

BAB II

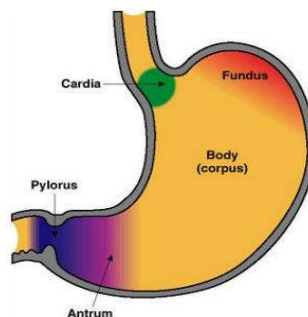
KAJIAN TEORI

Bab II berisi tentang kajian teori. Teori-teori yang digunakan dalam penelitian ini diantaranya mengenai kanker lambung, pengolahan citra digital dan ekstraksinya, *Fuzzy C-Means Clustering* (FCM), teori himpunan *fuzzy*, jaringan syaraf tiruan, *Adaptive Neuro Fuzzy Inference System* (ANFIS), uji ketepatan diagnosis, serta *Graphic User Interface* (GUI).

A. Kanker Lambung

1. Anatomi Lambung

Lambung atau perut besar (kantong nasi) adalah bagian dari saluran pencernaan yang dapat mekar, terletak di dalam rongga perut agak ke sebelah kiri atau dibawah diafragma, di depan pankreas, sedangkan limfe menempel pada sebelah kiri fundus. Bentuk, ukuran dan posisi sangat tergantung pada bentuk tubuh, sikap dan derajat peregangan lambung. Bila kosong, lambung menyerupai tabung berbentuk "J" dan bila terisi penuh, berbentuk seperti buah per raksasa. Kapasitas normal lambung adalah 1-2 liter (Irianto, 2004:179).



Gambar 2. 1 Anatomi Lambung (*American Cancer Society, 2017*)

Berdasarkan Gambar 2.1, terlihat bahwa lambung terdiri dari lima bagian sebagai berikut (*American Cancer Society, 2017*):

- a. *Cardia*: bagian pertama yang paling dekat dengan lubang yang ada disebelah kerongkongan.
- b. *Fundus*: bagian atas lambung yang terletak disebelah *cardio*.
- c. Tubuh (*corpus*): bagian utama lambung, terletak diantara *fundus* dan *antrum*.
- d. *Antrum*: bagian bawah lambung yang terletak di dekat usus.
- e. *Pylorus*: bagian terakhir dari lambung, yang berfungsi sebagai katup untuk mengendalikan pengosongan isi perut ke dalam usus kecil.

2. Pengertian Kanker Lambung

Kanker lambung atau kanker perut atau disebut juga kanker gaster adalah kanker yang terjadi ketika sel-sel abnormal (kanker) tumbuh pada lapisan lambung. Sebagian besar kanker lambung merupakan jenis adenokarsinoma (sel kanker yang mengeluarkan lendir serta cairan lainnya). Jenis kanker lambung lainnya adalah tumor karsinoid gastrointestinal, tumor stroma gastrointestinal, dan limfoma. Infeksi bakteri *H. pylori* adalah penyebab utama kanker lambung. Faktor risiko lain untuk kanker lambung antara lain gastritis kronis, usia tua, jenis kelamin laki-laki, banyak mengonsumsi makanan asin, diasapi, atau makanan yang diawetkan dan jarang mengonsumsi buah dan sayuran, merokok, anemia pernisiiosa, mempunyai riwayat operasi perut, serta mempunyai riwayat keluarga yang pernah menderita kanker lambung (*National Cancer Institute, 2006*).

Perkembangan kanker lambung cenderung lambat, sebelum sel kanker berkembang, perubahan pra-kanker sering terjadi pada lapisan dalam (mukosa)

lambung. Perubahan awal ini jarang menimbulkan gejala sehingga sering tidak terdeteksi (*American Cancer Society, 2017*).

Kanker lambung paling sering ditemukan pada stadium lanjut, kecuali di Asia Timur yang sudah melakukan deteksi menggunakan program *screening*. Deteksi dini kanker lambung sangat diperlukan untuk mengurangi jumlah penderita kanker lambung. Oleh karena itu di era sekarang ini sangat diperlukan program *screening* dan perawatan multimodalitas menggunakan kemoterapi, radioterapi yang canggih (Tsunehiro Takahashi, 2013).

3. Faktor Risiko Kanker Lambung

Seperti kanker pada umumnya, belum diketahui secara pasti apa penyebab utama munculnya sel abnormal yang lantas berkembang menjadi kanker pada dinding lambung. Namun, infeksi bakteri penyebab luka dinding lambung yaitu *H.pylori* diketahui dapat memicu timbulnya kanker melalui berbagai proses peradangan seperti gastritis. Kondisi kesehatan seperti mengalami penyakit anemia pernisiiosa dan memiliki polip pada dinding lambung juga meningkatkan risiko terjadinya kanker lambung. Hal lainnya yang juga meningkatkan risiko kanker lambung sebagai berikut (Czyzewska, 2013):

a. Usia

Jumlah penderita kanker lambung meningkat secara progresif seiring bertambahnya usia. Dari kasus yang didiagnosis antara tahun 2005 dan 2009 di Amerika Serikat, sekitar 1% kasus terjadi antara usia 20 dan 34 tahun, sedangkan 29% terjadi antara 75 dan 84 tahun (25). Selama periode ini, usia rata-rata saat diagnosis kanker lambung adalah 70 tahun.

b. Jenis Kelamin

Laki-laki memiliki risiko terkena kanker lambung lebih tinggi dibandingkan perempuan. Sekitar 1,8 sampai dengan 2 kali lebih tinggi laki-laki terkena kanker lambung dibandingkan dengan perempuan. Secara umum persentase menunjukkan 68% kasus lambung terjadi pada laki-laki dan hanya 32% terjadi pada perempuan.

c. Obesitas

Hasil penelitian menunjukkan 2,3 kali lipat peningkatan risiko tertular kanker lambung di kardiomi pada orang yang terkena obesitas. Telah ditunjukkan bahwa obesitas pada pria adalah faktor risiko perkembangan kanker lambung sekaligus perkembangan kanker colorectal, kanker hati dan kantong empedu. Pada gilirannya, obesitas pada wanita dikaitkan dengan peningkatan risiko pengembangan tumor hati, pankreas dan payudara.

d. Diet

Asupan makanan yang mengandung garam, asap ataupun yang diasinkan, misalnya daging panggang atau ikan asap diketahui meningkatkan faktor risiko terjadinya kanker lambung. Sedangkan makanan yang diawetkan, makanan yang kaya akan daging merah, serta kekurangan vitamin dan antioksidan diketahui mengurangi kemungkinan terjadinya kanker. Garam dan konsumsi makanan asin menyebabkan kenaikan risiko terjadinya kanker lambung 50% sampai dengan 100%. Hal ini terjadi karena natrium klorida merusak selaput lendir perut yang menyebabkan infeksi dan akibatnya memudahkan kolonisasi dan pertumbuhan *helicobacter pylori*. Mengurangi jumlah sayuran dan buah yang dikonsumsi dalam setiap hari juga dapat meningkatkan risiko terkena kanker lambung sekitar 30%

sampai dengan 50%. Hal ini terkait dengan efek antioksidasi dan zat yang terkandung dalam sayuran seperti asam askorbat (vitamin C), *karotenoid* dan *ta-copherol*. Mengurangi jumlah produk yang diawetkan secara kimiawi dalam makanan yang tertelan juga berkontribusi untuk mengurangi risiko terjadinya kanker lambung. Antioksidan seperti vitamin C dan E, beta-karoten, atau zat mikro seperti selenium, seng atau magnesium memiliki pelindung efek.

e. Penggunaan Alkohol dan Rokok

Penggunaan alkohol dan rokok diperkirakan meningkatkan risiko terkena kanker lambung, penderita kanker lambung di Indonesia naik sekitar lima kali lipat karena mengonsumsi rokok, hal ini disebabkan oleh efek karsinogenik yang terkandung di dalam rokok. Karsinogen adalah zat yang mampu membentuk ikatan kovalen dengan DNA, yang mengubah fungsi DNA yang benar dan dapat menyebabkan perkembangan kanker lambung. Demikian pula seseorang yang mengonsumsi alkohol memiliki risiko tinggi terkena kanker lambung dan tumor pada saluran pencernaan lainnya (kanker mulut, tenggorokan, laring dan kerongkongan). Etanol bukan merupakan zat yang mengandung karsinogen, namun zat nitrosamin terdapat pada minuman beralkohol, terutama di vodka, zat tersebut dapat meningkatkan risiko terkena kanker lambung.

f. Mengonsumsi obat-obatan

Aspirin adalah salah satu obat yang meningkatkan risiko terkena kanker lambung. Seseorang yang secara teratur mengonsumsi aspirin berisiko terjangkit kanker lambung meningkat hingga 30%. Aspirin meningkatkan permeabilitas

membran bakteri luar yang menyebabkan faktor risiko terkena kanker lambung meningkat.

g. Infeksi *Helicobacter Pylori* (*H. pylori*)

Infeksi kronis bakteri *helicobacter pylori* (bakteri yang menyebabkan ulkus peptikum atau tukak lambung) dapat meningkatkan faktor risiko terkena kanker lambung. Telah dipastikan bahwa pasien dengan infeksi *H. pylori* berisiko terkena kanker lambung meningkat tujuh kali lipat dibandingkan dengan seseorang yang tidak terinfeksi bakteri tersebut. Infeksi *H. pylori* berhubungan dengan gen CagA+ dan VacA yang berkontribusi terhadap munculnya risiko terjadinya karsinogenesis. Selain itu, *helicobacter pylori* menghasilkan urease yang memecah urea menjadi amonia dan karbon dioksida. Amonia menetralkan asam klorida yang terkandung dalam cairan perut yang menyebabkannya meningkatnya kadar pH sehingga memudahkan bakteri bertahan dan reproduksi. Peradangan kronis yang disebabkan oleh infeksi *helicobacter pylori* dapat mengakibatkan kerusakan pada DNA sel dengan mengganggu proses oksida sekaligus meningkatkan tingkat bentuk reaktif oksigen.

h. Infeksi Virus *Epstein-Barr*

Virus *Epstein-Barr* (EBV) adalah virus herpes gamma, yang menyebabkan *limfonitis oportunistik*. Prosentase seseorang yang terkena kanker lambung yang disebabkan oleh EBV berbeda-beda di setiap negara, berkisar antara 1,3% sampai dengan 20,1%, rata-rata sekitar 10%.

i. Status Sosial Ekonomi

Diseluruh dunia pada tahun 2002, tercatat sekitar 2/3 kasus kanker lambung berkembang di negara-negara yang kurang berkembang. Negara-negara di Afrika memiliki risiko tinggi terkena kanker lambung. Hal ini disebabkan karena diagnosis dan status perawatan medis yang tidak mencukupi di negara tersebut. Faktor status sosial lain juga tergantung dari jenis pekerjaannya, misalnya pekerjaan sebagai penjual daging, petani atau nelayan memiliki risiko terkena kanker lambung karena terpapar zat herbisida atau zat nitrat selama bekerja yang dapat meningkatkan risiko terkena kanker lambung.

j. Migrasi

Penelitian mengenai penurunan jumlah penderita kanker lambung diketahui karena migrasi dari suatu daerah yang memiliki risiko tinggi terkena kanker lambung ke daerah yang memiliki risiko rendah terkena kanker lambung. Penelitian yang dilakukan terhadap imigran Amerika Serikat dari negara-negara dengan risiko tinggi terkena kanker lambung seperti Jepang atau Polandia telah menunjukkan bahwa jumlah generasi warga Jepang atau Polandia yang terkena kanker lambung semakin menurun.

k. Penyakit Ulkus Peptik

Seseorang yang terkena kanker lambung sering disebabkan karena mempunyai penyakit ulkus peptik yang terdapat di perut atau duodenum.

l. Riwayat Keluarga Penderita Kanker Lambung

Seseorang yang mempunyai riwayat keluarga terkena kanker lambung juga memiliki risiko terkena kanker lambung. Faktor ini telah dipelajari di banyak daerah seperti Asia Timur, Amerika Utara, Eropa Utara dan negara-negara di wilayah

Mediterrania. Sekitar 10% sampai dengan 15% seseorang terkena kanker lambung memiliki riwayat keluarga yang terkena kanker lambung.

4. Gejala Kanker Lambung

Sebagian besar penderita kanker lambung memiliki gejala-gejala sebagai berikut (*American Cancer Society, 2017*):

- a. Nafsu makan menurun
- b. Penurunan berat badan
- c. Cepat kenyang saat makan
- d. Mulas atau gangguan pencernaan
- e. Mual
- f. Muntah

Kanker lambung pada stadium awal atau belum parah hanya mengalami muntah tanpa darah, namun apabila kanker lambung yang sudah pada tahap yang lebih parah, umumnya mengalami muntah darah

- g. Pembengkakan pada perut karena penumpukan cairan
- h. Anemia

Gejala kanker lambung pada tahap awal biasanya sulit dikenali karena gejalanya hampir sama dengan masalah lambung lainnya, misalnya tukak lambung. Karena itu diperlukan pemeriksaan dokter untuk memastikan diagnosis. Apabila seseorang merasakan gejala-gejala tersebut, seharusnya segera memeriksakan diri ke dokter agar dapat ditangani sedini mungkin.

5. Deteksi Dini Kanker Lambung

Menurut *National Cancer Institute* (2006), deteksi dini kanker lambung dapat dilakukan dengan metode sebagai berikut:

a. Pemeriksaan fisik dan riwayat

Pemeriksaan fisik diperlukan untuk memeriksa tanda-tanda umum kesehatan, termasuk memeriksa tanda-tanda yang tidak biasa seperti benjolan atau hal lain. Selain itu pemeriksaan riwayat penyakit seseorang juga diperlukan untuk mendeteksi adanya penyakit kanker lambung.

b. Tes darah

Tes darah merupakan prosedur dimana sampel darah diambil untuk mengukur jumlah zat tertentu yang dilepaskan ke dalam darah oleh organ dan jaringan di tubuh. Jumlah zat yang tidak biasa (lebih tinggi atau lebih rendah dari normal) bisa menjadi tanda-tanda seseorang mengidap penyakit tertentu.

c. *Complete Blood Count (CBC)*

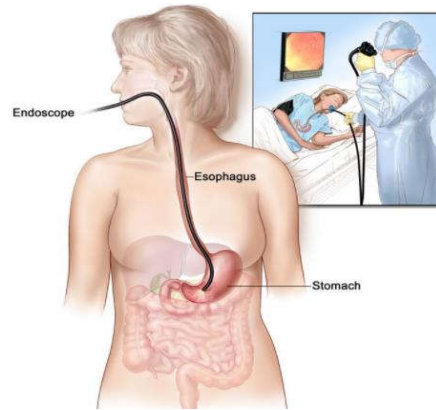
Prosedur di mana sampel darah ditarik dan diperiksa sebagai berikut:

- (1) Jumlah sel darah merah, sel darah putih, dan trombosit.
- (2) Jumlah hemoglobin (protein yang membawa oksigen) di dalam sel darah merah.
- (3) Bagian sampel terdiri dari sel darah merah.

d. *Upper Endoscopy*

Upper Endoscopy merupakan proses untuk melihat bagian kerongkongan, perut, dan duodenum untuk memeriksa area abnormal. Endoskopi (tabung tipis dan ringan) dilewatkan melalui mulut ke tenggorokan kemudian ke kerongkongan.

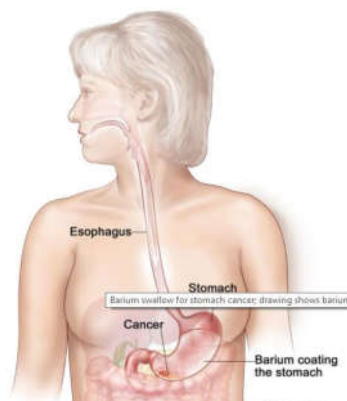
Gambar 2.2 merupakan proses deteksi dini kanker lambung menggunakan proses *upper endoscopy*.



Gambar 2. 2 Proses *Upper Endoscopy*

e. *Barium Swallow*

Barium Swallow merupakan proses dimana pasien minum cairan yang mengandung *barium* (senyawa logam putih perak) dan mengalir melalui kerongkongan dan masuk ke dalam perut. Kemudian hasil serangkaian sinar-X pada lapisan cairan kerongkongan dan perut diambil untuk mencari daerah abnormal. Gambar 2.3 merupakan proses deteksi dini kanker lambung menggunakan proses *barrium swallow*.



Gambar 2. 3 Proses *Barium Swallow*

f. CT-Scan

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.

g. *Biopsy*

Biopsy adalah proses pengangkatan sel atau jaringan sehingga bisa dilihat dengan menggunakan mikroskop untuk memeriksa tanda-tanda kanker. *Biopsy* perut biasanya dilakukan selama proses *endoscopy*.

6. Klasifikasi Kanker Lambung

Menurut *National Cancer Institute* (2006), klasifikasi stadium kanker lambung adalah sebagai berikut:

a. Stadium 0

Sel kanker hanya terdapat pada lapisan sel dinding lambung. Gejala-gejala pada kanker lambung stadium 0 sulit dikenali, sehingga apabila sudah diketahui lebih awal bahwa seseorang sudah terkena kanker lambung maka dapat melakukan penanggulangan lebih awal juga. Sehingga penyebaran sel-sel kanker pun dapat dicegah untuk tidak mengganggu organ lainnya.

b. Stadium I

Apabila kanker pada stadium 0 tidak ditemukan penanggulangan secara tepat, maka sel kanker akan terus berkembang. Perkembangan sel kanker ini menyebabkan stadium 0 akan meningkat menjadi stadium I. Stadium I dibagi

menjadi dua bagian. Bagian yang pertama menunjukkan penyebaran sel kanker masih di organ lambung, sedangkan bagian kedua telah menjangkiti kelenjar di sekitar organ.

c. Stadium II

Pada peningkatan stadium I ke stadium II, pertumbuhan sel kanker sudah mulai tidak dapat ditanggulangi. Kanker biasanya telah menyebar ke seluruh dinding lambung dan seluruh kelenjar di sekitarnya.

d. Stadium III

Stadium III merupakan salah satu jenis stadium yang paling mengkhawatirkan. Sel kanker pada stadium III ini menyebar ke seluruh bagian mulai dari organ, kelenjar, bahkan otot telah merasakan akibat dari penyebaran kanker tersebut, Biasanya penyebaran sel kanker pada tahap ini pun kemudian menjadi sangat agresif dan penuh efek membahayakan.

e. Stadium IV

Pada tahapan kanker ini, biasanya penderita telah mulai tidak bisa melakukan banyak hal. Aktivitas telah sangat terbatas dan proses pengobatan juga sudah semakin sulit dilakukan. Beberapa operasi mungkin juga dilakukan dengan resiko tinggi seperti kegagalan dan meninggal.

B. Pengolahan Citra Digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi, warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang

optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan *output*-nya adalah citra hasil pengolahan (Sutoyo & Mulyanto, 2009:5).

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu (Putra, 2010:19).

1. Jenis-jenis Citra Digital

Nilai suatu pixel memiliki nilai dalam rentang waktu tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0-255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut ini adalah jenis-jenis citra berdasarkan nilai pixelnya (Putra, 2010:39-44):

a. Citra Biner

Citra biner adalah citra digital yang memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (*black and white*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner. Citra biner seringkali muncul sebagai hasil dari proses pengolahan citra seperti segmentasi, morfologi, ataupun *dithering*.

b. Citra *Grayscale*

Citra *Grayscale* merupakan citra yang hanya memiliki satu nilai kanal pada setiap *pixel*-nya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih.

c. Citra Warna (8 bit)

Setiap piksel dari citra warna 8 bit hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu.

d. Citra Warna (16 bit)

Citra warna 16 bit (biasanya disebut sebagai citra *highcolor*) dengan setiap pikselnya diwakili dengan 2 *byte memory* (16 bit). Warna 16 bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau.

e. Citra Warna (24 bit)

Setiap piksel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup. Untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna

saja. Setiap poin informasi piksel (RGB) disimpan ke dalam 1 *byte* data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.

2. Peningkatan Kualitas Citra (*Image Enhancement*)

Peningkatan kualitas citra adalah suatu proses untuk mengubah sebuah citra menjadi citra baru sesuai dengan kebutuhan melalui berbagai cara. Cara-cara yang bisa dilakukan misalnya dengan fungsi transformasi, operasi matematis, pemfilteran, dan lain-lain. Tujuan utama dari peningkatan kualitas citra adalah untuk memproses citra sehingga citra yang dihasilkan lebih baik daripada citra aslinya untuk aplikasi tertentu (Sutoyo & Mulyanto, 2009:30).

Salah satu metode yang sering digunakan untuk peningkatan kualitas citra (*image enhancement*) adalah metode *histogram equalization*. Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna. Bila digambarkan pada koordinat kartesian maka sumbu X (absis) menunjukkan tingkat warna dan sumbu Y (ordinat) menunjukkan frekuensi kemunculan (Sutoyo & Mulyanto, 2009:29). Sedangkan *histogram equalization* adalah mengubah nilai-nilai intensitas citra sehingga penyebarannya seragam. Teknik ini dapat dilakukan pada keseluruhan citra atau pada beberapa bagian citra saja. Histogram hasil proses ekualisasi tidak akan seragam atau sama untuk seluruh intensitas. Teknik ini hanya melakukan distribusi ulang terhadap distribusi dari intensitas dari histogram awal. Jika histogram awal memiliki beberapa puncak dan lembah maka histogram hasil ekualisasi akan tetap memiliki puncak dan lembah. Akan tetapi puncak dan lembah tersebut akan digeser. Histogram hasil ekualisasi akan lebih disebarakan (*spreading*).

Distribusi ulang terhadap histogram awal dilakukan dengan memetakan setiap nilai piksel pada histogram awal menjadi nilai piksel baru dengan cara sebagai berikut.

$$n(g) = \max\left(0, \text{round}\left[\left(L - 1\right) * \frac{c(g)}{N}\right] - 1\right) \quad (2.1)$$

dengan $n(g)$ adalah nilai piksel baru, N menyatakan banyaknya piksel pada citra (bila citra berukuran 8×8 maka N adalah 64), g menyatakan nilai *gray level* awal yang nilainya dari $1 \dots L - 1$ (L menyatakan nilai *gray level* maksimum). Sedangkan $c(g)$ menyatakan banyaknya piksel yang memiliki nilai sama dengan g atau kurang yang secara matematis dapat dinyatakan sebagai:

$$c(g) = \sum_{i=1}^g h(i), \quad g = 1, 2, \dots, L - 1 \quad (2.2)$$

dengan $h(i)$ menyatakan histogram awal (Putra, 2010:130).

Transformasi Fourier dapat digunakan sebagai perbaikan citra atau peningkatan kualitas citra. Transformasi Fourier mengubah suatu citra dari domain spasial menjadi domain frekuensi. Citra hasil proses transformasi dapat dianalisis kembali, diinterpretasikan, dan dijadikan acuan untuk melakukan pemrosesan selanjutnya. Transformasi Fourier yang biasa diterapkan pada data citra adalah Transformasi Fourier Diskrit (*Discrete Fourier Transform*).

a. Transformasi Fourier Diskrit (DFT 2-D)

Transformasi Fourier Diskrit 2-D dari sebuah fungsi diskrit $f(x, y)$ dinyatakan sebagai berikut (Gonzales & Woods, 2002:154):

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-2\pi j \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (2.3)$$

untuk $u = 0, 1, 2, \dots, M - 1$, $v = 0, 1, 2, \dots, N - 1$.

Sedangkan transformasi balik ke domain spasial dapat dinyatakan sebagai berikut:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[2\pi j \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (2.4)$$

untuk $x = 0, 1, 2, \dots, M - 1$, $y = 0, 1, 2, \dots, N - 1$.

Misalkan $R(u, v)$ dan $I(u, v)$ adalah komponen real dan imajiner dari $F(u, v)$, spektrum fourier atau yang dikenal sebagai *magnitude* didefinisikan sebagai berikut:

$$|F(u, v)| = \sqrt{[R^2(u, v) + I^2(u, v)]} \quad (2.5)$$

dan sudut fase transformasi didefinisikan sebagai berikut:

$$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]. \quad (2.6)$$

Persamaan 2.5 dan 2.6 digunakan untuk merepresentasikan $F(u, v)$ dalam representasi polar kuantitas kompleks:

$$F(u, v) = |F(u, v)|e^{-j\phi(u,v)}. \quad (2.7)$$

Power spektrum didefinisikan sebagai kuadrat dari *magnitude* ebagai berikut:

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v). \quad (2.8)$$

Secara umum, untuk menentukan transformasi fourier pada suatu citra perlu dilakukan operasi perkalian pada fungsi *input* citra $f(x, y)$ dengan $(-1)^{x+y}$ terlebih dahulu sebagai berikut (Gonzales & Woods, 2002:154):

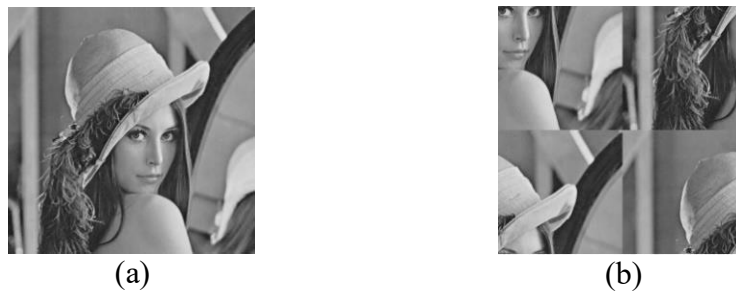
$$\mathfrak{F}[f(x, y)(-1)^{x+y}] = F\left(u - \frac{M}{2}, v - \frac{N}{2}\right) \quad (2.9)$$

dimana $\mathfrak{F}(\cdot)$ merupakan transformasi fourier. Persamaan () menyatakan bahwa titik origin pada transformasi fourier dari suatu citra adalah $F(0,0)$ berada pada titik koordinat suatu citra yaitu $u = \frac{M}{2}$ dan $v = \frac{N}{2}$. Dengan kata lain, melakukan operasi perkalian $f(x, y)$ dengan $(-1)^{x+y}$ adalah memindahkan titik origin dari $F(u, v)$ pada koordinat frekuensi $(\frac{M}{2}, \frac{N}{2})$ yang merupakan pusat daerah transformasi diskrit fourier 2-D berukuran $M \times N$.

b. Konsep Filter dalam Domain Frekuensi

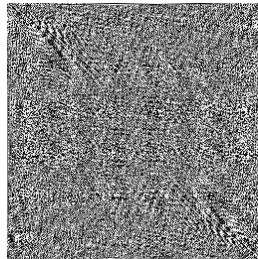
Pada dasarnya, ide dalam pemfilteran pada domain frekuensi adalah untuk memilih fungsi transfer filter yang memodifikasi $F(u, v)$ dengan cara tertentu. Berikut ini merupakan langkah-langkah melakukan *filtering* pada domain frekuensi (Gonzales & Woods, 2009:158):

- a. Kalikan citra *input* dengan $(-1)^{x+y}$ untuk memusatkan transformasi.
- Pada Gambar 2.4 (a) merupakan citra *input*, sedangkan Gambar 2.4 (b) merupakan perkalian citra *input* (a) dengan $(-1)^{x+y}$.



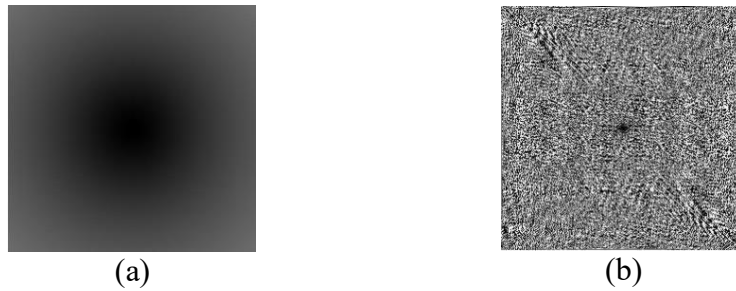
Gambar 2. 4 Citra Lena.jpg

- b. Hitung $F(u, v)$ pada Persamaan 2.3, DFT 2-D dari citra pada langkah 1.
- Pada Gambar 2.5 merupakan spektrum fourier dari citra pada Gambar 2.4 (b).
- Untuk menghitung spektrum fourier pada suatu citra menggunakan Persamaan 2.3.



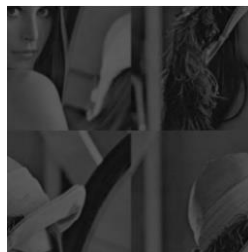
Gambar 2. 5 Spektrum Fourier Citra

- c. Kalikan $F(u, v)$ dengan fungsi filter $H(u, v)$.
- Pada Gambar 2.6 (a) merupakan spektrum fourier untuk fungsi filter $H(u, v)$.
- Hasil perkalian antara $F(u, v)$ dan $H(u, v)$ ditunjukkan pada Gambar 2.6 (b)



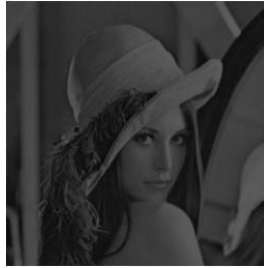
Gambar 2. 6 Spektrum Fourier (a) Spektrum Fourier Filter $H(u, v)$ (b) Spektrum Fourier hasil perkalian $F(u, v)$ dengan $H(u, v)$

- d. Hitung invers DFT 2-D dari citra pada langkah 3 dengan Persamaan 2.4. Untuk menghitung transformasi balik suatu citra dari domain frekuensi ke dalam domain spasial dengan menggunakan Persamaan 2.4. Selanjutnya menggunakan bagian real dari citra. Gambar 2.7 merupakan citra hasil transformasi balik dari citra pada langkah 3.



Gambar 2. 7 Citra Hasil Transformasi Balik dari Citra pada Langkah 3

- e. Kalikan hasil pada langkah 4 dengan $(-1)^{x+y}$. Pada Gambar 2.8 merupakan citra hasil akhir dari perbaikan kualitas citra pada Gambar 2.4 (a) dengan menggunakan domain frekuensi.



Gambar 2. 8 Citra Hasil Perbaikan Kualitas Citra pada Domain Frekuensi

Misalkan $F(u, v)$ dan $H(u, v)$ merupakan Transformasi Fourier dari $f(x, y)$ dan $h(x, y)$. Pernyataan sederhana dari teorema konvolusi bahwa $f(x, y) * h(x, y)$ dan $F(u, v)H(u, v)$ menyatakan pasangan Transformasi Fourier. Hal tersebut dapat dinyatakan sebagai berikut (Gonzales & Woods, 2002:162):

$$f(x, y) * h(x, y) \leftrightarrow F(u, v)H(u, v). \quad (2.10)$$

Simbol " * " menyatakan konvolusi dari dua fungsi dan simbol " \leftrightarrow " digunakan untuk menyatakan bahwa pada sisi kiri Persamaan 2.10 yang merupakan konvolusi pada domain spasial $[f(x, y) * h(x, y)]$ dapat diperoleh dengan mentransformasi balik hasil perkalian dari Transformasi Fourier $[F(u, v)H(u, v)]$ pada sisi kanan Persamaan 2.10. Sebaliknya, pada Persamaan 2.11 menyatakan bahwa pada sisi kanan Persamaan 2.11 yang merupakan konvolusi $[F(u, v)H(u, v)]$ dapat diperoleh dengan melakukan Transformasi Fourier pada sisi kiri Persamaan 2.11.

$$f(x, y)h(x, y) \leftrightarrow F(u, v) * H(u, v). \quad (2.11)$$

c. Filter Penajaman pada Domain Frekuensi

Filter penghalusan (*smoothing*) pada domain frekuensi dilakukan dengan pelemahan komponen frekuensi tinggi dari transformasi fourier atau yang disebut

dengan *lowpass filter*. Berbeda pada filter penajaman (*sharpening*) pada domain frekuensi dapat dicapai dengan melemahkan frekuensi rendah tanpa mengganggu informasi frekuensi tinggi dalam transformasi fouriernya. Filter penajaman biasanya disebut sebagai *highpass filter*. Filter *highpass* diperoleh dari filter *lowpass* dengan formula berikut (Gonzales & Woods, 2002: 180):

$$H_{hp}(u, v) = 1 - H_{lp}(u, v) \quad (2.12)$$

dengan

$H_{lp}(u, v)$ = fungsi transfer filter *lowpass*.

Butterworth highpass filter merupakan salah satu filter *highpass* pada domain frekuensi. Pada *Butterworth Highpass Filter* (BHPF) menampilkan citra yang lebih halus jika dibandingkan dengan *Ideal Highpass Filter* (IHPF). BHPF dengan order n , *cutt-off* frekuensi (D_0) dan jarak sembarang titik (u, v) ke titik origin ($D(u, v)$) didefinisikan sebagai berikut (Gonzales & Woods, 2002: 183):

$$H_{bt}(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}} \quad (2.13)$$

dengan

$$D(u, v) = \sqrt{\left[\left(u - \frac{M}{2} \right)^2 + \left(v - \frac{N}{2} \right)^2 \right]}.$$

d. High Frequency Emphasis Filter (HFEF)

Metode *High Frequency Emphasis Filter* (HFEF) merupakan salah satu metode untuk meningkatkan kualitas citra, menurut Sun (2009:1) terkadang dalam melakukan perbaikan kualitas citra pada domain frekuensi dapat difokuskan pada

komponen frekuensi tinggi suatu citra tersebut. Pada kasus ini dengan melakukan operasi perkalian pada fungsi *highpass* dengan suatu konstanta (b) dan menambahkan suatu *offset* (a) maka pendekatan ini disebut dengan *High Frequency Emphasis Filtering* (HFEF). Nilai konstanta pengali meningkatkan amplitudo frekuensi rendah, tetapi frekuensi rendah menerima efek *enhancement* yang lebih sedikit daripada frekuensi tinggi, dengan $a \geq 0$, $b > a$. Fungsi transfer filter *high frequency emphasis* diberikan sebagai berikut:

$$H_{hfe}(u, v) = a + bH_{hp}(u, v) \quad (2.14)$$

dengan

$H_{hp}(u, v)$ = fungsi transfer filter *highpass*.

3. Ekstraksi Fitur Citra

a. *Gray Level Co-Occurrence Matrix* (GLCM)

Gray Level Co-Occurrence Matrix (GLCM) adalah matriks derajat keabuan yang merepresentasikan hubungan suatu derajat keabuan dengan derajat keabuan lain. GLCM merupakan dasar dari teknik tekstur *haralick*, GLCM digunakan untuk analisis pasangan piksel yang bersebelahan tergantung dengan sudut yang digunakan (Pratama & Nyoman, 2017). Langkah pertama untuk menghitung fitur-fitur GLCM adalah dengan mengubah citra RGB menjadi citra berskala keabuan. Langkah kedua adalah menciptakan matrik *co-occurrence* dan dilanjutkan dengan menentukan hubungan spasial antara piksel referensi dan piksel tetangga berdasarkan sudut θ dan jarak d . Langkah selanjutnya adalah menciptakan matrik

simetris dengan menambahkan matrik *co-occurrence* dengan matrik transposenya. Kemudian dilakukan normalisasi terhadap matrik simetris dengan menghitung probabilitas setiap elemen matrik. Langkah terakhir adalah dengan menghitung fitur GLCM. Setiap fitur dihitung dengan satu piksel jarak di empat arah, yaitu $0^0, 45^0, 90^0$ dan 135^0 untuk mendeteksi *co-occurrence* [3], [5], [9], [11], [12], [13], [14], [15], [16], [17], [18], [19] (Satrio & Sutojo, 2010).

Ciri statistik yang dapat diekstraksi dari metode GLCM dengan menggunakan bantuan software Matlab antara lain *contrast*, *correlation*, *dissimilarity*, *entropy*, *homogeneity*, dan *difference entropy*.

(1) *Contrast*

Rumus *contrast* adalah sebagai berikut (Gadkari, 2004):

$$contrast = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - j)^2 P_{ij}. \quad (2.15)$$

dimana,

i, j = koordinat spasial fungsi P_{ij} .

P_{ij} = entri pada GLCM baris ke- i dan kolom ke- j

N_g = banyak tingkat keabuan pada suatu citra.

(2) *Correlation*

Rumus *correlation* adalah sebagai berikut (Gadkari, 2004):

$$correlation = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (ij)P_{ij} - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.16)$$

dimana,

i, j = koordinat spasial fungsi P_{ij} .

P_{ij} = entri pada GLCM baris ke- i dan kolom ke- j

N_g = banyak tingkat keabuan pada suatu citra.

$$\mu_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} iP(i, j).$$

$$\mu_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} jP(i, j).$$

$$\sigma_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 P(i, j).$$

$$\sigma_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (j - \mu_y)^2 P(i, j).$$

(3) *Dissimilarity*

Rumus *dissimilarity* adalah sebagai berikut (Abouelatta, 2013):

$$dissimilarity = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i, j) |i - j| \quad (2.17)$$

dimana,

i, j = koordinat spasial fungsi P_{ij} .

P_{ij} = entri pada GLCM baris ke- i dan kolom ke- j

N_g = banyak tingkat keabuan pada suatu citra.

(4) *Entropy*

Rumus *entropy* adalah sebagai berikut (Gadkari, 2004):

$$entropy = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_{ij} \log(P_{ij}) \quad (2.18)$$

dimana,

i, j = koordinat spasial fungsi P_{ij} .

P_{ij} = entri pada GLCM baris ke- i dan kolom ke- j

N_g = banyak tingkat keabuan pada suatu citra.

(5) *Homogeneity*

Rumus *homogeneity* adalah sabagai berikut (Girisha *et al*, 2013):

$$homogeneity = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1 + (i - j)^2} P_{ij} \quad (2.19)$$

dimana,

i, j = koordinat spasial fungsi P_{ij} .

P_{ij} = entri pada GLCM baris ke- i dan kolom ke- j

N_g = banyak tingkat keabuan pada suatu citra.

(6) *Difference Entropy*

Rumus *difference entropy* adalah sebagai berikut (Haralick, Shanmugam &

Dinstein, 1973):

$$DE = - \sum_{i=0}^{N_g-1} P_{x-y}(k) \log \{P_{x+y}(k)\} \quad (2.20)$$

dimana,

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

N_g = banyak tingkat keabuan pada suatu citra.

b. *Grey Level Run Length Matrix (GLRLM)*

Grey Level Run Length Matrix (GLRLM) merupakan salah satu metode populer untuk mengekstrak tekstur sehingga diperoleh ciri statistik atau atribut yang terdapat dalam tekstur dengan mengestimasi piksel-piksel yang memiliki derajat keabuan yang sama. Ekstraksi tekstur dengan metode *run-length* dilakukan dengan membuat rangkaian pasangan nilai (i, j) pada setiap baris piksel. Orientasi dibentuk dengan empat arah pergeseran dengan interval 45° , yaitu $0^\circ, 45^\circ, 90^\circ, 135^\circ$ (Jefry)

Berdasarkan penelitian yang dilakukan oleh P.T. Bharati & P.Subashini (2013), terdapat beberapa jenis ciri tekstural yang dapat diekstraksi dari *matrix run-length*. Berikut variabel-variabel yang terdapat metode statistik *Grey Level Run Length Matrix*:

1. *Short Run Emphasis* (SRE):

$$SRE = \frac{1}{n_r} \sum_{j=1}^N \frac{P_r(j)}{j^2}. \quad (2.21)$$

2. *Long Run Emphasis* (LRE):

$$LRE = \frac{1}{n_r} \sum_{j=1}^N P_r(j) \cdot j^2. \quad (2.22)$$

3. *Gray-Level Nonuniformity* (GLN):

$$GLU = \frac{1}{n_r} \sum_{i=1}^M P_g(j)^2. \quad (2.23)$$

4. *Run Percentage* (RP):

$$RP = \frac{n_r}{n_p}. \quad (2.24)$$

5. *Run Length Nonuniformity* (RLN):

$$RLN = \frac{1}{n_r} \sum_{j=1}^N P_r(i)^2. \quad (2.25)$$

6. *Low Gray-Level Run Emphasis* (LGRE):

$$LGRE = \frac{1}{n_r} \sum_{i=1}^M \frac{P_g(i)}{i^2}. \quad (2.26)$$

7. *Hogh Gray-Level Run Emphasis* (HGRE):

$$HGRE = LRE = \frac{1}{n_r} \sum_{i=1}^M P_g(i) \cdot i^2 \quad (2.27)$$

dimana,

i = nilai derajat keabuan.

j = piksel yang berurutan (*run*).

n_r = jumlah derajat keabuan pada sebuah gambar.

n_p = jumlah piksel berurutan pada sebuah gambar.

$P_r(j)$ = himpunan matriks i dan j dengan mempertimbangkan tingkat keabuan pada sebuah gambar.

$P_g(j)$ = himpunan matriks i dan j tanpa mempertimbangkan tingkat keabuan pada sebuah gambar.

C. Fuzzy C-Means (FCM) Clustering

Fuzzy clustering merupakan salah satu teknik yang digunakan untuk menentukan *cluster* optimal dalam suatu ruang vektor yang didasarkan pada bentuk normal *Euclidian* untuk jarak antar vektor. *Fuzzy clustering* sangat berguna bagi pemodelan *fuzzy* terutama dalam mengidentifikasi aturan-aturan *fuzzy* (Kusumadewi & Purnomo, 2010:79).

Fuzzy C-Means (FCM) adalah suatu teknik pengelompokan data yang mana keberadaan tiap-tiap titik data dalam suatu kelompok (*cluster*) ditentukan oleh derajat keanggotaan. Teknik ini pertama kali diperkenalkan oleh Jim Bezdek pada tahun 1998 (Zimmermann, 2001). Beberapa kelebihan dari *Fuzzy C-Means* antara lain memberikan hasil yang terbaik untuk kumpulan data yang *overlap* serta ditetapkan derajat keanggotaan untuk setiap pusat *cluster*, akibatnya memungkinkan untuk titik data masuk ke lebih dari satu pusat *cluster* meskipun dalam proses perhitungan FCM membutuhkan banyak proses iterasi, perhitungan yang lama, dan sensitif terhadap inisialisasi jumlah *cluster* (Dang Q.A. et al, 2015).

Algoritma *Fuzzy C-Means* (FCM) adalah sebagai berikut (Kusumadewi & Purnomo, 2010:80):

1. Input data yang akan di *cluster* X , berupa matriks $n \times m$ (n = jumlah sampel data, m = atribut setiap data).
 X_{ij} = data sampel ke i ($i = 1, 2, \dots, n$), atribut ke- j ($j = 1, 2, \dots, m$).
2. Tentukan:
 - a. Jumlah cluster = c ;
 - b. Pangkat (bobot) = w ;
 - c. Maksimum iterasi = $MaxIter$;
 - d. Error terkecil yang dihadapkan = ε ;
 - e. Fungsi obyektif awal = $P_0 = 0$;
 - f. Iterasi awal = $t = 1$;
3. Bangkitkan bilangan random μ_{ik} , $i = 1, 2, \dots, n$; $k = 1, 2, \dots, nc$; sebagai elemen-elemen matriks partisi awal U .

Menghitung jumlah setiap kolom (atribut):

$$Q_j = \sum_{k=1}^c \mu_{ik} \quad (2.28)$$

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

$$\mu_{ik} = \frac{\mu_{ik}}{Q_j}. \quad (2.29)$$

4. Menghitung pusat *cluster* ke- k : V_{kj} , dengan $k = 1, 2, \dots, c$; dan $j = 1, 2, \dots, m$

$$V_{ij} = \frac{\sum_{i=1}^n ((\mu_{ik})^w \cdot X_{kj})}{\sum_{i=1}^n (\mu_{ik})^w}. \quad (2.30)$$

5. Hitung fungsi obyektif pada iterasi ke- k ,

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \left(\left[\sum_{j=1}^m (X_{ij} - V_{ij})^2 \right] (\mu_{ik})^w \right). \quad (2.31)$$

6. Hitung perubahan matriks partisi:

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (X_{ij} - V_{ij})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[\sum_{j=1}^m (X_{ij} - V_{ij})^2 \right]^{\frac{-1}{w-1}}} \quad (2.32)$$

dengan $i = 1, 2, \dots, n$ dan $k = 1, 2, \dots, c$.

7. Cek kondisi berhenti

Jika: $(|P_t - P_{t-1}| < \varepsilon)$ atau $(t > Maxlter)$ maka berhenti; sedangkan jika tidak: $t = t + 1$, ulangi langkah ke-4.

Contoh 2.1 Sebagai contoh klasifikasi menggunakan ekstraksi fitur GLRLM dengan metode FCM.

Tabel 2.1 Data Ekstraksi Fitur GLRLM

Data ke- <i>i</i>	Atribut	
	GLN (X_1)	LGRE (X_2)
1	10049,04	91,51664
2	19215,57	77,13872
3	15846,32	87,6038
4	18856,44	71,26874

Langkah 1, *input* data yang akan dikluster X yang berupa matriks berukuran $n \times m$. Data Tabel 2.1 memuat dua variabel X_1 dan X_2 untuk masing-masing memiliki empat item 1,2,3, dan 4.

Langkah 2, data dikelompokkan menjadi 2 *cluster* yang dinotasikan dengan $k = 2$, dengan pangkat atau bobot $w = 2$, maksimal iterasi = 100, faktor koreksi (*error* terkecil) = 10^{-5} , fungsi objektif awal ($t = 0$), $J_{(0)} = 0$.

Langkah 3, membangkitkan bilangan random $\mu_{ik}, i = 1, 2, \dots, c$; sebagai elemen-elemen matriks partisi awal U .

$U = \text{initfcm}(4, 2)$

$$U_{(0)} = \begin{bmatrix} 0,295096 & 0,406589 \\ 0,328081 & 0,062716 \\ 0,045995 & 0,179066 \\ 0,330828 & 0,351629 \end{bmatrix}$$

Langkah 4, menghitung pusat *cluster* yang terbentuk berdasarkan matriks partisi awal.

Tabel 2.2 Pusat *Cluster* Pertama pada Iterasi ke-1

μ_{s1}	x_{s1}	x_{s2}	μ_{s1}^2	$\mu_{s1}^2 \times x_{s1}$	$\mu_{s1}^2 \times x_{s2}$
0,295096	10049,04	91,51664	0,087082	875,086	7,969411
0,328081	19215,57	77,13872	0,107637	2068,31	8,302995
0,045995	15846,32	87,6038	0,002116	33,52364	0,18533
0,330828	18856,44	71,26874	0,109447	2063,784	7,800162
			0,306281	5040,704	24,2579
$v_1 = \frac{\sum_{s=1}^4 (\mu_{s1})^2 x_{s1}}{\sum_{s=1}^4 (\mu_{s1})^2}$				16457,75	79,20133

Tabel 2.3 Pusat *Cluster* Kedua pada Iterasi ke-1

μ_{s2}	x_{s1}	x_{s2}	μ_{s2}^2	$\mu_{s2}^2 \times x_{s1}$	$\mu_{s2}^2 \times x_{s2}$
0,406589	10049,04	91,51664	0,165314	1661,252	15,12903
0,062716	19215,57	77,13872	0,003933	75,57977	0,303406
0,179066	15846,32	87,6038	0,032065	508,1084	2,808995
0,351629	18856,44	71,26874	0,123643	2331,467	8,811883
			0,324956	4576,407	27,05331
$v_1 = \frac{\sum_{s=1}^4 (\mu_{s1})^2 x_{s1}}{\sum_{s=1}^4 (\mu_{s1})^2}$				14083,18	83,25235

Sehingga pusat *cluster* yang terbentuk adalah

$$V_1 = \begin{bmatrix} 16457,75 & 79,20133 \\ 14083,18 & 83,25235 \end{bmatrix}$$

Fungsi objektif yang dihasilkan adalah

$$J_1 = \sum_{i=1}^2 \sum_{i=2}^4 \sum_{i=1}^2 (x_{si} - V_{ji})^2 (x_{sj})^2.$$

$$J_1 = 165955762,9$$

Tabel 2.4 Perhitungan Fungsi Objektif pada Iterasi ke-1

μ_{s1}^2	μ_{s2}^2	$\sum_{i=1}^2 (x_{s1}-V_{1i})^2$	$\sum_{i=1}^2 (x_{s2}-V_{2i})^2$	$\mu_{s1}^2 \cdot d_{s1}$	$\mu_{s2}^2 \cdot d_{s2}$	$L_1 + L_2$
		d_{s1}	d_{s2}	L_1	L_2	L_T
0,08708 2	0,16531 4	12681657,3	195535390, 4	110434 0	32324857, 76	33429197, 39
0,10763 7	0,00393 3	479355620	196340425, 9	515964 69	772265,14 07	52368734, 31
0,00211 6	0,03206 5	229694984	195754310, 8	485928, 9	6276790,0 08	6762718,9 39
0,10944 7	0,12364 3	448420073	196669568	490783 06	24316806, 28	73395112, 28
J_1						165955762 ,9

Karena $|J_1 - J_0| = |165955762,9 - 0| = 165955762,9 > \varepsilon = 10^{-5}$ dan $t = 1 < MaxIter = 100$, maka proses dilanjutkan ke Iterasi ke-2 dengan menghitung perubahan matriks partisi

Tabel 2.5 Perhitungan Matriks Partisi pada Iterasi ke-1

d_{s1}^{-1}	d_{s2}^{-1}	Total	$\frac{M1}{MT}$	$\frac{M2}{MT}$
$M1$	$M2$	MT	μ_{s1}	μ_{s2}
7,8854E-08	5,11416E-09	8,39682E-08	0,939094	0,06090595
2,0861E-09	5,09319E-09	7,17933E-09	0,290575	0,70942493
4,3536E-09	5,10844E-09	9,46204E-09	0,460112	0,53988804
2,2301E-09	5,08467E-09	7,31472E-09	0,304872	0,69512831

Diperoleh matriks partisi yang baru yaitu

$$U_1 = \begin{bmatrix} 0,939094 & 0,06090595 \\ 0,290575 & 0,70942493 \\ 0,460112 & 0,53988804 \\ 0,304872 & 0,69512831 \end{bmatrix}$$

Iterasi ke-2

Selanjutnya menghitung pusat-pusat *cluster* dengan matriks yang baru

Tabel 2.6 Pusat *Cluster* Pertama pada Iterasi ke-2

μ_{s1}	x_{s1}	x_{s2}	μ_{s1}^2	$\mu_{s1}^2 \times x_{s1}$	$\mu_{s1}^2 \times x_{s2}$
0,939094	10049,04	91,51664	0,881898	8862,22366	80,70829976
0,290575	19215,57	77,13872	0,084434	1622,44418	6,513117619
0,460112	15846,32	87,6038	0,211703	3354,71432	18,54599187
0,304872	18856,44	71,26874	0,092947	1752,64833	6,624211043
			1,270981	15592,0305	112,3916203
$v_2 = \frac{\sum_{s=1}^4 (\mu_{s1})^2 x_{s1}}{\sum_{s=1}^4 (\mu_{s1})^2}$				12267,7098	88,42900754

Tabel 2.7 Pusat *Cluster* Kedua pada Iterasi ke-2

μ_{s2}	x_{s1}	x_{s2}	μ_{s2}^2	$\mu_{s2}^2 \times x_{s1}$	$\mu_{s2}^2 \times x_{s2}$
0,06090595	10049,04	91,51664	0,00371	37,277263	0,339484156
0,70942493	19215,57	77,13872	0,503284	9670,88377	38,82266283
0,53988804	15846,32	87,6038	0,291479	4618,87102	25,53467641
0,69512831	18856,44	71,26874	0,483203	9111,4953	34,43729516
			1,281676	23438,5274	99,13411855
$v_2 = \frac{\sum_{s=1}^4 (\mu_{s1})^2 x_{s1}}{\sum_{s=1}^4 (\mu_{s1})^2}$				18287,4083	77,3472699

Sehingga pusat *cluster* yang terbentuk adalah

$$V_2 = \begin{bmatrix} 12267,7098 & 88,42900754 \\ 18287,4083 & 77,3472699 \end{bmatrix}.$$

Fungsi objektif yang dihasilkan adalah

$$J_2 = \sum_{i=1}^2 \sum_{j=2}^4 \sum_{k=1}^2 (x_{si} - V_{ji})^2 (x_{sj})^2 = 674195530,7.$$

Tabel 2.8 Perhitungan Fungsi Objektif pada Iterasi ke-2

μ_{s1}^2	μ_{s2}^2	$\sum_{i=1}^2 (x_{s1} - V_{1i})^2$	$\sum_{i=1}^2 (x_{s2} - V_{2i})^2$	$\mu_{s1}^2 \cdot d_{s1}$	$\mu_{s2}^2 \cdot d_{s2}$	$L_1 + L_2$
		d_{s1}	d_{s2}	L_1	L_2	L_T
0,88189	0,00371	59937653,4	330575025,	528588	1226433,3	54085330,
8		2	97	44	02	

μ_{s1}^2	μ_{s2}^2	$\sum_{i=1}^2 (x_{s1}-V_{1i})^2$	$\sum_{i=1}^2 (x_{s2}-V_{2i})^2$	$\mu_{s1}^2 \cdot d_{s1}$	$\mu_{s2}^2 \cdot d_{s2}$	$L_1 + L_2$
		d_{s1}	d_{s2}	L_1	L_2	L_T
0,08443	0,50328	679905687	331621513,	574071	166899801	224306958
4	4		5	57	,8	,6
0,21170	0,29147	373900278	330859655,	791558	96438641,	175594452
3	9		2	11	43	,1
0,09294	0,48320	642964324	332049231,	597616	160447185	220208790
7	3		9	05		
J_2						674195530 ,7

Karena $|J_2 - J_1| = |674195530,7 - 165955762,9| = 508239767,8 > \varepsilon = 10^{-5}$ dan $t = 1 < MaxIter = 100$, maka proses dilanjutkan ke Iterasi ke-2 dengan menghitung perubahan matriks partisi,

Tabel 2.9 Perhitungan Matriks Partisi pada Iterasi ke-2

d_{s1}^{-1}	d_{s2}^{-1}	Total	$\frac{M1}{MT}$	$\frac{M2}{MT}$
$M1$	$M2$	MT	μ_{s1}	μ_{s2}
1,6684E-08	3,02503E-09	1,9709E-08	0,846515	0,15348453
1,4708E-09	3,01549E-09	4,48628E-09	0,327842	0,67215759
2,6745E-09	3,02243E-09	5,69694E-09	0,469464	0,53053566
1,5553E-09	3,0116E-09	4,5669E-09	0,340559	0,65944142

diperoleh matriks partisi yang baru yaitu

$$U_2 = \begin{bmatrix} 0,846515 & 0,15348453 \\ 0,327842 & 0,67215759 \\ 0,469464 & 0,53053566 \\ 0,340559 & 0,65944142 \end{bmatrix}.$$

demikian seterusnya sampai terpenuhi kondisi $|J_{(t+1)} - J_{(t)}| < \varepsilon$ atau $t > MaxIter$. Proses pengerjaannya berhenti pada Iterasi ke-15 ($t = 15$) karena terpenuhinya salah satu syarat yaitu:

$$|J_{15} - J_{14}| > |6186298,068968 - 6186298,068968| = 0 < \varepsilon = 10^{-5}.$$

Pusat *cluster* yang terbentuk adalah

$$V_2 = \begin{bmatrix} 10167,1257506830 & 91,4349714191081 \\ 18165,9679710536 & 77,8398321036266 \end{bmatrix}.$$

- a. *Cluster* yang pertama terdiri dari objek yang memiliki rata-rata X_1 sebesar 10167,1257506830 dan rata-rata X_2 adalah 91,4349714191081.
- b. *Cluster* yang kedua terdiri dari objek yang memiliki rata-rata X_1 sebesar 18165,9679710536 dan rata-rata X_2 adalah 77,8398321036266.

Matriks partisi U pada Iterasi terakhir adalah:

$$U_{15} = \begin{bmatrix} 0,999788398533598 & 0,000211601466401688 \\ 0,0132768762980910 & 0,986723123701909 \\ 0,142978272548251 & 0,857021727451749 \\ 0,00627515293094026 & 0,993724847069060 \end{bmatrix}.$$

Didapatkan informasi mengenai kecenderungan setiap objek untuk masuk ke *cluster* tertentu. Derajat keanggotaan terbesar atau nilai maksimum pada suatu *cluster* menunjukkan bahwa objek itu cenderung menjadi anggota dari *cluster*

tersebut, data kecenderungan objek masuk ke *cluster* tertentu dapat dilihat pada

Tabel 2.10

Tabel 2.10 Derajat Keanggotaan Setiap Objek pada Iterasi Terakhir

Objek	Variabel		Derajat Keanggotaan		Kecenderungan Data Masuk pada <i>Cluster</i>	
	GLN (X_1)	LGRE (X_2)	U_1	U_2	K_1	K_2
1	10049,04	91,51664	0,9997	0,0002	*	
2	19215,57	77,13872	0,0132	0,9867		*
3	15846,32	87,6038	0,14297	0,8570		*
4	18856,44	71,26874	0,0062	0,9937		*

Hasil akhirnya adalah terbentuknya 2 buah *cluster*, dimana untuk *cluster* pertama beranggotakan objek 1 dan *cluster* kedua beranggotakan objek 2, 3, dan 4.

Matlab menyediakan fungsi khusus untuk melakukan *clustering* menggunakan metode FCM untuk memudahkan perhitungan. Berikut ini merupakan sintaks fungsi FCM pada Matlab:

$$[U, \text{Center}] = \text{fcm}(X, \text{JumlahCluster})$$

dengan,

Center = pusat *cluster* yang dihasilkan, tiap-tiap baris menunjukkan pusat suatu *cluster*.

X = matriks data yang akan di *cluster*.

Jumlah *Cluster* = jumlah *cluster* yang diinginkan (>1).

U = derajat keanggotaan masing-masing data pada setiap *cluster*.

D. Teori Himpunan *Fuzzy*

1. Himpunan *Fuzzy*

Himpunan *fuzzy* pada ruang semesta U didefinisikan sebagai suatu fungsi keanggotaan $\mu_A(x)$ yang memiliki nilai pada interval $[0, 1]$. Oleh karena itu, himpunan *fuzzy* merupakan bentuk umum dari himpunan klasik dengan nilai fungsi keanggotaan pada interval $[0, 1]$. Dengan kata lain, fungsi keanggotaan pada himpunan klasik memiliki dua nilai 0 dan 1. Sedangkan fungsi keanggotaan pada himpunan *fuzzy* adalah fungsi kontinu dengan *range* $[0, 1]$. Himpunan *fuzzy* A dalam U dapat direpresentasikan sebagai himpunan pasangan berurutan elemen x dan nilai keanggotaannya, yaitu:

$$A = \{(x, \mu_A(x)) | x \in U\} \quad (2.33)$$

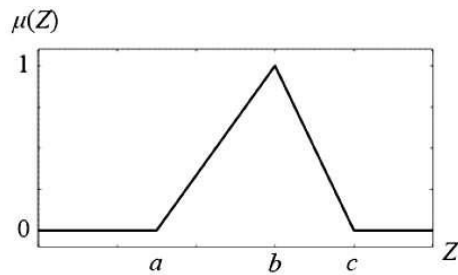
dengan $\mu_A(x)$ adalah derajat keanggotaan x di A (Wang, 1997: 21-22).

2. Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi (Kusumadewi & Purnomo, 2010:8). Ada beberapa fungsi yang dapat digunakan melalui pendekatan fungsi untuk

mendapatkan nilai keanggotaan, seperti *triangular*, *trapezoidal*, *gaussian*, dan *generalized bell* (Widodo & Handayanto, 2012:61).

a. Fungsi Keanggotaan *Triangular*

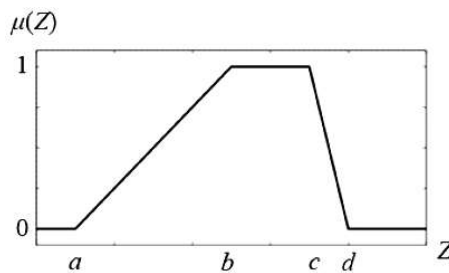


Gambar 2. 9 Kurva Fungsi Keanggotaan *Triangular*

Fungsi Keanggotaan *triangular* yang ditunjukkan pada Gambar 2.9 terbentuk oleh tiga parameter: a , b , dan c sebagai berikut

$$\mu(Z) = \begin{cases} 0 & , Z \leq a \text{ atau } Z \geq c \\ \frac{Z - a}{b - a} & , a \leq Z \leq b \\ \frac{c - Z}{c - b} & , b \leq Z \leq c \end{cases} \quad (2.34)$$

b. Fungsi Keanggotaan *Trapezoidal*

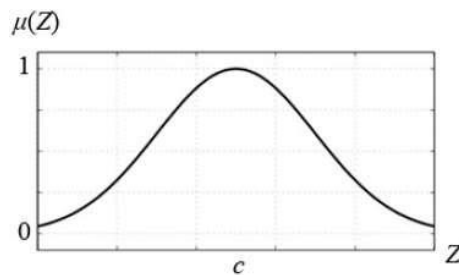


Gambar 2. 10 Kurva Fungsi Keanggotaan *Trapezoidal*

Fungsi keanggotaan *trapezoidal* yang ditunjukkan pada Gambar 2.10 terbentuk oleh empat parameter: a, b, c , dan d sebagai berikut:

$$\mu(Z) = \begin{cases} 0 & , Z \leq a \text{ atau } Z \geq c \\ \frac{Z - a}{b - a} & , a \leq Z \leq b \\ 1 & , b \leq Z \leq c \\ \frac{d - Z}{d - c} & , c \leq Z \leq d \end{cases} \quad (2.35)$$

c. Fungsi Keanggotaan *Gaussian*

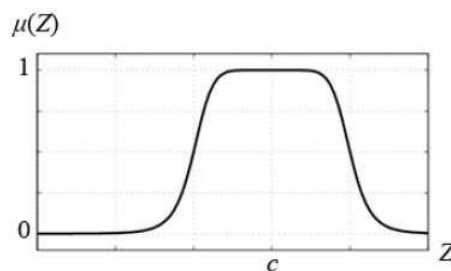


Gambar 2. 11 Kurva Fungsi Keanggotaan *Gaussian*

Fungsi Keanggotaan *gaussian* yang ditunjukkan pada Gambar 2.11 terbentuk oleh parameter: lebar kurva σ dan pusat dengan domain c , sebagai berikut.

$$\mu(Z) = e^{-\frac{1}{2}\left(\frac{Z-c}{\sigma}\right)^2}. \quad (2.36)$$

d. Fungsi Keanggotaan *Generalized Bell*



Gambar 2. 12 Kurva Fungsi Keanggotaan *Generalized Bell*

Fungsi keanggotaan *generalized bell* yang ditunjukkan pada Gambar 2.12 terbentuk oleh tiga parameter: rata-rata (c) dan deviasi standar (a), sebagai berikut:

$$\mu(Z) = \frac{1}{1 + \left| \frac{Z - c}{a} \right|^2}. \quad (2.37)$$

3. Operator Dasar Zadeh

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α -predikat. Menurut Zimmermann (1996: 16-17) terdapat 3 operator dasar yang diciptakan oleh Zadeh:

a. Operator *AND*

Operator *AND* berhubungan dengan operasi interseksi pada himpunan. Fungsi keanggotaan $\mu_C(x)$ dari interseksi $C = A \cap B$ didefinisikan sebagai berikut:

$$\mu_C(x) = \min\{\mu_A(x), \mu_B(x)\}, x \in X. \quad (2.38)$$

b. Operator *OR*

Operator *OR* merupakan operator yang berhubungan dengan operasi *union* pada himpunan. Fungsi keanggotaan $\mu_D(x)$ dari *union* $D = A \cup B$ didefinisikan sebagai berikut:

$$\mu_D(x) = \max\{\mu_A(x), \mu_B(x)\}, x \in X. \quad (2.39)$$

c. Operator *NOT*

Operator *NOT* merupakan operator yang berhubungan dengan operasi komplement pada himpunan. Fungsi keanggotaan dari komplement himpunan *fuzzy* A didefinisikan sebagai berikut:

$$\mu_{CA}(x) = 1 - \mu_A(x), x \in X. \quad (2.40)$$

4. Sistem *Fuzzy*

Sistem *fuzzy* terdiri dari empat tahapan dasar, yaitu menentukan *input* dan *output*, fuzzifikasi, pembentukan aturan, sistem inferensi *fuzzy*, dan defuzzifikasi.

Tahapan-tahapan model *fuzzy* tersebut dijelaskan sebagai berikut:

a. Fuzzifikasi

Fuzzifikasi adalah pemetaan dari himpunan tegas ke himpunan *fuzzy* dengan suatu fungsi keanggotaan. Kriteria yang harus dipenuhi pada proses fuzzifikasi adalah semua anggota pada himpunan tegas harus termuat dalam himpunan *fuzzy*. Melalui fungsi keanggotaan yang telah disusun maka nilai-nilai *input* yang ditentukan menjadi informasi *fuzzy* (Wang, 1997:105).

b. Menentukan Aturan *Fuzzy*

Aturan yang digunakan pada himpunan *fuzzy* dipresentasikan dalam bentuk *If-Then* atau Jika-Maka. Aturan *fuzzy If-Then* dapat direpresentasikan dalam bentuk berikut (Wang, 1997:105):

$$IF < \text{proposisi fuzzy} > THEN < \text{proposisi fuzzy} >.$$

Proposisi *fuzzy* dibedakan menjadi dua, yaitu proposisi *fuzzy atomic* dan proposisi *fuzzy compound*. Proposisi *fuzzy atomic* adalah pernyataan single dengan x sebagai variabel linguistik dan F adalah himpunan *fuzzy* dari x . Proposisi *fuzzy compound* adalah gabungan dari proposisi *fuzzy atomic* yang dihubungkan dengan operator “or”, “and” dan “not”.

c. Sistem Inferensi *Fuzzy*

Sistem Inferensi *Fuzzy* (*Fuzzy Inference System*) merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *If-Then*, dan penalaran *fuzzy*. Terdapat beberapa metode pada Sistem Inferensi *Fuzzy* (Kusumadewi & Purnomo, 2010:31):

1) Metode Tsukamoto

Metode Tsukamoto merupakan perluasan dari penalaran monoton, pada metode tsukamoto, setiap konsekuen pada aturan yang berbentuk *If-Then* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

2) Metode Mamdani

Metode Mamdani sering dikenal sebagai Metode Max-Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan empat tahapan:

a) Pembentukan Himpunan *fuzzy*

Pada metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

b) Aplikasi fungsi implikasi

Pada metode Mamdani, fungsi implikasi yang digunakan adalah Min.

c) Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada tiga metode yang dilakukan dalam melakukan inferensi sistem *fuzzy*, yaitu: max, additive dan probabilistik OR (probor).

d) Penegasan (*defuzzy*)

Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, antara lain metode *Centroid (Composite Moment)*, metode Bisektor, metode *Mean of Maximum (MOM)*, metode *Largest Of Maximum (LOM)*, dan Metode *Smallest Of Maximum (SOM)*.

3) Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK, metode TSK terdiri dari dua jenis, yaitu:

a) Model *Fuzzy* Sugeno Orde-Nol

Secara umum bentuk model *fuzzy* Sugeno Orde-Nol adalah

IF $(x_1 \text{ is } A_1) \text{ o } (x_2 \text{ is } A_2) \text{ o } (x_3 \text{ is } A_3) \text{ o } \dots \text{ o } (x_i \text{ is } A_i)$ THEN $z = k$

dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b) Model *Fuzzy* Sugeno Orde-Satu

Secara umum bentuk model *fuzzy* Sugeno orde-satu adalah:

IF $(x_1 \text{ is } A_1) \text{ o } (x_2 \text{ is } A_2) \text{ o } \dots \text{ o } (x_N \text{ is } A_N)$ THEN $z = p_1 * x_1 + \dots + p_i * x_i + q$

dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke- i dan q juga merupakan konstanta dalam konsekuen.

4) Defuzzifikasi

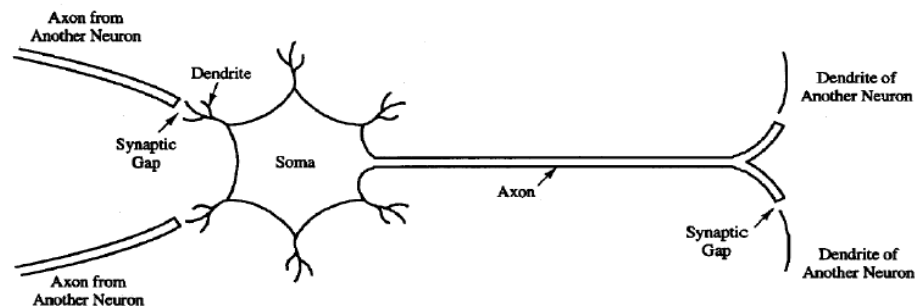
Defuzzifikasi merupakan proses terakhir dalam pemodelan *fuzzy*. Proses ini mengubah himpunan *fuzzy* ke dalam bilangan real. Nilai dari hasil defuzzifikasi adalah *output* dari model *fuzzy*.

E. Jaringan Syaraf Tiruan (*Neural Network*)

1. Jaringan Syaraf Biologi

Otak manusia memiliki struktur yang sangat kompleks dan memiliki kemampuan yang luar biasa. Otak terdiri dari neuron-neuron dan penghubung yang disebut sinapsis. Neuron bekerja berdasarkan impuls/sinyal yang diberikan pada neuron. Neuron meneruskannya pada neuron lain. Diperkirakan manusia memiliki 10^{12} neuron dan $6 \cdot 10^{18}$ sinapsis. Dengan jumlah yang begitu banyak, otak mampu mengenali pola, melakukan perhitungan, dan mengontrol organ-organ tubuh dengan kecepatan yang lebih tinggi dibandingkan komputer digital. Sebagai perbandingan, pengenalan wajah seseorang yang sedikit berubah (misal memakai

topi, memiliki jenggot tambahan, dll) akan lebih cepat dilakukan manusia dibandingkan komputer (Siang, 2009:1). Gambar 2.13 menunjukkan struktur jaringan syaraf secara biologi.



Gambar 2. 13 Jaringan Syaraf Secara Biologi (Siang, 2009:1)

2. Jaringan Syaraf Tiruan

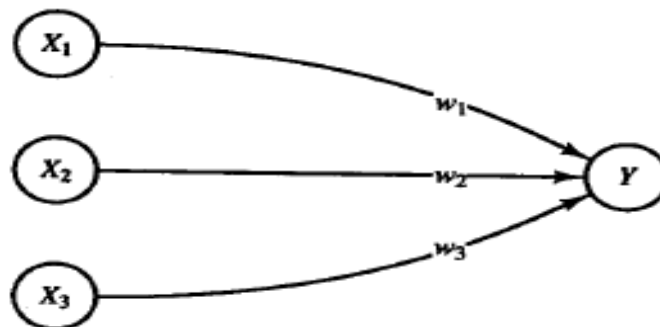
Jaringan syaraf tiruan merupakan sistem pemrosesan informasi yang memiliki karakteristik kinerja tertentu yang sama dengan jaringan syaraf biologis. Jaringan syaraf tiruan telah dikembangkan sebagai generalisasi model matematika dari jaringan syaraf manusia, dengan asumsi bahwa (Fausett, 1994:3):

- a. Pemrosesan informasi terjadi pada banyak elemen sederhana yang disebut dengan *neuron*.
- b. Sinyal dikirimkan diantara *neuron-neuron* melalui penghubung-penghubung.
- c. Tiap penghubung antar *neuron* memiliki bobot yang dapat memperkuat atau memperlemah sinyal yang dikirimkan.
- d. Tiap *neuron* menggunakan suatu fungsi aktivasi (biasanya fungsi nonlinear) yang dikenakan pada jumlahan *input* yang diterima untuk menentukan sinyal *output*.

Jaringan syaraf tiruan memiliki karakteristik antara lain (1) arsitektur yaitu pola dari hubungan antar *neuron*, (2) algoritma pembelajaran yaitu metode untuk menentukan bobot pada hubungan *neuron* dan (3) fungsi aktivasi (Fausett, 1994:3).

Pada Gambar 2.14 menunjukkan jaringan syaraf tiruan sederhana dimana sebuah *neuron* akan mengolah *input* X_1, X_2 , dan X_3 yang masing-masing memiliki bobot w_1, w_2 , dan w_3 dengan rumus (Fausett, 1994:3):

$$y_{in} = w_1x_1 + w_2x_2 + w_3x_3. \quad (2.41)$$



Gambar 2. 14 Jaringan Syaraf Tiruan Sederhana

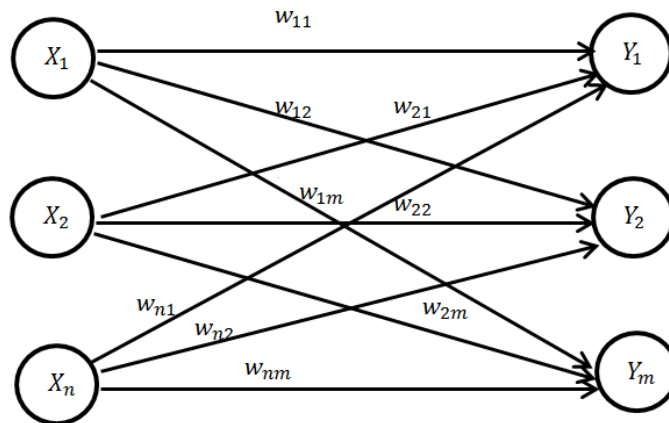
Jaringan syaraf tiruan ditentukan oleh tiga hal (Siang, 2009: 24-26):

- a. Pola hubungan antar neuron (Arsitektur Jaringan)

Beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan antara lain:

- 1) Jaringan Layar Tunggal (*single layer network*)

Dalam jaringan ini, sekumpulan input neuron dihubungkan langsung dengan sekumpulan *output*-nya. Dalam beberapa model (misal perceptron), hanya ada sebuah unit neuron *output*.

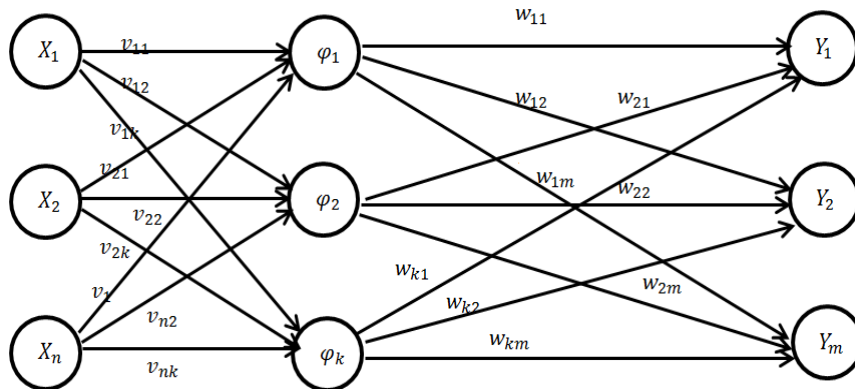


Gambar 2. 15 Arsitektur Jaringan Layer Tunggal

Gambar 2.15 menunjukkan arsitektur jaringan dengan n unit input (X_1, X_2, \dots, X_n) dan m buah unit *output* (Y_1, Y_2, \dots, Y_m). Dalam jaringan ini, semua unit *input* dihubungkan dengan semua unit *output*, meskipun dengan bobot yang berbeda-beda. Tidak ada unit *input* yang dihubungkan dengan unit *input* lainnya. Demikian pula dengan unit *output*. Besaran w_{ij} menyatakan bobot hubungan antara unit ke- i dalam *input* dengan unit ke- j dalam *output*. Bobot-bobot ini saling independen. Selama proses pelatihan, bobot-bobot tersebut akan dimodifikasi untuk meningkatkan keakuratan hasil. Model semacam ini tepat digunakan untuk pengenalan pola karena kesederhanaannya.

2) Jaringan Layer Jamak (*multi layer network*)

Jaringan layer jamak merupakan perluasan dari layer tunggal. Dalam jaringan ini, selain unit *input* dan *output*, ada unit-unit lain (sering disebut layer tersembunyi). Dimungkinkan pula ada beberapa layer tersembunyi. Sama seperti pada unit *input* dan *output*, unit-unit dalam satu layer tidak saling berhubungan.



Gambar 2. 16 Arsitektur Jaringan Layer Jamak

Gambar 2.16 adalah jaringan dengan n buah unit input (X_1, X_2, \dots, X_n), sebuah layer tersembunyi yang terdiri dari k buah unit $\varphi_1, \varphi_2, \dots, \varphi_k$ dan m unit buah *output* (Y_1, Y_2, \dots, Y_m). Jaringan layer jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layer tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.

3) Jaringan *Reccurent*

Model jaringan *reccurent* mirip dengan jaringan layer tunggal ataupun ganda. Hanya saja, ada *neuron output* yang memberikan sinyal pada unit *input* (sering disebut *feedback loop*).

b. Fungsi Aktivasi

Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu *neuron*. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya). Beberapa fungsi aktivasi yang sering dipakai adalah sebagai berikut:

1) Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1 & \text{jika } x \geq a \\ 0 & \text{jika } x < a \end{cases} \quad (2.42)$$

untuk beberapa kasus, fungsi *threshold* yang dibuat tidak berharga 0 atau 1, tapi berharga -1 atau 1 (sering disebut *threshold* bipolar).

Jadi

$$f(x) = \begin{cases} 1 & \text{jika } x \geq a \\ -1 & \text{jika } x < a \end{cases} \quad (2.43)$$

2) Fungsi Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.44)$$

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1 dan dapat diturunkan dengan mudah.

$$f'(x) = f(x)(1 - f(x)). \quad (2.45)$$

3) Fungsi Identitas

$$f(x) = x. \quad (2.46)$$

Fungsi Identitas sering dipakai apabila kita menginginkan keluaran jaringan berupa sembarang bilangan riil (bukan hanya pada range [0 1] atau [-1 1]).

c. Algoritma Pembelajaran (*Learning Algorithm*)

Salah satu bagian terpenting dari konsep jaringan syaraf adalah terjadinya proses pembelajaran. Tujuan utama dari proses pembelajaran adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan syaraf, sehingga diperoleh bobot akhir yang tepat sesuai dengan pola yang dilatih. Selama proses pembelajaran akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu. Nilai bobot

akan bertambah, jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, sebaliknya jika informasi tidak disampaikan oleh suatu *neuron* ke *neuron* yang lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi. Pada saat pembelajaran dilakukan pada *input* yang berbeda, maka nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Apabila nilai ini telah tercapai mengindikasikan bahwa tiap-tiap *input* telah berhubungan dengan *output* yang diharapkan. Pada dasarnya terdapat 2 metode pembelajaran pada jaringan syaraf tiruan yaitu (Kusumadewi & Hartati, 2010:84-85):

1) Pembelajaran Terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Pada proses pembelajaran, satu pola *input* akan diberikan ke satu neuron pada lapisan *input*. Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke *neuron* pada lapisan *output*. Lapisan *output* ini akan membangkitkan pola *output* yang nantinya akan dicocokkan dengan pola *output* targetnya. Apabila terjadi perbedaan antara pola *output* hasil pembelajaran dengan pola target, maka di sini akan muncul *error*. Apabila nilai *error* ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

2) Pembelajaran Tak Terawasi (*unsupervised learning*)

Pada metode pembelajaran yang tak terawasi tidak memerlukan target *output*. Pada metode ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *input* yang diberikan. Tujuan

pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokkan (klasifikasi) pola.

F. *Adaptive Neuro Fuzzy Inference System (ANFIS)*

1. Pengertian ANFIS

ANFIS adalah arsitektur yang secara fungsional sama dengan *fuzzy rule* base model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Bisa dikatakan bahwa ANFIS adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. Pada ANFIS juga memungkinkan aturan-aturan untuk beradaptasi (Kusumadewi & Hartati, 2010:377).

Agar jaringan dengan fungsi basis radial ekuivalen dengan fuzzy berbasis aturan model Sugeno orde 1, diperlukan batasan sebagai berikut (Kusumadewi & Hartati, 2010:378):

- a. Aturan-aturan harus memiliki metode agregasi yang sama (rata-rata terbobot atau penjumlahan terbobot) untuk menurunkan semua outputnya.
- b. Jumlah fungsi aktivasi harus sama dengan jumlah aturan fuzzy (IF-THEN).
- c. Jika ada beberapa input pada basis aturannya, maka tiap-tiap fungsi aktivasi harus sama dengan fungsi keanggotaan tiap-tiap inputnya.
- d. Fungsi aktivasi dan aturan-aturan fuzzy harus memiliki fungsi yang sama untuk neuron-neuron dan aturan-aturan yang ada di sisi outputnya.

2. Arsitektur ANFIS

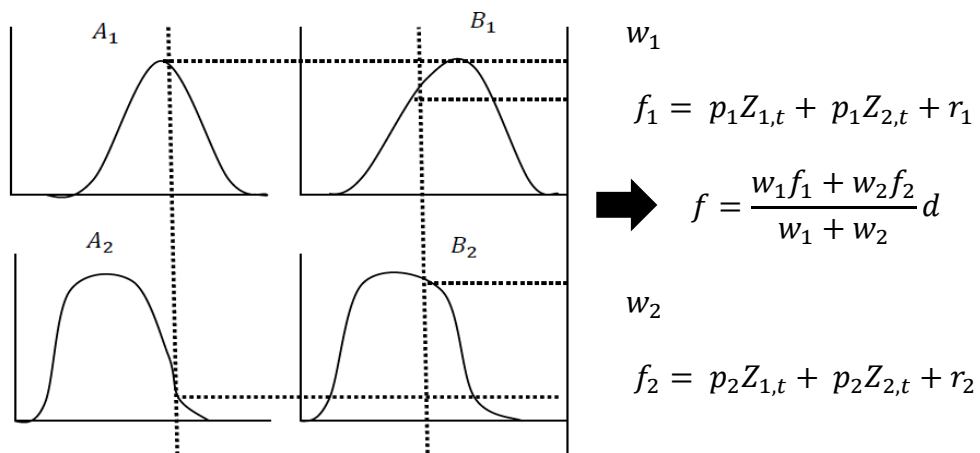
Misalkan ada 2 input x_1 , x_2 dan satu output y . Ada 2 aturan pada basis model Sugeno (Kusumadewi & Hartati, 2010:378):

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ Then } y_1 = c_{11}x_1 + c_{12}x_2 + c_{10}.$$

$$\text{If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ Then } y_2 = c_{21}x_1 + c_{22}x_2 + c_{20}.$$

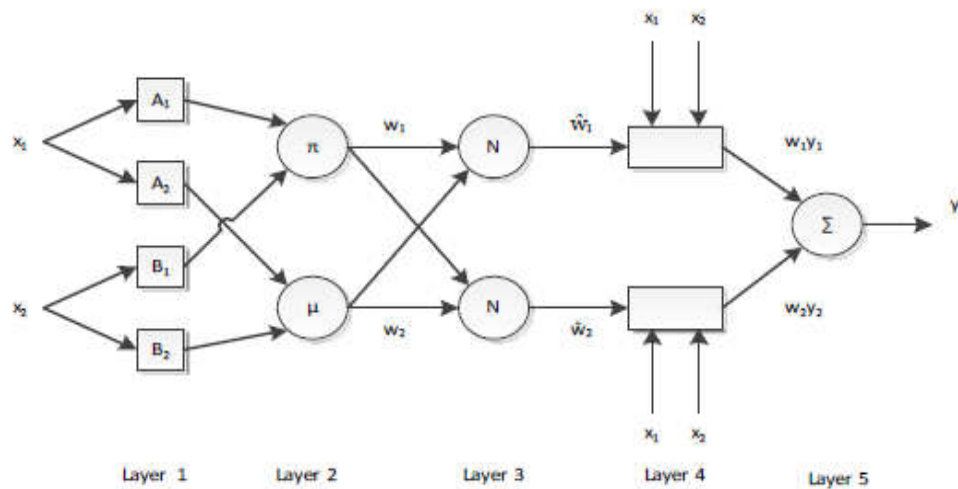
Jika a predikat untuk aturan kedua aturan adalah w_1 dan w_2 , maka dapat dihitung rata-rata terbobot:

$$y = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2} = \bar{w}_1 y_1 + \bar{w}_2 y_2. \quad (2.47)$$



Gambar 2. 17 Model Sugeno orde-1 dengan 2 input dan 2 rule

Adaptive Neuro Fuzzy Inference System (ANFIS) memiliki arsitektur yang dapat terlihat dalam struktur pada Gambar 2.18, dengan sistem inferensi *fuzzy* yang diterapkan adalah inferensi *fuzzy* model Takagi Sugeno Kang



Gambar 2. 18 Struktur ANFIS

Dari Gambar 2.19 dapat diketahui bahwa arsitektur ANFIS terdiri dari lima lapisan, dan setiap lapis terdapat node/simpul. Terdapat dua macam node yaitu node adaptif (bersimbol kotak) dan node tetap (bersimbol lingkaran). Fungsi dari setiap lapisan adalah sebagai berikut (Kusumadewi & Hartati, 2010:379):

- a. Tiap-tiap neuron i pada lapisan pertama adaptif terhadap parameter suatu fungsi aktivasi. Output dari tiap neuron berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan input, yaitu $\alpha_{A1}(x_1), \alpha_{B1}(x_1), \alpha_{A2}(x_1)$, atau $\alpha_{B2}(x_2)$. Sebagai contoh, misalkan fungsi keanggotaan diberikan sebagai:

$$\mu(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (2.48)$$

dimana $\{a,b,c\}$ adalah parameter-parameter, biasanya $b=1$. Jika nilai parameter-parameter ini berubah, maka bentuk kurva yang terjadi-pun akan ikut berubah.

Parameter-parameter pada lapisan itu biasanya dikenal dengan nama *premise parameters*.

- b. Tiap-tiap neuron pada lapisan kedua berupa neuron tetap yang outputnya adalah hasil dari masukan. Biasanya digunakan operator AND. Tiap-tiap node mempresentasikan α predikat dari aturan ke-i.
- c. Tiap-tiap neuron pada lapisan ketiga berupa node tetap yang merupakan hasil penghitungan rasio dari α predikat (w), dari aturan ke-i terhadap jumlah dari keseluruhan α predikat.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \text{ dengan } i=1,2. \quad (2.49)$$

hasil ini dikenal dengan nama *normalized firing strength*.

- d. Tiap-tiap neuron pada lapisan keempat merupakan node adaptif terhadap suatu output.

$$\bar{w}_i y_i = \bar{w}_i (c_{i1} x_1 + c_{i2} x_2 + c_{i0}); \text{ dengan } i=1,2. \quad (2.50)$$

dengan \bar{w}_i adalah *normalized firing strength* pada lapisan ketiga dan $\{c_{i1}, c_{i2}, c_{i0}\}$ adalah parameter-parameter pada *neuron* tersebut. Parameter-parameter pada lapisan tersebut disebut dengan nama *consequent parameters*.

- e. Tiap-tiap neuron pada lapisan kelima adalah node tetap yang merupakan jumlahan dari semua masukan.

3. Algoritma pembelajaran *Hybrid*

ANFIS dilatih dengan algoritma pelatihan *hybrid*. Ada dua langkah dalam pelatihan *hybrid* yaitu langkah maju (*forward*) dan langkah mundur (*backward*).

Pada langkah maju, parameter premis tetap, input jaringan akan merambat maju sampai pada lapisan ke empat, dimana parameter konsekuensi (p, q, r) akan diidentifikasi dengan menggunakan metode *least-square estimation* (LSE). Sedangkan pada langkah mundur error sinyal antara keluaran yang diinginkan dengan keluaran aktual, akan merambat mundur dan premis parameter (a, b, c) akan diperbaiki dengan metode *gradient-descent* (penurunan Gradien). Pada penelitian ini, langkah maju untuk menentukan parameter konsekuensi akan diidentifikasi dengan menggunakan metode dekomposisi nilai singular, sedangkan langkah mundur untuk menentukan model propagasi *error* menggunakan metode *gradient descent*.

a. Langkah Maju dengan Dekomposisi Nilai Singular

Suatu proses dekomposisi akan memfaktorkan sebuah matriks menjadi lebih dari satu matriks. Demikian halnya dengan dekomposisi nilai singular adalah salah satu teknik dekomposisi berkaitan dengan nilai singular suatu matriks yang merupakan salah satu karakteristik matriks tersebut (Ariyanti, 2010). Dekomposisi nilai singular berkaitan dengan nilai eigen dan nilai singular, definisi-definisi dan teorema yang berkaitan dengan dekomposisi nilai singular adalah sebagai berikut:

Definisi 2.1 (Johnson & Wichern, 2007:97). *Diberikan matriks persegi A dengan ukuran $k \times k$ dan I adalah matriks identitas dari matriks $k \times k$. Kemudian apabila skalar $\lambda_1, \lambda_2, \dots, \lambda_k$ memenuhi persamaan polinomial $|A - \lambda I| = 0$ maka skalar λ disebut nilai eigen dari matriks A . Persamaan $|A - \lambda I| = 0$ disebut persamaan karakteristik.*

Contoh 2.2

Diberikan matriks A sebagai berikut:

$$A = \begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix}$$

dengan menggunakan persamaan karakteristik pada Definisi 2.1 diperoleh:

$$|A - \lambda I| = \left| \begin{bmatrix} 9 & -2 \\ -2 & 6 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \left| \begin{bmatrix} 9 - \lambda & -2 \\ -2 & 6 - \lambda \end{bmatrix} \right| = 0$$

$$((9 - \lambda)(6 - \lambda)) - 4 = 0$$

$$\lambda^2 - 15\lambda + 50 = 0$$

$$(\lambda - 10)(\lambda - 5) = 0$$

$$\lambda = 10 \text{ atau } \lambda = 5$$

dengan demikian diperoleh nilai eigen dari matriks A yaitu $\lambda_1 = 10$ dan $\lambda_2 = 5$.

Definisi 2.2 (Johnson & Wichern, 2007:98). *Diberikan matriks persegi A dengan ukuran $k \times k$ dan misalkan λ adalah nilai eigen dari matriks A . Jika x adalah vektor tak nol ($x \neq 0$) sedemikian sehingga:*

$$Ax = \lambda x \tag{2.51}$$

kemudian x dinyatakan sebagai vektor eigen (vektor karakteristik) dari matriks A .

Contoh 2.3

Dari contoh 2.2, akan dicari vektor eigen dari matriks A dengan menggunakan persamaan 2.51

Untuk nilai eigen $\lambda_1 = 10$,

$$(A - \lambda I)x = \begin{bmatrix} 9 - \lambda & -2 \\ -2 & 6 - \lambda \end{bmatrix} x = \begin{bmatrix} -1 & -2 \\ -2 & -4 \end{bmatrix} x = 0$$

memiliki salah satu vektor eigen, yaitu $\begin{bmatrix} 2 \\ -1 \end{bmatrix} x_2$ dengan $x_2 \neq 0$.

Kemudian untuk nilai eigen $\lambda_2 = 5$,

$$(A - \lambda I)x = \begin{bmatrix} 9 - \lambda & -2 \\ -2 & 6 - \lambda \end{bmatrix} x = \begin{bmatrix} 4 & -2 \\ -2 & 1 \end{bmatrix} x = 0$$

memiliki salah satu vektor eigen, yaitu $\begin{bmatrix} 1 \\ 2 \end{bmatrix} x_1$ dengan $x_1 \neq 0$.

Definisi 2.3 (Anton, 1994:192). *Sebuah himpunan vektor pada ruang hasil kali dalam dinamakan himpunan ortogonal jika semua pasangan vektor-vektor yang berbeda dalam himpunan tersebut ortogonal. Sebuah himpunan ortogonal yang setiap vektornya mempunyai norma 1 dinamakan ortonormal.*

Teorema 2.1 (Anton, 1994:196). *Setiap ruang hasil kali dalam berdimensi berhingga tak nol mempunyai sebuah basis ortonormal.*

Jika diberikan himpunan ortogonal dari vektor tak nol $\{v_1, v_2, \dots, v_n\}$, maka himpunan tersebut dapat diubah menjadi himpunan ortonormal dengan

$$u_i = \left(\frac{1}{\|v_i\|} \right) v_i, \quad i = 1, 2, \dots, n. \quad (2.52)$$

Contoh 2.4

Diberikan himpunan ortogonal di R^3 , $\{v_1, v_2, v_3\} = \{(1,1,1), (0,1,1), (0,0,1)\}$, himpunan ortogonal tersebut akan ditransformasikan ke dalam basis ortonormal.

Penyelesaian:

Menggunakan persamaan 2.52, diperoleh himpunan ortonormal $\{u_1, u_2, u_3\}$ dengan

$$u_1 = \left(\frac{1}{\|v_1\|} \right) v_1 = \frac{1}{\sqrt{1^2 + 1^2 + 1^2}} (1,1,1) = \frac{1}{\sqrt{3}} (1,1,1).$$

$$u_2 = \left(\frac{1}{\|v_2\|} \right) v_2 = \frac{1}{\sqrt{0^2 + 1^2 + 1^2}} (0,1,1) = \frac{1}{\sqrt{2}} (0,1,1).$$

$$u_3 = \left(\frac{1}{\|v_3\|} \right) v_3 = \frac{1}{\sqrt{0^2 + 0^2 + 1^2}} (0,0,1) = (0,0,1).$$

Definisi 2.4 (Ariyanti, 2010). *Dekomposisi nilai singular matriks riil A dengan ukuran $m \times n$ adalah faktorisasi*

$$A = U \Sigma V^T \quad (2.53)$$

dengan U matriks ortogonal $m \times m$, V matriks orthogonal $n \times n$ dan Σ matriks $m \times n$ bernilai riil tak negatif yang disebut nilai-nilai singular. Dengan kata lain $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ terurut sehingga $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$.

Jika $U = (u_1, u_2, \dots, u_n)$ dan $V = (v_1, v_2, \dots, v_n)$ maka

$$A = \sum_{i=1}^n \sigma_i u_i v_i^T. \quad (2.54)$$

Definisi tersebut juga menyatakan bahwa matriks A $m \times n$ dapat dinyatakan sebagai dekomposisi matriks yaitu matriks U , Σ dan V . Matriks Σ merupakan matriks diagonal dengan elemen diagonalnya berupa nilai-nilai singular matriks A , sedangkan matriks U dan V merupakan matriks-matriks yang kolom-kolomnya

berupa vektor singular kiri dan vektor singular kanan dari matriks A untuk nilai singular yang bersesuaian.

Dekomposisi nilai singular meliputi langkah-langkah menentukan nilai eigen dan vektor eigen dari matriks AA^T dan $A^T A$. Vektor eigen dari $A^T A$ membentuk kolom V , sedangkan vektor eigen dari AA^T membentuk kolom U . Nilai singular dalam Σ adalah akar pangkat dua dari nilai-nilai eigen matriks AA^T atau $A^T A$. Nilai singular adalah elemen elemen diagonal dari Σ dan disusun dengan urutan menurun.

Contoh 2.5

Tentukan dekomposisi nilai singular dari matriks:

$$A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

Penyelesaian:

Untuk menentukan nilai singular maka dibentuk matriks

$$AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

dengan menggunakan persamaan 2.45 diperoleh

$$|AA^T - \lambda I|_x = \left| \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right|$$

$$= \begin{vmatrix} 11 - \lambda & 1 \\ 1 & 11 - \lambda \end{vmatrix} = 0$$

$$\lambda^2 - 22\lambda + 120 = 0$$

$$(\lambda - 10)(\lambda - 12) = 0$$

$$\lambda_1 = 10 \text{ dan } \lambda_2 = 12$$

Selanjutnya, menentukan nilai eigen dari AA^T , yaitu $\lambda = 10$ dan $\lambda = 12$, sehingga diperoleh nilai singular dari A yaitu $\sqrt{10}$ dan $\sqrt{12}$. Selanjutnya dalam menentukan vektor-vektor eigen dapat diperoleh dengan mensubstitusikan nilai-nilai eigen ke dalam persamaan $(AA^T - \lambda I)x$.

Untuk $\lambda = 10$, diperoleh:

$$(11 - 10)x_1 + x_2 = 0$$

$$x_1 = -x_2$$

maka vektor eigen $u_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ bersesuaian dengan nilai eigen $\lambda = 10$.

Untuk $\lambda = 12$, diperoleh

$$(11 - 12)x_1 + x_2 = 0$$

$$x_1 = x_2$$

maka vektor eigen $u_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ bersesuaian dengan nilai eigen $\lambda = 12$,

dengan menormalisasikan u_1 dan u_2 diperoleh

$$\bar{u}_1 = \frac{u_1}{|u_1|} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \text{ dan } \bar{u}_2 = \frac{u_2}{|u_2|} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

diperoleh $U = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$

selanjutnya, untuk menentukan matriks V , maka dibentuk matriks

$$A^T A = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}.$$

Menggunakan langkah yang sama, diperoleh nilai eigen dari $A^T A$ yaitu $\lambda_1 = 0, \lambda_2 = 10$ dan $\lambda_3 = 12$. Oleh karena itu, diperoleh nilai singular dari A yaitu $0, \sqrt{10}$ dan $\sqrt{12}$. Dengan mencari vektor eigen yang bersesuaian dengan nilai eigen diperoleh

$$u_3 = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix} \text{ bersesuaian dengan nilai eigen } \lambda = 0$$

$$u_2 = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} \text{ bersesuaian dengan nilai eigen } \lambda = 10$$

$$u_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \text{ bersesuaian dengan nilai eigen } \lambda = 12$$

akibatnya, vektor-vektor singular yang orthonormal adalah

$$\bar{v}_1 = \begin{bmatrix} 1/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}; \bar{v}_2 = \begin{bmatrix} 2/\sqrt{5} \\ -1/\sqrt{5} \\ 0 \end{bmatrix}; \text{ dan } \bar{v}_3 = \begin{bmatrix} 1/\sqrt{30} \\ 2/\sqrt{30} \\ -5/\sqrt{30} \end{bmatrix};$$

$$\text{Jadi, } V^T = \begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{30} & 2/\sqrt{30} & -5/\sqrt{30} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}.$$

Dari proses diatas, diperoleh dekomposisi nilai singular matriks tersebut adalah

$$\begin{aligned} A &= U \Sigma V^T \\ &= \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{30} & 2/\sqrt{30} & -5/\sqrt{30} \end{bmatrix} \\ &= \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}. \end{aligned}$$

Penelitian ini menggunakan model *adaptive neuro fuzzy inference system* (ANFIS) dengan model Takagi Sugeno Kang (TSK) order satu. Diberikan N data *training* dengan n input satu *output* data $(x_1^k, x_2^k, \dots, x_n^k; d^k)$ untuk $k = 1, 2, \dots, N$. Misalkan suatu aturan *fuzzy* ke- i untuk model TSK order satu ditulis:

$$R^i: x_1 \text{ adalah } A_{i1} \text{ dan } \dots \text{ dan } x_n \text{ adalah } A_{in}, \text{ maka } y_i = b_{i0} + b_{i1}x_1 + \dots + b_{in}x_n$$

dengan $i = 1, 2, \dots, L$ dan L adalah banyaknya aturan *fuzzy*, A_{ij} adalah himpunan *fuzzy* pada *input* ke- j , aturan ke- i , y_i adalah *output* aturan *fuzzy* ke- i , b_{ij} adalah parameter real yang akan dicari. *Output* model *fuzzy* dengan fuzzifier singleton, mesin inferensi pergandaan dan *defuzzifier* rata-rata pusat berbentuk:

$$y = \sum_{i=1}^L \bar{w}_i (b_{i1}x_1 + \dots + b_{in}x_n + b_{i0}) \quad (2.55)$$

dengan $\bar{w}_i = \frac{w_i}{w_1 + w_2 + \dots + w_i}$, dimana $i = 1, 2, \dots, n$.

Selanjutnya akan dibentuk model yang meminimumkan fungsi tujuan J :

$$J = \sum_{k=1}^N (d(k) - y(k))^2 \quad (2.56)$$

dengan $d(k)$ adalah *output* sebenarnya untuk pasangan data ke- k , dan $y(k)$ adalah *output* model TSK untuk pasangan data ke- k . Jadi $y(k)$ adalah nilai y pada Persamaan 2.49 untuk data input ke- k ($x_1^k, x_2^k, \dots, x_n^k$).

Persamaan 2.56 dapat ditulis kembali dalam bentuk: (Yen, *et al.*, 1998)

$$J = \sum_{k=1}^N (d(k) - y(k))^2 = (d - Xb)^T (d - Xb) \quad (2.57)$$

dengan

$$d = \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}$$

$$X = \begin{bmatrix} \bar{w}_1(1)x_1(1) & \dots & \bar{w}_1(1)x_n(1) & \bar{w}_1(1) & \dots & \bar{w}_L(1)x_n(1) & \dots & \bar{w}_L(1) \\ \bar{w}_1(2)x_1(2) & \dots & \bar{w}_1(2)x_n(2) & \bar{w}_1(2) & \dots & \bar{w}_L(2)x_n(2) & \dots & \bar{w}_L(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{w}_1(N)x_1(N) & \dots & \bar{w}_1(N)x_n(N) & \bar{w}_1(N) & \dots & \bar{w}_L(N)x_n(N) & \dots & \bar{w}_L(N) \end{bmatrix}$$

Jadi X adalah matriks ukuran $N \times [(n + 1) \times L]$, dan

$$b = [b_{10} \ b_{11} \ \dots \ b_{1n} \ \dots \ b_{L0} \ b_{L1} \ \dots \ b_{Ln}]^t \quad (2.58)$$

Suatu matriks ukuran $[(n + 1) \times L] \times 1$

Selanjutnya fungsi J pada Persamaan 2.57 akan mencapai minimum jika $d - Xb = 0$ sehingga diperoleh parameter-parameter b_{ij} yang merupakan entri-entri matriks b diestimasi dengan entri-entri matriks \hat{b} berikut:

$$\hat{b} = \sum_{i=1}^r \sigma_i^{-1} \langle d, u_i \rangle v_i = \sum_{i=1}^r \frac{u_i^T d}{\sigma_i} v_i \quad (2.59)$$

dengan r adalah banyaknya nilai singular tak nol.

b. Langkah Mundur dengan Model Propagasi *Error*

Model propagasi *error* digunakan untuk melakukan perbaikan terhadap parameter premis (a dan c). Konsep yang digunakan adalah *gradient descent*. Apabila diketahui jaringan adaptif ε_{ij} menyatakan error pada neuron ke- j pada lapisan ke- i maka perhitungan *error* pada tiap *neuron* pada tiap lapisan dirumuskan sebagai berikut (Kusumadewi & Hartati, 2006:53):

1) *Error* pada Lapisan 5

Pada lapisan 5 terdapat satu buah *neuron*. Propagasi *error* yang menuju lapisan ini dirumuskan sebagai berikut:

$$\varepsilon_{51} = \frac{\partial E_{Z_t}}{\partial f} = -2 (Z_t - f) \quad (2.60)$$

dengan

$Z_t = \text{output target}$.

$f = \text{hasil output jaringan}$.

$E_{Z_t} = \text{jumlah kuadrat error (SSE) pada lapisan kelima}$.

2) *Error* Lapisan 4

Pada lapisan 4 terdapat sebanyak dua buah *neuron*. Propagasi *error* yang menuju lapisan ini dapat dirumuskan sebagai berikut:

$$\varepsilon_{4j} = \left(\frac{\partial E_{z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{4j}} \right) \quad (2.61)$$

dengan ε_{4j} adalah *error* pada *neuron* ke- j ($j=1,2$), f_{4j} adalah *output neuron* lapisan 4 ke- j . Karena $f = \bar{w}_1 f_1 + \bar{w}_2 f_2$, maka:

$$\frac{\partial f}{\partial f_{41}} = \frac{\partial f}{\partial (\bar{w}_1 f_1)} = 1$$

$$\frac{\partial f}{\partial f_{42}} = \frac{\partial f}{\partial (\bar{w}_2 f_2)} = 1$$

sehingga

$$\varepsilon_{41} = \left(\frac{\partial E_{z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) = \varepsilon_{51}(1) = \varepsilon_{51} \quad (2.62)$$

$$\varepsilon_{42} = \left(\frac{\partial E_{z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{42}} \right) = \varepsilon_{51}(1) = \varepsilon_{51} \quad (2.63)$$

3) *Error* pada Lapisan 3

Pada lapisan 3 ini, terdapat sebanyak dua buah *neuron*. Propagasi *error* yang menuju lapisan ini dapat dirumuskan sebagai berikut:

$$\varepsilon_{3j} = \left(\frac{\partial E_{z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{4j}} \right) \left(\frac{\partial f_{4j}}{\partial f_{3j}} \right) \quad (2.64)$$

dengan ε_{3j} adalah *error* pada *neuron* ke- j ($j=1,2$), f_{3j} adalah *output neuron* lapisan 3 ke- j . Karena $f_{31} = \bar{w}_1$ dan $f_{32} = \bar{w}_2$, maka:

$$\frac{\partial f}{\partial f_{31}} = \frac{\partial (\bar{w}_1 f_1)}{\partial (\bar{w}_1)} = f_1$$

$$\frac{\partial f}{\partial f_{32}} = \frac{\partial (\bar{w}_2 f_2)}{\partial (\bar{w}_2)} = f_2$$

sehingga

$$\varepsilon_{31} = \left(\frac{\partial E_{Z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) \left(\frac{\partial f_{41}}{\partial f_{31}} \right) = \varepsilon_{51} f_1. \quad (2.65)$$

$$\varepsilon_{32} = \left(\frac{\partial E_{Z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) \left(\frac{\partial f_{41}}{\partial f_{32}} \right) = \varepsilon_{51} f_2. \quad (2.66)$$

4) Error Lapisan 2

Pada lapisan 2 terdapat sebanyak dua buah *neuron*. Propagasi *error* pada lapisan ke 2 dirumuskan sebagai berikut:

$$\varepsilon_{21} = \left(\frac{\partial E_{Z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) \left(\frac{\partial f_{41}}{\partial f_{31}} \right) \left(\frac{\partial f_{31}}{\partial f_{21}} \right) + \left(\frac{\partial E_{Z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) \left(\frac{\partial f_{41}}{\partial f_{32}} \right) \left(\frac{\partial f_{32}}{\partial f_{21}} \right) \quad (2.67)$$

$$\varepsilon_{22} = \left(\frac{\partial E_{Z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) \left(\frac{\partial f_{41}}{\partial f_{32}} \right) \left(\frac{\partial f_{32}}{\partial f_{22}} \right) + \left(\frac{\partial E_{Z_t}}{\partial f} \right) \left(\frac{\partial f}{\partial f_{41}} \right) \left(\frac{\partial f_{41}}{\partial f_{31}} \right) \left(\frac{\partial f_{31}}{\partial f_{22}} \right) \quad (2.68)$$

dengan f_{21} adalah *output neuron* ke-1 dan f_{22} adalah *output neuron* ke-2 pada lapisan 2. Karena $f_{21} = w_1$ dan $f_{22} = w_2$, maka:

$$\frac{\partial f_{31}}{\partial f_{21}} = \frac{\partial \left(\frac{w_1}{w_1 + w_2} \right)}{\partial w_1} = \frac{w_2}{(w_1 + w_2)^2} \quad (2.69)$$

$$\frac{\partial f_{32}}{\partial f_{21}} = \frac{\partial \left(\frac{w_2}{w_1 + w_2} \right)}{\partial w_1} = -\frac{w_2}{(w_1 + w_2)^2} \quad (2.70)$$

$$\frac{\partial f_{31}}{\partial f_{22}} = \frac{\partial \left(\frac{w_1}{w_1 + w_2} \right)}{\partial w_2} = \frac{w_1}{(w_1 + w_2)^2} \quad (2.71)$$

$$\frac{\partial f_{32}}{\partial f_{22}} = \frac{\partial \left(\frac{w_2}{w_1 + w_2} \right)}{\partial w_2} = -\frac{w_1}{(w_1 + w_2)^2} \quad (2.72)$$

sehingga

$$\varepsilon_{21} = \varepsilon_{31} \left(\frac{w_2}{(w_1 + w_2)^2} \right) + \varepsilon_{32} \left(-\frac{w_2}{(w_1 + w_2)^2} \right) = \frac{w_2}{(w_1 + w_2)^2} (\varepsilon_{31} - \varepsilon_{32}). \quad (2.73)$$

$$\varepsilon_{22} = \varepsilon_{32} \left(\frac{w_1}{(w_1 + w_2)^2} \right) + \varepsilon_{31} \left(-\frac{w_1}{(w_1 + w_2)^2} \right) = \frac{w_1}{(w_1 + w_2)^2} (\varepsilon_{32} - \varepsilon_{31}). \quad (2.74)$$

5) Error pada Lapisan 1

Pada lapisan 1 terdapat sebanyak empat buah *neuron*. Propagasi *error* yang menuju lapisan ini dapat dirumuskan sebagai berikut:

$$\begin{aligned} \varepsilon_{11} &= \varepsilon_{21} \left(\frac{\partial f_{21}}{\partial f_{11}} \right); \varepsilon_{12} = \varepsilon_{22} \left(\frac{\partial f_{22}}{\partial f_{12}} \right); \varepsilon_{13} = \varepsilon_{21} \left(\frac{\partial f_{21}}{\partial f_{13}} \right); \\ \varepsilon_{14} &= \varepsilon_{22} \left(\frac{\partial f_{22}}{\partial f_{14}} \right) \end{aligned} \quad (2.75)$$

karena

$f_{21} = w_1 = \mu_{A1}(Z) \cdot \mu_{B1}(Z)$; $f_{22} = w_2 = \mu_{A2}(Z) \cdot \mu_{B2}(Z)$; $f_{11} = A_1$; $f_{12} = A_2$; $f_{13} = B_1$, dan $f_{14} = B_2$, maka:

$$\varepsilon_{11} = \varepsilon_{21} \left(\frac{\partial(\mu_{A1}(Z) \cdot \mu_{B1}(Z))}{\partial(\mu_{A1}(Z))} \right) = \varepsilon_{21} \cdot \mu_{B1}(Z). \quad (2.76)$$

$$\varepsilon_{12} = \varepsilon_{22} \left(\frac{\partial(\mu_{A2}(Z) \cdot \mu_{B2}(Z))}{\partial(\mu_{A2}(Z))} \right) = \varepsilon_{22} \cdot \mu_{B2}(Z). \quad (2.77)$$

$$\varepsilon_{13} = \varepsilon_{21} \left(\frac{\partial(\mu_{A1}(Z) \cdot \mu_{B1}(Z))}{\partial(\mu_{B1}(Z))} \right) = \varepsilon_{21} \cdot \mu_{A1}(Z). \quad (2.78)$$

$$\varepsilon_{14} = \varepsilon_{22} \left(\frac{\partial(\mu_{A2}(Z) \cdot \mu_{B2}(Z))}{\partial(\mu_{B2}(Z))} \right) = \varepsilon_{22} \cdot \mu_{A2}(Z). \quad (2.79)$$

Error tersebut digunakan untuk mencari informasi *error* terhadap parameter a (a_{11} dan a_{12} untuk A_1 dan A_2 , b_{11} dan b_{12} untuk B_1 dan B_2) serta c (c_{11} dan c_{12} untuk A_1 dan A_2 , c_{11} dan c_{12} untuk B_1 dan B_2) sebagai berikut:

$$\varepsilon_{a_{11}} = \varepsilon_{11} \left(\frac{\partial f_{11}}{\partial a_{11}} \right) + \varepsilon_{12} \left(\frac{\partial f_{12}}{\partial a_{11}} \right) \quad (2.80)$$

$$\varepsilon_{a_{12}} = \varepsilon_{11} \left(\frac{\partial f_{11}}{\partial a_{12}} \right) + \varepsilon_{12} \left(\frac{\partial f_{12}}{\partial a_{12}} \right) \quad (2.81)$$

$$\varepsilon_{a_{21}} = \varepsilon_{13} \left(\frac{\partial f_{13}}{\partial a_{21}} \right) + \varepsilon_{14} \left(\frac{\partial f_{14}}{\partial a_{21}} \right) \quad (2.82)$$

$$\varepsilon_{a_{22}} = \varepsilon_{13} \left(\frac{\partial f_{13}}{\partial a_{22}} \right) + \varepsilon_{14} \left(\frac{\partial f_{14}}{\partial a_{22}} \right) \quad (2.83)$$

karena fungsi keanggotaan yang digunakan adalah *generalized bell*:

$$\mu(Z) = \frac{1}{1 + \left| \frac{Z - c}{a} \right|^2} \quad (2.84)$$

maka

$$\frac{\partial f}{\partial a} = \frac{2(Z - c)^2}{a^3 \left(1 + \left(\frac{Z - c}{a} \right)^2 \right)^2} \quad (2.85)$$

dan

$$\frac{\partial f}{\partial c} = \frac{2(Z - c)}{a^2 \left(1 + \left(\frac{Z - c}{a} \right)^2 \right)^2} \quad (2.86)$$

serta

$$\frac{\partial f_{12}}{\partial a_{11}} = 0; \frac{\partial f_{12}}{\partial a_{12}} = 0; \frac{\partial f_{14}}{\partial a_{21}} = 0; \frac{\partial f_{14}}{\partial a_{22}} = 0 \quad (2.87)$$

sehingga

$$\varepsilon_{a_{11}} = \varepsilon_{11} \left(\frac{2(Z - c_{11})^2}{a_{11}^3 \left(1 + \left(\frac{Z - c_{11}}{a_{11}} \right)^2 \right)^2} \right) \quad (2.88)$$

$$\varepsilon_{a_{12}} = \varepsilon_{12} \left(\frac{2(Z - c_{12})^2}{a_{12}^3 \left(1 + \left(\frac{Z - c_{12}}{a_{12}} \right)^2 \right)^2} \right) \quad (2.89)$$

$$\varepsilon_{a_{21}} = \varepsilon_{13} \left(\frac{2(Z - c_{21})^2}{a_{21}^3 \left(1 + \left(\frac{Z - c_{21}}{a_{21}} \right)^2 \right)^2} \right) \quad (2.90)$$

$$\varepsilon_{a_{22}} = \varepsilon_{14} \left(\frac{2(Z - c_{22})^2}{a_{22}^3 \left(1 + \left(\frac{Z - c_{22}}{a_{22}} \right)^2 \right)^2} \right) \quad (2.91)$$

dan

$$\varepsilon_{c_{11}} = \varepsilon_{11} \left(\frac{2(Z - c_{11})}{a_{11}^2 \left(1 + \left(\frac{Z - c_{11}}{a_{11}} \right)^2 \right)^2} \right) \quad (2.92)$$

$$\varepsilon_{c_{12}} = \varepsilon_{12} \left(\frac{2(Z - c_{12})}{a_{12}^2 \left(1 + \left(\frac{Z - c_{12}}{a_{12}} \right)^2 \right)^2} \right) \quad (2.93)$$

$$\varepsilon_{c_{21}} = \varepsilon_{13} \left(\frac{2(Z - c_{21})}{a_{21}^2 \left(1 + \left(\frac{Z - c_{21}}{a_{21}} \right)^2 \right)^2} \right) \quad (2.94)$$

$$\varepsilon_{c_{22}} = \varepsilon_{14} \left(\frac{2(Z - c_{22})}{a_{11}^2 \left(1 + \left(\frac{Z - c_{22}}{a_{22}} \right)^2 \right)^2} \right). \quad (2.95)$$

Kemudian ditentukan perubahan nilai parameter a_{ij} dan c_{ij} (Δa_{ij} dan Δc_{ij}), yang dirumuskan sebagai berikut:

$$\Delta a_{ij} = \eta \varepsilon_{a_{ij}} x_i \quad (2.96)$$

$$\Delta c_{ij} = \eta \varepsilon_{c_{ij}} x_i \quad (2.97)$$

dengan η adalah laju pembelajaran yang terletak pada interval $[0,1]$. Sehingga nilai a_{ij} dan c_{ij} yang baru adalah:

$$a_{ij} = a_{ij}(\text{lama}) + \Delta a_{ij}. \quad (2.98)$$

$$c_{ij} = c_{ij}(\text{lama}) + \Delta c_{ij}. \quad (2.99)$$

Contoh 2.6

Misal diberikan pasangan *input-output* dengan 2 variabel *input* dan 1 variabel *output* pada Tabel 2.12

Tabel 2.11 Data Ekstraksi Fitur GLRLM

Data ke- <i>i</i>	<i>Input</i>		<i>Target</i>
	GLN (x_1)	LGRE (x_2)	<i>Output</i>
1	10049,03963	91,51664205	0
2	8544,45554	85,42597005	1
3	12919,21188	78,04093595	2
4	12546,11915	89,05372733	3
5	11330,37219	88,6840316	4

Misalkan ada 2 aturan pada basis aturan model Sugeno:

Jika x_1 adalah A_1 dan x_2 adalah B_1 , maka $y_1 = c_{11}x_1 + c_{12}x_2 + c_{10}$.

Jika x_1 adalah A_2 dan x_2 adalah B_2 , maka $y_2 = c_{21}x_1 + c_{22}x_2 + c_{20}$.

Karena ada 2 aturan yang akan dibentuk, maka 5 data *input* dibagi menjadi 2 kelompok menggunakan algoritma FCM untuk mencari parameter *mean* (c) dan deviasi standar (a). Tabel 2.13 Menunjukkan kecenderungan suatu data masuk ke suatu *cluster*.

Tabel 2.12 Kecenderungan Masuk *Cluster*

Data ke-	Matriks Partisi (U)		Maksimal m_i^s	<i>Cluster</i> ke-
	m_i^1	m_i^2		
1	0,978372546	0,021627454	0,978372546	1
2	0,997442471	0,002557529	0,997442471	1
3	0,787341874	0,212658126	0,787341874	1
4	0,091952924	0,908047076	0,908047076	2
5	0,037944523	0,962055477	0,962055477	2

1. Lapisan Pertama

Setelah diperoleh kecenderungan data masuk suatu *cluster*, maka diperoleh nilai *mean* dan deviasi standar sebagai berikut:

$$c = \begin{bmatrix} 12265,23 & 85,25956 \\ 9296,748 & 88,47131 \end{bmatrix}.$$

$$a = \begin{bmatrix} 830,8279 & 6,254248 \\ 1063,902 & 4,306755 \end{bmatrix}.$$

Kemudian hasil *output* lapisan pertama berupa derajat keanggotaan dengan menggunakan fungsi keanggotaan *generalized bell* ditunjukkan pada Tabel 2.14

Tabel 2.13 *Output* Lapisan Pertama

Data ke-	μ_{A1}	μ_{A2}	μ_{B1}	μ_{B2}
1	0,1232	0,6667	0,4998	0,6667
2	0,0475	0,6667	0,9993	0,6667
3	0,6174	0,0794	0,4288	0,1457

Data ke-	μ_{A1}	μ_{A2}	μ_{B1}	μ_{B2}
4	0,8974	0,0968	0,731	0,982
5	0,4413	0,2149	0,7693	0,9976

2. Lapisan Kedua

Tiap-tiap neuron pada lapisan kedua berupa neuron tetap yang outputnya adalah hasil dari masukan. Biasanya digunakan operator AND. Tiap-tiap node merepresentasikan α predikat dari aturan ke-i. *Output* dari lapisan kedua berupa hasil perkalian dari derajat keanggotaan pada layer pertama.

$$w_i = \mu_{A_i} \times \mu_{B_i} .$$

Berikut adalah *output* yang diperoleh pada lapisan ke dua:

Tabel 2.14 *Output* Lapisan Kedua

Data ke-	w_1	w_2
1	0,061575	0,444489
2	0,047467	0,444489
3	0,264741	0,011569
4	0,655999	0,095058
5	0,339492	0,214384

3. Lapisan Ketiga

Tiap neuron pada lapisan ketiga berupa node tetap yang merupakan hasil penghitungan rasio dari α predikat (w), dari aturan ke-i terhadap jumlah dari

keseluruhan α predikat. *Output* ini sering disebut dengan *normalized firing strength*.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \text{ dengan } i = 1, 2.$$

berikut ini adalah *normalized firing strength* dari data diatas adalah.

Tabel 2.15 *Output* Lapisan Ketiga

Data ke-	\bar{w}_1	\bar{w}_2
1	0,121701	0,878299
2	0,09648	0,90352
3	0,958141	0,041859
4	0,873405	0,126595
5	0,612974	0,387026

4. Lapisan Keempat

Tiap-tiap neuron pada lapisan keempat merupakan node adaptif terhadap suatu output. Pada lapisan ke empat terdapat parameter baru yang disebut *consequent parameters*, pada penelitian ini nilai parameter konsekuen diperoleh dengan menggunakan metode dekomposisi nilai singular. Pada lapisan ke empat, akan dicari nilai dari $\bar{w}_i x_{ij}$, dengan $i = 1, 2; j = 1, 2$; sebagai koefisien dari $c_{ij} = 1, 2$ dan $j = 0, 1, 2$; pada persamaan 2. Hasil dari lapisan keempat ditunjukkan pada Tabel 2.17.

Tabel 2.16 Koefisien Parameter

c_{11}	c_{12}	c_{10}	c_{21}	c_{22}	c_{20}
-0,00237	0,045174	28,98261	0,000748	-0,357078	23,87664031

Jaringan yang telah terbentuk akan di latih degan menggunakan algoritma pembelajaran *hybrid*, dengan *maksimum epoh* = 100, toleransi *error* = 10^{-6} , laju pembelajaran = 0,5, dan *momentum* = 0,5. Selanjutnya adalah melakukan perbaikan terhadap parameter *a* dan *c* dengan menggunakan model propagasi *error* dengan konsep *gradient descent*.

- a. Propagasi *error* pada lapisan ke lima berdasarkan persamaan 2.60

$$\varepsilon_{13} = -2(0 - 2,24E - 13) = 4,48E - 13.$$

- b. Propagasi *error* pada lapisan ke empat berdasarkan persamaan 2.62 dan 2.63

$$\varepsilon_{11} = \varepsilon_{13} = 4,48E - 13.$$

$$\varepsilon_{12} = \varepsilon_{13} = 4,48E - 13.$$

- c. Propagasi *error* pada lapisan ke tiga berdasarkan persamaan 2.65 dan 2.66

$$\varepsilon_9 = \varepsilon_{11}f_1 = 4,48E - 13 \times 10049,04 = 4,502E - 09.$$

$$\varepsilon_{10} = \varepsilon_{12}f_2 = 4,48E - 13 \times 91,51664 = 4,009E - 11.$$

- d. Propagasi *error* pada lapisan ke dua berdasarkan persamaan 2.73 dan 2.74

$$\varepsilon_7 = \frac{w_2}{(w_1 + w_2)^2} (\varepsilon_9 - \varepsilon_{10})$$

$$\varepsilon_7 = \frac{0,444}{(0,061 + 0,444)^2} ((4,502E - 09) - (4,009E - 11))$$

$$\varepsilon_7 = 7,75E - 09.$$

$$\varepsilon_8 = \frac{w_1}{(w_1 + w_2)^2} (\varepsilon_{10} - \varepsilon_9)$$

$$\varepsilon_8 = \frac{0,061}{(0,061 + 0,444)^2} ((4,009E - 11) - (4,502E - 09))$$

$$\varepsilon_8 = -1,07E - 09.$$

- e. Propagasi *error* pada lapisan pertama berdasarkan persamaan 2.76-2.79

$$\varepsilon_3 = \varepsilon_7 \cdot \mu_{B1}(X) = 7,75E - 09 \times 0,4998 = 3,88E - 09.$$

$$\varepsilon_4 = \varepsilon_8 \cdot \mu_{B2}(X) = -1,07E - 09 \times 0,6667 = -7,16E - 10.$$

$$\varepsilon_5 = \varepsilon_7 \cdot \mu_{A1}(X) = 7,75E - 09 \times 0,1232 = 9,55E - 10$$

$$\varepsilon_4 = \varepsilon_8 \cdot \mu_{A2}(X) = -1,07E - 09 \times 0,6667 = -7,16E - 10 .$$

- f. Perubahan parameter c dan a berdasarkan persamaan 2.96 dan 2.97

$$\Delta c = \begin{bmatrix} -1,9E - 13 & -2,8E - 17 \\ 2,82E - 13 & -1,3E - 17 \end{bmatrix}.$$

$$\Delta a = \begin{bmatrix} 5,04E - 13 & -4,5E - 17 \\ 2E - 13 & -3,1E - 17 \end{bmatrix}.$$

- g. Nilai parameter a dan c yang baru berdasarkan persamaan 2.98 dan 2.99

$$c = \begin{bmatrix} 12265,23 & 85,25956 \\ 9296,748 & 88,47131 \end{bmatrix}.$$

$$a = \begin{bmatrix} 830,8279 & 6,254248 \\ 1063,902 & 4,306755 \end{bmatrix}.$$

5. Lapisan Kelima

Tiap-tiap neuron pada lapisan kelima adalah node tetap yang merupakan jumlahan dari semua masukan. Pada lapisan ini hanya ada satu simpul *output* yang

merupakan keluaran atau hasil dari seluruh perhitungan yang sudah dilakukan diatas.

$$y = (\bar{w}_1 x_1) c_{11} + (\bar{w}_1 x_2) c_{12} + \bar{w}_1 c_{10} + (\bar{w}_2 x_1) c_{21} + (\bar{w}_2 x_2) c_{22} + \bar{w}_2 c_{20}$$

dengan \bar{w}_1 dan \bar{w}_2 adalah output lapisan ketiga yang ditunjukkan pada Tabel 2.16 dan $c_{11}, c_{12}, c_{21}, c_{22}$ adalah koefisien parameter yang ditunjukkan pada Tabel 2.17 menggunakan persamaan pada lapisan kelima, diperoleh hasil *output* jaringan sebagai berikut:

Tabel. 2.17 Output Jaringan ANFIS

Data ke-	Target Output	Output Jaringan
1	0	2,24E-13
2	1	1
3	2	2
4	3	3
5	4	4

G. Uji Ketepatan Diagnosis

Menurut Zhu, Zeng & Wang (2010) tingkat ketepatan diagnosis suatu penelitian dapat ditentukan dengan menghitung sensitivitas, spesifisitas, dan akurasi dari model yang telah dibuat sebelumnya. Ukuran hasil diagnosis untuk menghitung sensitivitas, spesifisitas, dan akurasi adalah sebagai berikut:

Tabel 2.18 Hasil Uji Diagnosa 2 Kategori (Zhu, Zeng & Wang, 2010:1)

Hasil Uji Diagnosa	Kondisi Penyakit sebagaimana yang ditetapkan oleh Standar Kebenaran		
	Positif	Negatif	Jumlah Baris
Positif	<i>TP</i>	<i>FP</i>	<i>TP + FP</i>
Negatif	<i>FN</i>	<i>TN</i>	<i>FN + TN</i>
Jumlah Kolom	<i>TP + FN</i>	<i>FP + TN</i>	$N = TP + FN + FP + FN$

dengan,

TP (True Positive) : subjek yang memiliki penyakit dan hasil diagnosa menyatakan subjek memiliki penyakit.

FP (False Positive) : subjek yang tidak memiliki penyakit namun hasil diagnosa menyatakan subjek memiliki penyakit.

TN (True Negative) : subjek yang tidak memiliki penyakit dan hasil diagnosa menyatakan subjek tidak memiliki penyakit.

FN (False Negative) : subjek yang memiliki penyakit namun hasil diagnosa menyatakan subjek tidak memiliki penyakit.

1. Sensitivitas

Sensitivitas mengacu pada kemampuan tes untuk mengidentifikasi pasien dengan penyakit secara tepat. Semakin tinggi nilai sensitivitas, semakin rendah tingkat kesalahan suatu tes diagnostik. Rumus sensitivitas adalah sebagai berikut:

$$\text{Sensitivitas} = \frac{TP}{TP + FN} \quad (2.100)$$

Misalnya, jika sensitivitas = 99%, artinya ketika dilakukan tes diagnostik pada pasien dengan penyakit tertentu, pasien ini berpeluang 99% teridentifikasi positif terjangkit penyakit tersebut.

2. Spesifisitas

Spesifisitas mengacu pada kemampuan tes untuk mengidentifikasi pasien tanpa penyakit secara tepat. Nilai Spesifisitas merupakan peluang tes diagnosa penyakit tertentu tanpa memberikan hasil *False Positive*. Rumus spesifisitas adalah sebagai berikut:

$$\text{Spesifisitas} = \frac{TN}{TN + FP} \quad (2.101)$$

Misalnya, jika spesifisitas suatu tes 99%, ini artinya ketika dilakukan tes diagnosa pada pasien tanpa penyakit tertentu, pasien ini berpeluang 99% teridentifikasi negatif terjangkit penyakit tersebut.

3. Akurasi

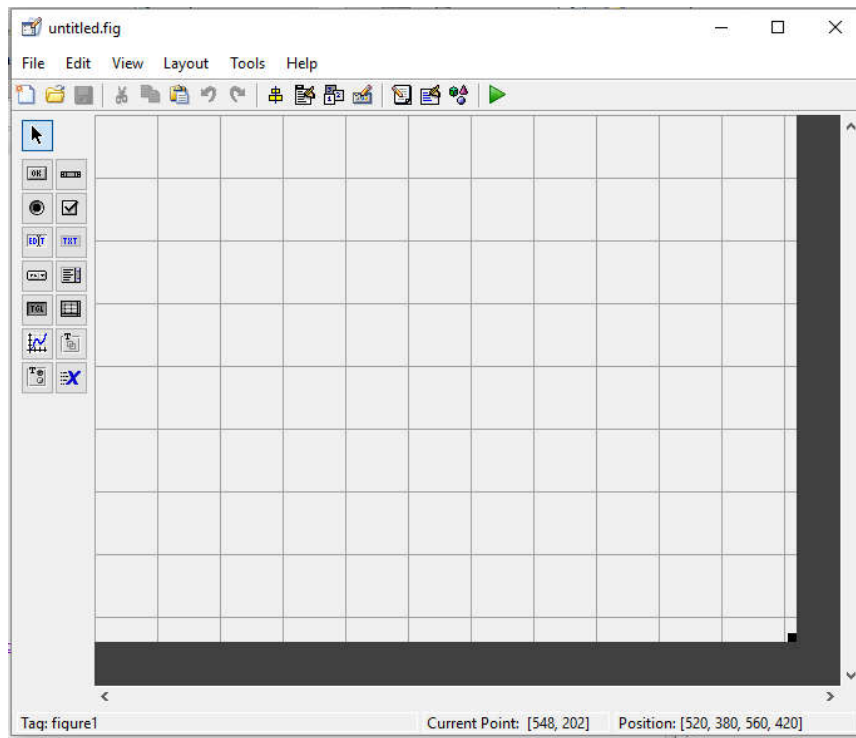
Akurasi adalah proporsi dari hasil yang benar (True), baik *True Positive* maupun *True Negative*, dalam suatu populasi. Akurasi mengukur atau mengidentifikasi dengan benar kondisi pasien. Rumus untuk menghitung akurasi adalah sebagai berikut:

$$Akurasi = \frac{TN + TP}{TN + TP + FN + FP} = \frac{Jumlah\ diagnosis\ benar}{Jumlah\ diagnosis\ keseluruhan} \quad (2.102)$$

Misalnya nilai keakurasian 99% dengan mengabaikan nilai positif atau negatif, namun perlu diperhatikan bahwa jika sensitivitas dan spesifitas adalah tinggi, maka nilai akurasi belum tentu tinggi juga.

H. Graphic User Interfaces (GUI)

Penggunaan *Graphic User Interfaces* (GUI) memberikan/ menyediakan fasilitas seperti *menu*, *pushbuttons*, *sliders*, dan sebagainya, sesuai dengan program yang diinginkan atau digunakan tanpa *knowledge* dari MATLAB. GUI juga memberikan cara untuk efisiennya manajemen data. GUI merupakan MATLAB script file yang dibuat untuk menunjukkan analisa suatu permasalahan khusus. Ada dua cara merancang GUI, yaitu dengan metode sederhana, dengan menggunakan *tool* khusus untuk merancang sesuai yang diinginkan. MATLAB menyediakan *tool*-nya dan dapat dilihat dengan mengetikkan '**guide**' pada MATLAB *command window*. Kemampuan maksimum dan kontrolnya adalah dengan pemrograman, namun demikian hasil yang diperoleh menggunakan perintah-perintah *basic user interface commands* (Arhami & Anita Desiani, 2005:173). Tampilan GUI pada Matlab ditunjukkan pada Gambar 2.20.



Gambar 2. 19 Tampilan Dasar GUI