yang dinyatakan tidak memiliki penyakit kanker kolorektal maupun dinyatakan memiliki penyakit kanker kolorektal stadium 1, stadium 2, stadium 3, dan stadium 4. Rumus akurasi adalah sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah klasifikasi tepat (cocok)}}{\text{Jumlah data klasifikasi keseluruhan}} \times 100\% \quad (2.59)$$

□

DAFTAR PUSTAKA

- Al-Daoud, E. (2010). Cancer Diagnosis Using Modified Fuzzy Network. *Universal Journal of Computer Science and Engineering Technology*, Vol.1, No.2.
- Anami, B.S. & Burkpalli, V.C. (2009). Texture Based Identification and Classification of Bulk Sugary Food Object. *ICGSTGVIP Journal*, Vol. 9, Issue 4, Hlm. 9-14.
- Andri Kristanto. (2004). *Jaringan Syaraf Tiruan: Konsep Dasar, Algoritma, dan Aplikasi*. Yogyakarta: Gava Media.
- Arif Hermawan. (2006). *Jaringan Syaraf Tiruan (Teori dan Aplikasi)*. Yogyakarta: Andi.
- Asril Zahari. (2010). Deteksi Dini, Diagnosa, dan Penatalaksanaan Kanker Kolon dan Rektum. Padang: *Repository Universitas Andalas*.
- Barua, A., Muduruni, L.S., & Kosheleva, O. (2014). Why Trapezoidal and Triangular Membership Function Work So Well: Towards Theoretical Explanation. *Journal of Uncertain Systems*, Vol. 8.
- Beale, M.H., et al. (2010). *Neural Network Toolbox TM 7 User's Guide*. Natick: The Math Works, Inc.
- Fausett, L. (1994). *Fundamentals of Neural Network: Architectures, Algorithms, and Applications*. Upper Saddle River, New Jersey: Prentice-Hall.
- Gonzales, R.C. & Woods, R.E. (2002). *Digital Image Processing: Second Edition*. Upper Saddle River, New Jersey: Prentice Hall.
- Guntau, J. *Endoskopie Atlas*. Hamburg, Jerman: Albertinen-Krankenhaus, Diakses dari http://endoskopiebilder.de/kolon_normalbefunde.html pada tanggal 21 Maret 2013 pukul 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, K. & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transaction on System, Man and Cybernetics*, Vol. 3, Hlm. 610-621.
- Haykin, S. (1999). *Neural Network: A Comprehensive Foundation*. New York: Prentice Hall.
- Hota, H. S., Shrivasa, A. K., & Singhai, S. K. (2013). Artificial Neural Network, Decision Tree and Statistical Techniques Applied for Designing and

11□

Developing E-mail Classifier. *International Journal of Recent Technology and Engineering (IJRTE)*. Vol. 1. Issue 6.

Ibrahim, A.M. (2004). *Fuzzy Logic for Embedded Systems Applications*. Burlington: Elsevier Science.

Iskandar Zulkarnaini. (2013). Diagnosa Penyakit Kanker Payudara dengan Menggunakan Sistem Fuzzy MADM (Multi Attribute Decision Making). Aceh: *Repository Universitas Al Muslim*.

Kalas, M.S. (2010). An Artificial Neural Network for Detection of Biological Early Brain Cancer. *International Journal of Computer Applications*, Vol.1, No. 6.

Khalil, R.A. & Al-Kazzaz, S.A. (2009). Digital Hardware Implementation of Artificial Neurons Models Using FPGA. *Al-Rafidain Engineering*, Vol. 17, No.2.

Kim, Kyung-Joong & Cho, Sung-Bae. (2003). Prediction of Colon Cancer Using an Evolutionary Neural Network. *Journal from Elsevier*.

Klir, G.J., St. Clair U., Yuan, Bo. (1997). *Fuzzy Set Theory: Foundations and Applications*. Upper Saddle River, New Jersey: Prentice Hall International Inc.

Leondes, C.T. (1998). *Fuzzy Logic and Expert System Applications*. California: Academic Press.

Lin, C.-T., & Lee, G. (1996). *Neuro Fuzzy Systems*. Upper Saddle River, New Jersey: Prentice-Hall.

MathWorks. (1999). *Fuzzy Logic Toolbox for Use with Matlab (User's Guide Version 2)*. Natick: The MathWorks, Inc.

Mohanaiah, P., Sathyanarana, P. & Guru Kumar, L. (2013). Image Texture Feature Extraction Using GLCM Approach. *International Journal of Scientific and Research Publications*, Vol. 3, Hlm. 1-5.

Murra-Saca, J. *Vision and Image Analysis Group*. El Salvador: Cornell University. Diakses dari <http://www.gastrointestinalatlas.com/English/> pada tanggal 21 Maret 2013 pukul 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 12 Maret 2014 pukul 16.25 WIB.

11□

- Paulin, F. & Santhakumaran, A. (2010). Back Propagation Neural Network by Comparing Hidden Neurons: Case study on Breast Cancer Diagnosis. *International Journal of Computer Applications*, Vol. 2, No. 4.
- Prabowo Pudjo W. & Rahmadya Trias H. (2012). *Penerapan Soft Computing Dengan MATLAB*. Bandung: Rekayasa Sains.
- R.H. Sianipar. (2013) *Pemrograman Matlab dalam Contoh dan Penerapan*. Bandung : Informatika.
- Rama Diananda. (2008). *Mengenal Seluk-Beluk Kanker*. Jogjakarta: Katahari.
- Reni Setianingrum. (2014). Klasifikasi Stadium Kanker Kolorektal Menggunakan Model Recurrent Neural Network. *Skripsi Universitas Negeri Yogyakarta*.
- Ross, T. (2004). *Fuzzy Logic with Engineering Applications*. England: John Wiley & Sons Ltd.
- Samarangsinhe, S. (2007). *Neural Networks for Applied Sciences and Engineering*. New York: Auerbach Publications.
- Sarah Muharomah. (2014). *Karakteristik Demografi dan Klinikopatologi Pasien Kanker Kolorektal Usia Lanjut*. Yogyakarta: Repository Universitas Gadjah Mada.
- Sarle, W.S. (1994). Neural Networks and Statistical Models. *Proceeding of the Nineteenth Annual SAS Users Group International Conference*.
- Sharma, M. & Mukharjee, S. (2014). Fuzzy C-Means, ANFIS, and Genetic Algorithm for Segmenting Astrocytoma-A Tybe of Brain Tumor. *IAES International Journal of Artificial Intelligence*, Vol. 3, Hlm. 16-23.
- Siang, Jong Jek. (2009). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.
- Soh, L. & Tsatsoulis, C. (1999). Texture Analysis of SAR Sea Ice Imagery Using Gray Level Coourance Matrices. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37, No. 2, Hlm. 780-795.
- Sri Kusumadewi & Sri Hartati (2010). *Neuro Fuzzy: Integrasi Sistem Fuzzy & Jaringan Syaraf Edisi 2*. Yogyakarta: Graha Ilmu.
- Sri Kusumadewi (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*. Yogyakarta: Graha Ilmu.
- Sri Kusumadewi. (2003). *Artificial Intelligence: Teknik dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- Suyanto, S. M. (2008). *Soft Computing Membangun Mesin Ber-IQ Tinggi*. Bandung: Informatika.
- The Gastrointestinal Site. www.gastrolab.net. Diakses pada 03 Maret 2014 pukul 18.50 WIB.
- Usman Ahmad. (2005). *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
- Varalakshmi, K. (2013). Classification of Lung Cancer Nodules using a Hybrid Approach. *Journal of Emerging Trends in Computing and Information Sciences*. Vol. 4, No. 1.
- Wang, Li-Xin. (1997). *A Course in Fuzzy Systems and Control*. Upper Saddle River: Prentice-Hall International Inc.
- Wong, Hwee B. & Lim, Gek Hsiang. (2011). Measures of Diagnostic Accuracy: Sensitivity, Specificity, PPV and NPV. *Proceeding of Singapore Healthcare*, Vol. 20, No. 4, Hlm. 316-318.
- Zimmermann. (1991). *Fuzzy Sets Theory and its Applications 2nd Edition*. Massachusetts: Kluwer Academic Publishers.