

BAB II TINJAUAN PUSTAKA

2.1 Penyakit Jantung

Jantung atau *cardia* dalam bahasa Yunani atau *heart* dalam bahasa Inggris adalah organ berbentuk kerucut, berotot kuat dan berongga. Jantung terletak di dalam rongga dada (*cavum thorac*) dengan posisi agak ke bawah dan sedikit ke arah sebelah kiri. Puncak jantung (*apex cordis*) letaknya miring ke sebelah kiri. Pada dasarnya jantung adalah alat tubuh yang berfungsi sebagai pompa darah yang tidak akan berhenti selama hidup. Jantung terbentuk dari serabut-serabut otot bersifat khusus dan dilengkapi jaringan syaraf yang secara teratur dan otomatis memberikan rangsangan berdenyut bagi otot jantung. Denyutan ini menyebabkan jantung memompa darah yang kaya akan oksigen ke seluruh tubuh termasuk arteri koroner serta darah yang kurang oksigen ke paru-paru untuk mengambil oksigen (Soeharto, 2004).

Penyakit jantung adalah sebuah kondisi yang menyebabkan jantung tidak dapat melaksanakan tugasnya dengan baik. Penyakit yang mengenai jantung biasa disebut sebagai penyakit *kardiovaskular*. Masalah pada jantung dibagi menjadi dua bagian, yaitu penyakit jantung dan serangan jantung (*stroke*). Di seluruh dunia, jumlah penderita penyakit ini terus bertambah. Semua kategori penyakit ini tidak lepas dari gaya hidup yang kurang sehat seiring dengan berubahnya pola hidup.

Penyakit jantung yang umum dikenal dan paling banyak diderita adalah penyakit jantung koroner (PJK). Penyakit ini paling sering menyebabkan serangan jantung pada seseorang yang bisa menyebabkan kematian. Gejala-gejala yang umumnya terjadi pada penderita penyakit jantung yaitu irama jantung tak beraturan, dada tertekan seperti ditimpa beban berat, rasa sakit, terjepit atau terbakar. Rasa sakit ini bisa menjalar ke seluruh dada, bahu kiri, lengan kiri, punggung, leher bawah dan rahang leher bawah. Dirasakan seperti tercekik atau sesak yang berlangsung selama 20 menit disertai keringat dingin, rasa lemas, dan berdebar bahkan terkadang pingsan, dan sering terjadi di pagi hari, serta bibir kebiruan saat melakukan aktivitas dan tungkai bawah bengkak.

Penyakit jantung dapat memberikan perbedaan pengaruh pada pasien yang berbeda untuk tingkatan penyakit yang berbeda pula. Sebuah gejala juga bisa mengindikasikan beberapa penyakit jantung yang berbeda. Menurut NYHA (New York Heart Assosiation), Penyakit jantung dibagi dalam 4 kelas yaitu 1, 2, 3, dan 4 (Schulman, 2004) yaitu :

1. Kelas pertama adalah penyakit jantung kategori ringan, dimana penderita tidak mengalami sesak napas atau jantung berdebar. Jadi seakan-akan penderita baik-baik saja, tanpa keterbatasan aktifitas fisik.
2. Kelas kedua adalah penyakit jantung kategori sedang, dimana penderita sehari-hari merasa sehat tetapi begitu beraktivitas sedikit berat, seperti berlari, maka jantung terasa sesak, berdebar atau cepat lelah.
3. Kelas ketiga sudah termasuk penyakit jantung kategori berat; saat istirahat penderita merasa nyaman, tetapi saat mengerjakan pekerjaan sehari-hari kendati aktivitas itu ringan, penderita akan mengalami sesak atau muncul gejala kelemahan jantung
4. Kelas keempat atau sudah masuk kategori sangat berat, tanpa mengerjakan apa-apa pun penderita sudah menderita sesak

Rata-rata untuk keluhan penyakit jantung yang ringan bisa sembuh dalam 10 hari, untuk keluhan yang sedang bisa sembuh dalam 20 hari, sedangkan untuk keluhan penyakit jantung yang sudah parah bisa sembuh dalam 30 hari. Namun angka ini adalah rata-rata, hasil pada setiap pasien bisa bervariasi.

Berikut ini beberapa parameter yang dapat mendukung untuk pengambilan keputusan seseorang terdiagnosa penyakit jantung atau tidak, yaitu umur (age), tekanan darah tinggi (tresbps), kolesterol (chol), denyut jantung (thalach), dan *oldpeak*.

1. Umur
Semakin bertambahnya umur seseorang, maka keadaan sistem kardiovaskuler semakin berkurang. Banyak studi menunjukkan sekitar 3% dari semua kasus penyakit jantung terjadi pada usia dibawah 40 tahun dan pada orang yang berumur 65 tahun ke atas, ditemukan 20% pada laki-laki dan 12% pada wanita.
2. Tekanan darah tinggi
Tekanan darah tinggi merupakan faktor yang dapat dipakai sebagai indikator untuk menilai sistem *kardiovaskular*. Tekanan

darah seseorang dipengaruhi oleh berbagai faktor di antaranya adalah perubahan posisi tubuh dan aktivitas fisik.

3. Kolesterol

Resiko penyakit jantung meningkat sesuai dengan peningkatan kadar kolesterol. Kolesterol dalam tubuh yang berlebihan akan tertimbun di dalam dinding pembuluh darah dan menimbulkan suatu kondisi yang disebut *aterosklerosis* yaitu penyempitan atau pengerasan pembuluh darah (Lipi, 2009).

4. Denyut jantung

Denyut jantung yang meningkat, kekuatan kontraksi yang tinggi, tegangan dinding vertikal yang meningkat merupakan beberapa faktor yang dapat meningkatkan kebutuhan akan oksigen dari otot-otot jantung.

5. *Oldpeak*

Oldpeak adalah besar ST depresi pada elektrokardiograf. Depresi segmen ST dapat menurun apabila beristirahat secara relative (Adeli, 2010).

2.2 Data Mining

2.2.1 Pengertian Data Mining

Lahirnya *data mining* diawali oleh kebutuhan terhadap informasi dari data yang melimpah sebagai pendukung keputusan untuk solusi bisnis dan dukungan infrastruktur bidang teknologi informasi (Moertini, 2002).

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan dalam basis data yang besar. Kata *mining* sendiri memiliki arti usaha untuk memperoleh sesuatu yang berharga dari material dasar dalam jumlah besar. Berdasarkan pengertian kata *mining* maka *data mining* dapat diartikan usaha untuk memperoleh sesuatu yang berharga (informasi) dari data dalam jumlah besar.

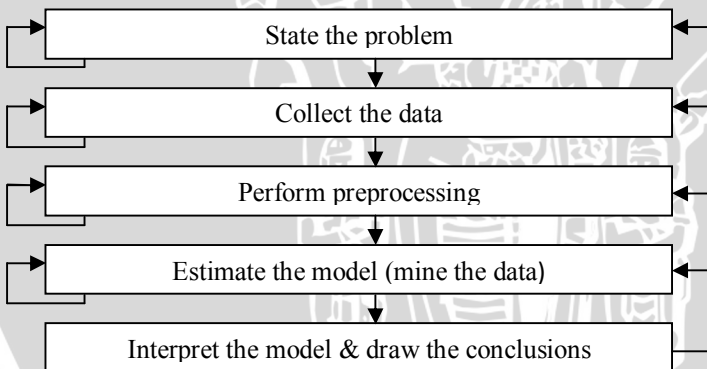
Menurut Han dan Kamber (2001), *data mining* merupakan solusi yang mampu menemukan kandungan informasi yang tersembunyi berupa pola dan aturan dari sekumpulan data yang besar agar mudah dipahami. Informasi yang tersembunyi ini sangat menguntungkan dari sudut pandang penelitian, bisnis dan lainnya

2.2.2 Proses Data Mining

Prosedur yang umum digunakan untuk permasalahan *data mining* meliputi tahap-tahap sebagai berikut (Kantardzic, 2003):

1. Menentukan permasalahan dan merumuskan hipotesis.
Pada tahap ini, ditentukan variabel-variabel dan hipotesis awal.
2. Mengumpulkan data
Tahap ini berkaitan dengan bagaimana data dihasilkan dan dikumpulkan
3. *Preprocessing* data
Dilakukan pembersihan terhadap *outlier*, penanganan *missing value* maupun transformasi data.
4. Memperkirakan model
Pemilihan dan penerapan teknik *data mining* yang sesuai adalah tugas utama dalam tahap ini.
5. Menafsirkan model dan menarik kesimpulan. Pada tahap ini, dilakukan penafsiran model untuk membantu dalam pengambilan keputusan.

Tahap-tahap pada *data mining* ditampilkan oleh Gambar 2.1.



Gambar 2.1 Tahapan *Data Mining* (Kantardzic, 2003)

2.2.3 Teknik Data Mining

Teknik – teknik *data mining* yang paling populer antara lain :

1. *Association Rule Mining*
Association Rule Mining adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi *item*.
2. *Classification*

Classification adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data dengan tujuan untuk dapat memperkirakan kelas dari suatu obyek yang labelnya tidak diketahui.

3. *Clustering*

Clustering adalah proses pengelompokkan data tanpa berdasarkan kelas data tertentu. *Clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui.

Dalam skripsi ini, digunakan teknik *data mining*, yaitu klasifikasi.

2.3 Logika Fuzzy

Kata *fuzzy* merupakan kata sifat yang berarti kabur, tidak jelas. *Fuzziness* atau kekaburan atau ketidakjelasan atau ketidakpastian selalu meliputi keseharian manusia. Orang yang belum pernah mengenal logika *fuzzy* pasti akan mengira bahwa logika *fuzzy* adalah sesuatu yang rumit dan tidak menyenangkan. Namun, sekali seseorang mulai mengenalnya, pasti akan tertarik untuk ikut mempelajari logika *fuzzy*. Logika *fuzzy* dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika *fuzzy* modern dan metodis baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada sejak lama (Kusumadewi, 2002).

Konsep logika *fuzzy* dikembangkan oleh Prof. Lofti Zadeh pada tahun 1965. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama (Kusumadewi, 2004). Logika *fuzzy* digunakan untuk menerjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat dan sangat.

2.3.1 Himpunan Fuzzy

Himpunan *fuzzy* adalah himpunan elemen yang setiap elemennya memiliki derajat keanggotaan tertentu. Himpunan *fuzzy* digolongkan oleh suatu fungsi keanggotaan (*membership function*) yang memberikan nilai derajat keanggotaan tertentu kepada setiap elemen dalam himpunan tersebut. Nilai derajat keanggotaan tersebut

berada pada interval $[0,1]$. Adapun logika crips atau logika klasik, derajat keanggotaan suatu elemen dalam suatu himpunan hanya ditentukan dengan dua nilai yaitu nol dan satu (Tangel, 2008).

Himpunan fuzzy dapat dilakukan 3 operasi, yaitu:

1. *Intersection* dari dua himpunan fuzzy A dan B dengan fungsi keanggotaan berturut-turut $\mu_A(x)$ dan $\mu_B(x)$ didefinisikan sebagai $\mu(A \cap B)(x) = \min(\mu_A(x), \mu_B(x))$ (2.1)
2. *Union* dari dua himpunan fuzzy A dan B dengan fungsi keanggotaan berturut-turut $\mu_A(x)$ dan $\mu_B(x)$ didefinisikan sebagai $\mu(A \cup B)(x) = \max(\mu_A(x), \mu_B(x))$ (2.2)
3. *Complement* dari suatu himpunan fuzzy dengan notasi \bar{A} dan fungsi keanggotaan $\mu_A(x)$ didefinisikan sebagai $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ (2.3)

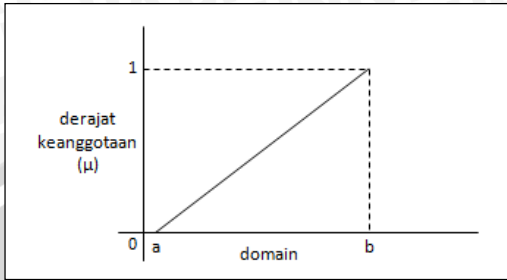
2.3.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Representasi dari fungsi keanggotaan ini dapat digambarkan dengan dua bentuk yaitu linear atau garis lurus dan kurva. Ada beberapa fungsi yang bisa digunakan (Kusumadewi, 2004), yaitu:

a. Representasi Linier

Pada representasi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Terdapat 2 keadaan himpunan fuzzy yang linier:

1. *Representasi Linear Naik*. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Representasi linear naik dapat dilihat pada Gambar 2.2.

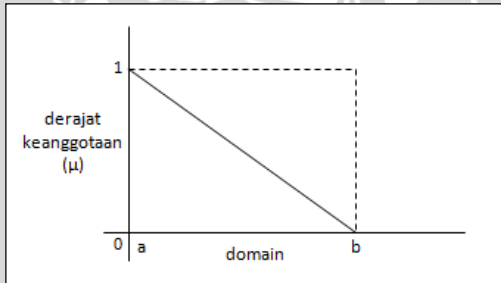


Gambar 2.2 Representasi Linear Naik

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots (2.4)$$

2. Representasi Linear Turun. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Representasi linear turun dapat dilihat pada Gambar 2.3.



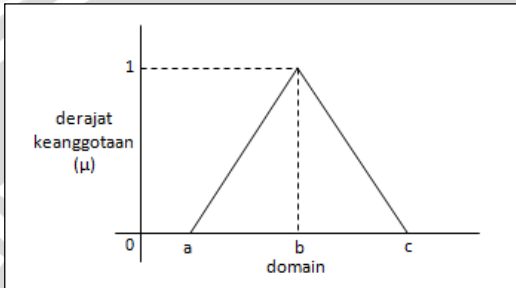
Gambar 2.3 Representasi Linear Turun

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots (2.5)$$

b. Representasi Kurva Segitiga

Fungsi keanggotaan segitiga ditandai adanya 3 parameter {a,b,c} yang akan menentukan koordinat x dari tiga sudut. Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear). Representasi kurva segitiga dapat dilihat pada Gambar 2.4.



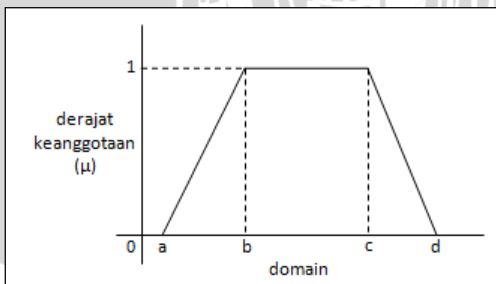
Gambar 2.4 Kurva Segitiga

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ \frac{b-x}{c-b}; & b \leq x \leq c \end{cases} \dots\dots\dots (2.6)$$

c. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja terdapat beberapa titik yang memiliki nilai keanggotaan 1. Representasi kurva trapesium dapat dilihat pada Gambar 2.5.



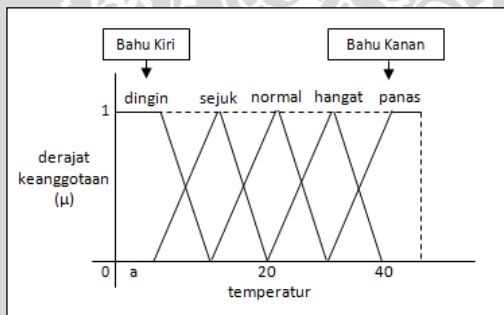
Gambar 2.5 Kurva Trapesium

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c \leq x \leq d \end{cases} \dots\dots\dots (2.7)$$

d. Representasi Kurva Bentuk Bahu

Representasi fungsi keanggotaan *fuzzy* dengan menggunakan kurva bahu pada dasarnya adalah gabungan dari kurva segitiga dan kurva trapesium. Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang pada salah sisi dari variabel *fuzzy* yang ditinjau ini terdapat nilai yang konstan, yaitu pada himpunan ekstrim kiri dan ekstrim kanan. Representasi kurva bentuk bahu dapat dilihat pada Gambar 2.6.



Gambar 2.6 Kurva Bentuk Bahu

2.3.3 Sistem Inferensi Fuzzy Mamdani

Fuzzy mamdani merupakan salah satu metode yang sangat fleksibel dan memiliki toleransi pada data yang ada. *Fuzzy* Mamdani sering dikenal sebagai metode Min-Max. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 (Lee, 2005). *Fuzzy* mamdani memiliki kelebihan yakni, lebih intuitif, diterima oleh banyak pihak. Dengan berdasarkan logika *fuzzy* akan dihasilkan suatu model *fuzzy* mamdani yang mampu mengklasifikasi penyakit jantung.

Penggunaan *fuzzy* mamdani sama halnya dengan penggunaan metode peramalan pada bidang statistik. Penentuan analisis berdasarkan pendekatan *fuzzy* lebih efisien dalam pendekatan menggunakan angka dibanding dengan metode peramalan. Peramalan dalam statistik dapat menghasilkan galat error lebih besar dari pendekatan *fuzzy*. Dengan melakukan pendekatan *fuzzy* menghasilkan out put yang lebih dekat dengan keadaan sebenarnya.

2.3.4 Fuzzifikasi

Fuzzifikasi adalah proses pengubahan data keanggotaan dari himpunan suatu bobot skor biasa (konvensional) ke dalam keanggotaan himpunan bilangan *fuzzy*. Proses fuzzifikasi memerlukan suatu fungsi keanggotaan (*membership function*) untuk mendapatkan derajat keanggotaan suatu bobot skor ke dalam suatu himpunan (kelas). Fungsi keanggotaan dibuat berdasarkan pendekatan fungsi keanggotaan (Kainz, 2003).

2.3.5 Fungsi Implikasi

Fungsi implikasi yang digunakan pada pengambilan keputusan dengan metode Mamdani ada 2 yaitu fungsi implikasi untuk premis OR adalah Max dan fungsi implikasi untuk premis AND adalah Min

2.3.6 Komposisi Aturan

Metode komposisi aturan pada FIS mamdani ada 3 yaitu:

- Metode Max (union). Metode ini mengambil nilai Max aturan kemudian menggunakannya untuk modifikasi daerah *fuzzy* dan mengaplikasikannya ke output dengan menggunakan operator OR (union). Secara umum metode ini dapat dituliskan sebagai berikut:

$$\mu_{sf}[x_i] = \max(\mu_{sf}[x_i], \mu_{kf}[x_i]) \dots\dots\dots (2.8)$$

Dengan:

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzys* sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

- Metode Additive sum. Metode ini, solusi himpunan *fuzzy* diperoleh dengan melakukan *bounded-sum* terhadap semua

output daerah *fuzzy*. Secara umum metode ini dapat dituliskan sebagai berikut:

$$\mu_{sf}[x_i] = \min(1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \dots\dots\dots (2.9)$$

- Metode Probabilistik OR. Metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua output daerah *fuzzy*. Secara umum metode ini dapat dituliskan sebagai berikut:

$$\mu_{sf}[x_i] = (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] \times \mu_{kf}[x_i]) \dots\dots\dots (2.10)$$

2.3.7 Defuzzifikasi

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 keluarannya.

Ada 5 metode defuzzifikasi pada sistem inferensi *fuzzy* Mamdani (Sutikno, 2000), yaitu:

1. Metode Centroid

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy*, secara umum dirumuskan pada persamaan 2.11 untuk variabel kontinyu dan persamaan 2.12 untuk variabel diskrit.

$$\mu(x) = \frac{\int_a^b x\mu(x)dx}{\int_a^b \mu(x)dx} \dots\dots\dots (2.11)$$

$$\mu(x) = \frac{\sum_{i=1}^n x_i\mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \dots\dots\dots (2.12)$$

Dengan:

x_i = nilai tiap titik sampel

$\mu(x_i)$ = derajat keanggotaan titik sampel x_i

2. Metode Bisektor

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan separo

dari jumlah total nilai keanggotaan pada daerah fuzzy. Secara umum dituliskan pada persamaan 2.13.

$$\int_{\alpha}^{z^{BOA}} \mu(z) dz = \int_{z^{BOA}}^{\beta} \mu(z) dz \quad \dots\dots\dots (2.13)$$

Dengan :

$$\alpha = \min \{z | z \in Z\}$$

$$\beta = \max \{z | z \in Z\}$$

3. Metode Mean of Maximum (MOM)

Pada solusi ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

4. Metode Largest of Maximum (LOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

5. Metode Smallest of Maximum (SOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

2.4 Decision Tree

Decision tree merupakan suatu pendekatan yang sangat populer dan praktis dalam machine learning untuk menyelesaikan permasalahan klasifikasi. Metode ini digunakan untuk memperkirakan nilai diskret dari fungsi target, yang mana fungsi pembelajaran direpresentasikan oleh decision tree (Liang, 2005). *Decision tree* juga dapat dikatakan sebagai *flowchart* seperti struktur tree, dimana tiap node internal menunjukkan sebuah tes pada sebuah atribut, tiap cabang menunjukkan hasil dari tes dan setiap node leaf menunjukkan kelas-kelas atau distribusi kelas.

Decision tree merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode *decision tree* mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. *Decision tree* sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain (Quinlan, 1993).

2.5 Algoritma C4.5

Algoritma C4.5 adalah algoritma klasifikasi data dengan teknik pohon keputusan yang terkenal dan disukai, karena memiliki kelebihan seperti dapat mengolah data numerik dan diskret, dapat menangani nilai atribut yang hilang, menghasilkan aturan-aturan yang mudah diinterpretasikan dan tercepat di antara algoritma-algoritma yang menggunakan memori utama di komputer (Quinlan, 1993).

Algoritma C4.5 dapat dijelaskan sebagai berikut:

Input : sampel training, label training, atribut

1. Membuat simpul akar untuk pohon yang dibuat.
2. Jika semua sampel positif, berhenti dengan suatu pohon dengan satu simpul akar, beri label (+)
3. Jika semua sampel negatif, berhenti dengan suatu pohon dengan satu simpul akar, beri label (-)
4. Jika atribut kosong, berhenti dengan suatu pohon dengan satu simpul akar, dengan label sesuai nilai yang terbanyak yang ada pada label training
5. Untuk yang lain, Mulai :
 - a. $A \leftarrow$ atribut yang mengklasifikasikan sampel dengan hasil terbaik (berdasarkan gain ratio)
 - b. Atribut keputusan untuk simpul akar \leftarrow
 - c. Untuk setiap nilai v_i yang mungkin untuk A
 - a) Tambahkan cabang dibawah akar yang berhubungan dengan $A=v_i$
 - b) Tentukan sampel S_{v_i} sebagai subset dari sampel yang mempunyai nilai v_i untuk atribut A
 - c) Jika sampel S_{v_i} kosong,
 - Dibawah cabang tambahkan simpul daun dengan label=nilai yang terbanyak yang ada pada label training
 - Yang lain, tambah cabang baru dibawah cabang yang sekarang C4.5 (sampel training, label training, atribut-[A])
 - d. Berhenti.

Algoritma C4.5 menggunakan rasio perolehan (*gain ratio*). Sebelum menghitung rasio perolehan, perlu menghitung dulu nilai informasi dalam satuan bits dari suatu kumpulan objek. Cara

menghitungnya dilakukan dengan menggunakan konsep *Entropy* yang akan dijelaskan dibawah ini.

2.5.1 Entropy, Information Gain, Split Info, Gain Ratio C4.5

Entropy adalah suatu parameter untuk mengukur tingkat keberagaman (heterogenitas) dari kumpulan data. Semakin heterogen, nilai entropi semakin besar (Khairina, 2007). $Entropy(S)$ sama dengan 0, jika semua contoh pada S berada dalam kelas yang sama. $Entropy(S)$ sama dengan 1, jika jumlah contoh positif dan negative dalam S adalah sama. $Entropy(S)$ lebih dari 0 tetapi kurang dari 1, jika jumlah contoh positif dan negatif dalam S tidak sama.

Misalkan S berisi s data sampel. Anggap atribut untuk class memiliki n nilai yang berbeda, C_i untuk $i=1, \dots, I$. Anggap s_i menjadi jumlah sampel S pada class C_i . Maka besar informationnya dapat dihitung dengan persamaan 2.14 (Quinlan, 1993).

$$I(s_1, s_2, \dots, s_n) = -\sum_{i=1}^n p_i * \log_2(p_i) \dots\dots\dots (2.14)$$

Dimana:

- n : Jumlah partisi S
- p_i : Probabilitas s_i terhadap s dari sampel yang mempunyai class C_i
- s_i : Jumlah data sampel S pada class C_i
- s : Jumlah data sampel dalam S

Misalkan atribut A mempunyai nilai v yang berbeda, $\{a_1, a_2, \dots, a_v\}$. Atribut A dapat digunakan untuk membagi S ke dalam v bagian $\{S_1, S_2, \dots, S_v\}$, dimana S_j berisi contoh di S yang mempunyai nilai a_j dari A . Jika A terpilih sebagai tes atribut maka bagian ini akan sesuai dengan pertumbuhan cabang dari simpul yang berisi S. Anggap s_{ij} sebagai jumlah *Entropy* atau informasi dari subset A dituliskan dalam persamaan 2.15 sebagai berikut.

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{s} I(s_{1j}, \dots, s_{mj}) \dots\dots\dots (2.15)$$

$\frac{s_{1j} + \dots + s_{mj}}{s}$ adalah bobot dari bagian j dan merupakan jumlah contoh pada subbagian dibagi oleh total jumlah contoh dalam S.

Nilai *entropy* terkecil adalah kemurnian (*purity*) terbesar pada pembagian subbagian. Untuk subbagian s_j nilai *information gain* dihitung menggunakan persamaan 2.16 sebagai berikut.

$$I(s_{1j}, s_{2j}, \dots, s_{nj}) = -\sum_{j=1}^m p_{1j} \log_2(p_{1j}) \dots \dots \dots (2.16)$$

Dimana $p_{1j} = \left| \frac{s_{1j}}{s_j} \right|$ adalah probabilitas pada contoh S_j kepunyaan kelas C_i . Jika atribut memiliki *missing value* maka perhitungan *entropy* hanya dilakukan pada atribut yang ada nilainya.

Information gain digunakan untuk memilih tes atribut pada setiap simpul dalam *tree*. Atribut dengan informasi tertinggi dipilih sebagai tes atribut untuk simpul tersebut. Secara matematis, *infomation gain* pada algoritma C4.5 dapat dituliskan pada persamaan 2.17 untuk atribut yang tak memiliki *missing value*.

$$Gain = I(s_1, s_2, \dots, s_m) - E(A) \dots \dots \dots (2.17)$$

Persamaan 2.18 untuk *gain* pada atribut yang memiliki *missing value*, dengan nilai *entropy* pada atribut yang ada nilainya.

$$Gain(S,A) = \text{Prob S yang diketahui} * I(s_1, s_2, \dots, s_m) - E(A) \dots \dots \dots (2.18)$$

Dimana:

A : Atribut dengan *missing value* yang sedang dicari nilai gainnya

S : Jumlah sample pada subset A yang diketahui nilainya

Untuk menghitung nilai *split info* pada suatu term baru dapat dilihat pada persamaan 2.19 untuk atribut yang tak memiliki *missing value*.

$$Split Info(S,A) = -\sum_{i=1}^n p_i \log_2 p_i \dots \dots \dots (2.19)$$

Sedangkan pada persamaan 2.20 adalah persamaan untuk *split info* pada atribut yang memiliki *missing value*.

$$Split Info(S,A) = -u * \log_2 u - \sum_{i=1}^n p_i \log_2 p_i \dots \dots \dots (2.20)$$

Dimana:

u : Probabilitas sampel pada atribut A terhadap s yang merupakan *missing value*

p_i : Probabilitas s_i terhadap s yang diketahui nilainya.

s_i : Jumlah data sampel S pada partisi ke i

s : Jumlah data sampel dalam S

Gain ratio dapat dicari dengan persamaan 2.21 berikut

$$\text{Gain Ratio (S,A)} = \frac{\text{Gain (S,A)}}{\text{SplitInfo (S,A)}} \dots\dots\dots (2.21)$$

Atribut dengan *gain ratio* maksimal akan dipilih sebagai *splitting attribute*. Perlu diperhatikan jika *split info* mendekati 0, maka perbandingan menjadi tidak stabil. Oleh karena itu, perlu ditambahkan batasan untuk memastikan bahwa *information gain* dari sebuah pengujian haruslah besar, dan minimal sama besar dengan *information gain* rata-rata dari seluruh pengujian (Khairina, 2007).

2.5.2 Pembangunan *Decision Tree* dengan Algoritma C4.5

Algoritma dasar untuk pembangunan *decision tree* pada C4.5 (dan turunannya) adalah algoritma *greedy* yang membangun pohon keputusan dari atas ke bawah (*top-down*) secara rekursif. Masukan dari algoritma ini adalah himpunan data yang berisi sampel-sampel data dan kandidat atribut yang harus ditelaah yang terdiri dari minimal sebuah atribut prediktor dan sebuah atribut kelas. Atribut prediktor dapat bertipe diskret atau numerik, sedangkan atribut kelas harus bertipe diskret (Quinlan, 1993).

Setelah tahap pembentukan pohon, tahap selanjutnya adalah pemangkasan pohon atau *pruning tree*. Teknik *pruning tree* yaitu teknik untuk memotong *rule* pada *decision tree*, jika *rule* yang dihasilkan sudah tidak signifikan. Dengan cara serupa dengan pasca-keputusan pemangkasan pohon, mengurangi kesalahan pemangkasan dengan cara menghapus salah satu dalam aturan kemudian bandingkan tingkat kesalahan pada set validasi sebelum dan setelah pemangkasan, jika memperbaiki kesalahan, lakukan proses *prune*. Pada skripsi ini dilakukan proses *pruning* dengan menggunakan *Threshold dalam Fuzzy Decision Tree* (FDT) sebagaimana akan dijelaskan pada subbab 2.7.1 dibawah ini.

2.6 Fuzzy Decision Tree

Fuzzy Decision tree merupakan suatu pendekatan yang sangat populer dan praktis dalam *machine learning* untuk menyelesaikan permasalahan klasifikasi yang mengalami ketidakpastian. *Fuzzy decision tree* memungkinkan untuk menggunakan nilai-nilai *numeric-symbolic* selama konstruksi atau saat mengklasifikasikan kasus-kasus baru. Manfaat dari teori himpunan *fuzzy* dalam *decision tree* ialah meningkatkan kemampuan atribut-atribut kuantitatif. bahkan dengan menggunakan teknik *fuzzy* dapat meningkatkan ketahanan saat melakukan klasifikasi kasus-kasus baru. (Romansyah,dkk, 2009).

2.6.1 Threshold Fuzzy Decision Tree (FDT)

Jika pada proses *learning* dari FDT dihentikan sampai semua data contoh pada masing-masing *leaf-node* menjadi anggota sebuah kelas, akan dihasilkan akurasi yang rendah. Oleh karena itu untuk meningkatkan akurasinya, proses *learning* harus dihentikan lebih awal atau melakukan pemotongan *tree* secara umum. Untuk itu diberikan 2 *threshold* yang harus terpenuhi jika *tree* akan diekspansi (Liang, 2005), yaitu:

1. *Fuzziness Control Threshold* (FCT) / θ_r
Jika proporsi dari himpunan data dari kelas C_k lebih besar atau sama dengan nilai *threshold* θ_r , maka hentikan ekspansi *tree*. Sebagai contoh: jika pada sebuah *sub-dataset* rasio dari kelas 1 adalah 90%, kelas 2 adalah 10% dan θ_r adalah 85% maka hentikan ekspansi *tree*
2. *Leaf Decision Threshold* (LDT) / θ_n
Jika banyaknya anggota himpunan data pada suatu *node* lebih kecil dari *threshold* θ_n , hentikan ekspansi *tree*. Sebagai contoh, sebuah himpunan data memiliki 600 contoh dengan θ_n adalah 2%. Jika jumlah data contoh pada sebuah *node* lebih kecil dari 12 (2% dari 600), maka hentikan ekspansi *tree*.

2.6.2 Fuzzy Entropy, Information Gain, Split Info, Gain Ratio

Pada metode *fuzzy decision tree* dengan algoritma C4.5, setiap data mungkin dapat masuk kedalam satu atau lebih dari kelas yang ada. Algoritma C4.5 mengklasifikasikan data kedalam kategori

berdasarkan nilai *gain ratio*. Atribut dengan *gain ratio* yang terbesar dipilih sebagai tes atribut untuk simpul tersebut (Tokumaru, 2009).

Pada himpunan data *fuzzy* terdapat penyesuaian rumus untuk menghitung nilai *entropy* untuk atribut, *information gain*, *split info* dan *gain ratio* karena adanya ekspresi data *fuzzy*. Persamaan 2.22 berikut adalah persamaan untuk mencari nilai *fuzzy entropy* dari keseluruhan data.

$$Info(S) = - \sum_{i=1}^k \left\{ \frac{freq(C_i, S)}{|S|} \times \log_2 \frac{freq(C_i, S)}{|S|} \right\} \dots \dots (2.22)$$

Dimana $Info(S)$ adalah *entropy* seluruh data dengan $S = \{s_1, s_2, \dots, s_x\}$ dan $freq(C_i, S)$ adalah frekuensi data sampel yang masuk kedalam kelas C_i . $|S|$ adalah jumlah data sampel yang termasuk dalam S . k adalah jumlah kategori yang membagi data kedalam beberapa kelas.

$Info_{X_p}$ adalah *entropy* untuk atribut X_p dimana sampel yang masuk kedalam T yang membagi kedalam beberapa subset $T_j (j: 1 - n)$ dengan atribut X_p yang dijelaskan dalam persamaan 2.24. Perhitungan $Info(T_j)$ dijelaskan pada persamaan 2.25.

$$Info_{X_p}(T) = \sum_{j=1}^n \frac{|T_j|}{|T|} \times Info(T_j) \dots \dots (2.24)$$

$$Info(T_j) = \sum_{i=1}^k \left\{ \frac{freq(C_i, T_j)}{|T_j|} \times \log_2 \frac{freq(C_i, T_j)}{|T_j|} \right\} \dots \dots (2.25)$$

Atribut X_p membagi data kedalam *fuzzy set* $T_j (j: 1 - n)$ dan diberikan derajat kemungkinan $\mu(T_j, S_h) (h: 1 - x)$. $freq(C_i, T_j)$ adalah frekuensi data sampel yang masuk kedalam kelas C_i dan termasuk kedalam subset T_j seperti pada persamaan 2.26.

$$freq(C_i, T_j) = \sum_{h=1}^x \{ \mu(C_i, S_h) \times \mu(T_j, S_h) \} \dots \dots (2.26)$$

Information gain pada atribut X_p dilihat pada persamaan 2.27 berikut

$$Gain(X_p) = Info(S) - Info_{X_p}(T) \dots \dots (2.27)$$

Algoritma C4.5 merupakan suksesor dari ID3 menggunakan *gain ratio* untuk memperbaiki *information gain*. Pendekatan ini menerapkan normalisasi pada *information gain* dengan menggunakan apa yang disebut sebagai *split info*. Selanjutnya, untuk menghitung nilai *split info* atau rasio perolehan yang perlu diketahui pada suatu term baru dapat dilihat pada persamaan 2.28

$$Split\ Info(X_p) = -\sum_{j=1}^n \frac{|T_j|}{|T|} \times \log_2 \frac{|T_j|}{|T|} \dots\dots (2.28)$$

Dimana T_j adalah jumlah *membeship function* dari pemecahan S pada atribut X_p dan T adalah jumlah data sampel dalam S . Setelah mendapatkan nilai *Split Info*, maka *gain ratio* dapat dicari dengan persamaan 2.29 berikut.

$$Gain\ Ratio(X_p) = \frac{Gain(X_p)}{Split\ Info(X_p)} \dots\dots (2.29)$$

Untuk menangani masalah *missing value* pada *fuzzy decision tree* adalah menggunakan metode yang sama seperti yang dikemukakan oleh Quinlan pada pembangunan decision tree. Caranya adalah membagi rata ke semua contoh anak-anak, jika fitur yang akan diuji *missing value*. Misalkan e_k adalah distribusi merata pada semua anak cabang jika nilai dari u_k^i pada atribut X_p *missing value* (Wang, 2003).

$$\mu_{a_p^i}(u_k^i) = \frac{1}{|D_i|} \dots\dots (2.30)$$

dimana jika u_k^i *missing value* dan $|D_i|$ adalah jumlah fuzzy set pada atribut A_i .

Information gain pada data uji dengan *missing value* jelas tidak dapat memberikan informasi tentang kelas keanggotaan. Oleh karena itu penilaian kandidat atribut harus diubah, sehingga atribut yang *missing value* dihilangkan. Misalkan diberi satu set referensi E memiliki *missing value* untuk atribut X_p . Kemudian perhitungan *Information gain* untuk atribut X_p dari fuzzy diubah sebagai berikut dengan perhitungan *entropy* pada atribut yang ada nilainya.

$$Gain(X_p) = \frac{|Tr|}{|T|} \times (Info(S) - Info_{X_p}(T)) \quad \dots \dots (2.31)$$

$$\alpha = \frac{|Tr|}{|T|} \quad \dots \dots (2.32)$$

Dimana T_r adalah jumlah data pada subset X_p yang diketahui nilainya dan T adalah jumlah data sampel pada subset X_p .

2.6.3 Pembangunan *Fuzzy Decision Tree* dengan Algoritma C4.5

Algoritma *fuzzy decision tree* C4.5 adalah sebagai berikut (Tokumaru, 2009).

1. Membuat Root node yang memiliki himpunan data fuzzy dengan nilai keanggotaan 1
2. Jika t node dengan himpunan data fuzzy D memenuhi kondisi berikut, maka itu adalah sebuah node daun dan ditugaskan oleh nama kelas.
 - proporsi C_k kelas lebih besar dari atau sama dengan Θ_r ,
 $\frac{D_{ck}}{D} \geq \Theta_r$
 - jumlah set data kurang dari Θ_n tidak ada atribut untuk klasifikasi yang lebih
3. Jika node D tidak ada memenuhi kondisi di atas, maka bukan daun-node. dan sub-node baru dihasilkan sebagai berikut:
 - menghitung information gain setiap A_i 's ($i=1, \dots, L$).
 - menghitung split info setiap A_i 's.
 - menghitung gain ratio setiap A_i 's, lalu pilih atribut uji A_{max} yang paling terbesar.
 - membagi D kedalam subset D_1, \dots, D_m menurut A_{max} , dimana nilai keanggotaan data dalam D_j adalah hasil dari nilai keanggotaan dalam D dan nilai $F_{max,j}$ nilai A_{max} di D
 - membangkitkan node baru $t_1 \dots t_m$ untuk Fuzzy subset D_1, \dots, D_m dan label Fuzzy set $F_{max,j}$ ke tepi yang menghubungkan antara node t_j dan t
 - mengganti D dengan D_j ($j=1,2, \dots, m$) dan ulangi dari 2 rekursif

2.7 Data Haberman's Survival

Data Haberman's Survival merupakan data hasil dari penelitian mengenai penyakit jantung di V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. Data ini disumbangkan oleh David W Aha. Data Haberman's survival terdiri dari 303 kasus pasien penderita penyakit jantung. Masing-masing data terdiri dari 5 atribut yaitu umur, tekanan darah, kolesterol, denyut jantung, *oldpeak* dan status klasifikasi pasien. Status klasifikasi pasien terdiri dari 5 kelas yaitu pasien sehat (*healthy*), pasien memiliki penyakit jantung dalam kategori ringan (*sick1*), sedang (*sick2*), berat (*sick3*) dan sangat berat (*sick4*).

Fungsi keanggotaan yang digunakan untuk mengubah atribut diperoleh dari internet yang digunakan dalam jurnal dan penelitian yang pernah dilakukan oleh Ali Adeli dan Mehdi Neshat tahun 2010. Atribut umur dibagi menjadi 4 variabel linguistik yaitu *young* (umur kurang dari 38), *middle* (diantara umur 33 dan 45 tahun), *old* (diantara umur 40 dan 58 tahun) dan *very old* (umur lebih dari 52 tahun). Dari pembagian itu dapat ditentukan derajat keanggotaan masing-masing variable linguistik menggunakan persamaan 2.6 berikut ini.

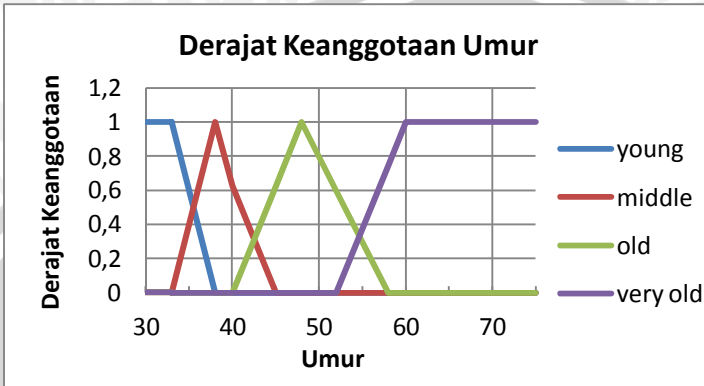
$$\mu_{young}(x) = \begin{cases} 1 & ; x < 33 \\ \frac{38-x}{5} & ; 33 \leq x < 38 \\ 0 & ; x \geq 38 \end{cases} \dots \dots (2.33)$$

$$\mu_{mid}(x) = \begin{cases} 0 & ; x < 33 \text{ atau } x \geq 45 \\ \frac{x-33}{5} & ; 33 \leq x < 38 \\ 1 & ; x = 38 \\ \frac{45-x}{7} & ; 38 \leq x < 45 \end{cases} \dots \dots (2.34)$$

$$\mu_{old}(x) = \begin{cases} 0 & ; x < 40 \text{ atau } x \geq 58 \\ \frac{x-40}{8} & ; 40 \leq x < 48 \\ 1 & ; x = 48 \\ \frac{58-x}{10} & ; 48 \leq x < 58 \end{cases} \dots \dots (2.35)$$

$$\mu_{veryold}(x) = \begin{cases} 0 & ; x < 52 \\ \frac{x-52}{8} & ; 52 \leq x < 60 \\ 1 & ; x \geq 60 \end{cases} \dots \dots (2.36)$$

Himpunan fuzzy untuk setiap variabel linguistik atribut umur menggunakan kurva berbentuk segitiga seperti pada Gambar 2.7.



Gambar 2.7 Fungsi keanggotaan Umur

Atribut tekanan darah dibagi menjadi 4 variabel linguistik, yaitu *low* (tekanan darah kurang dari 134 mm Hg), *medium* (diantara tekanan darah 127 – 153 mm Hg), *high* (diantara tekanan darah 142 – 172 mm Hg) dan *very high* (tekanan darah lebih dari 154 mm Hg). Dari pembagian itu dapat ditentukan derajat keanggotaan masing-masing variable linguistik menggunakan persamaan 2.6 berikut ini.

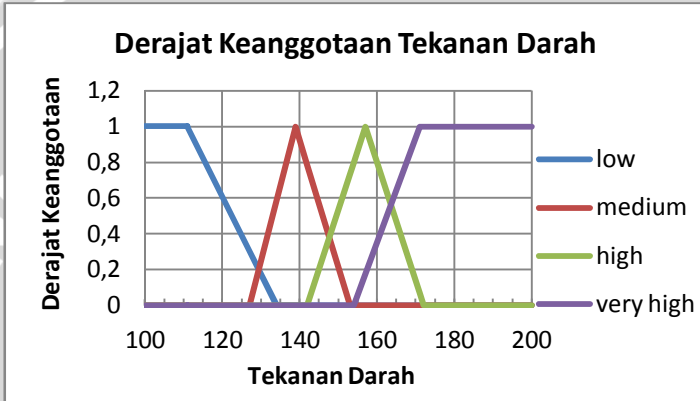
$$\mu_{low}(x) = \begin{cases} 1 & ; x < 111 \\ \frac{134-x}{23} & ; 111 \leq x < 134 \\ 0 & ; x \geq 134 \end{cases} \dots\dots (2.37)$$

$$\mu_{medium}(x) = \begin{cases} 0 & ; x < 127 \text{ atau } x \geq 153 \\ \frac{x-127}{12} & ; 127 \leq x < 139 \\ 1 & ; x = 139 \\ \frac{153-x}{14} & ; 139 \leq x < 153 \end{cases} \dots\dots (2.38)$$

$$\mu_{high}(x) = \begin{cases} 0 & ; x < 142 \text{ atau } x \geq 172 \\ \frac{x-142}{15} & ; 142 \leq x < 157 \\ 1 & ; x = 157 \\ \frac{172-x}{15} & ; 157 \leq x < 172 \end{cases} \dots\dots (2.39)$$

$$\mu_{veryhigh}(x) = \begin{cases} 0 & ; x < 154 \\ \frac{x-154}{17} & ; 154 \leq x < 171 \\ 1 & ; x \geq 171 \end{cases} \dots (2.40)$$

Himpunan fuzzy untuk setiap variabel linguistik atribut tekanan darah menggunakan kurva berbentuk segitiga seperti Gambar 2.8.



Gambar 2.8 Fungsi keanggotaan Tekanan Darah

Atribut kolesterol dibagi menjadi 4 variabel linguistik yaitu *low* (kolesterol kurang dari 197 mg/dL), *medium* (diantara kolesterol 188 – 250 mg/dL), *high* (diantara kolesterol 217 – 307 mg/dL) dan *very high* (kolesterol lebih dari 281 mg/dL). Dari pembagian itu dapat ditentukan derajat keanggotaan masing-masing variable linguistik menggunakan persamaan 2.6 berikut ini.

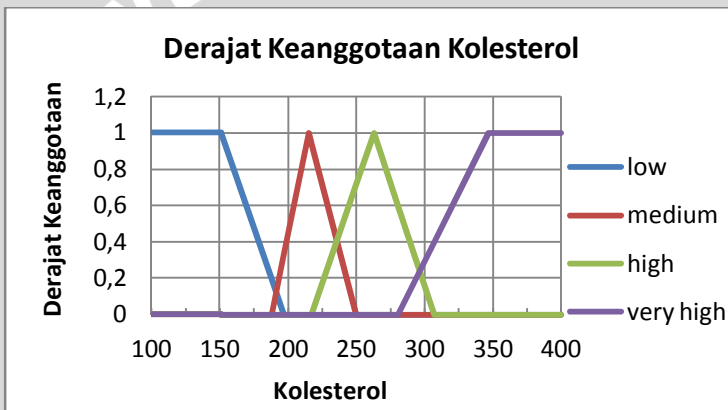
$$\mu_{low}(x) = \begin{cases} 1 & ; x < 151 \\ \frac{197-x}{46} & ; 151 \leq x < 197 \\ 0 & ; x \geq 197 \end{cases} \dots (2.41)$$

$$\mu_{medium}(x) = \begin{cases} 0 & ; x < 188 \text{ atau } x \geq 250 \\ \frac{x-188}{27} & ; 188 \leq x < 215 \\ 1 & ; x = 215 \\ \frac{250-x}{35} & ; 215 \leq x < 250 \end{cases} \dots (2.42)$$

$$\mu_{high}(x) = \begin{cases} 0; & x < 217 \text{ atau } x \geq 307 \\ \frac{x-217}{46}; & 217 \leq x < 263 \\ 1; & x = 263 \\ \frac{307-x}{44}; & 263 \leq x < 307 \end{cases} \dots\dots (2.43)$$

$$\mu_{veryhigh}(x) = \begin{cases} 0; & x < 281 \\ \frac{x-281}{66}; & 281 \leq x < 347 \\ 1; & x \geq 347 \end{cases} \dots\dots (2.44)$$

Himpunan fuzzy untuk setiap variabel linguistik atribut kolesterol menggunakan kurva berbentuk segitiga seperti pada Gambar 2.9.



Gambar 2.9 Fungsi keanggotaan Atribut Kolesterol

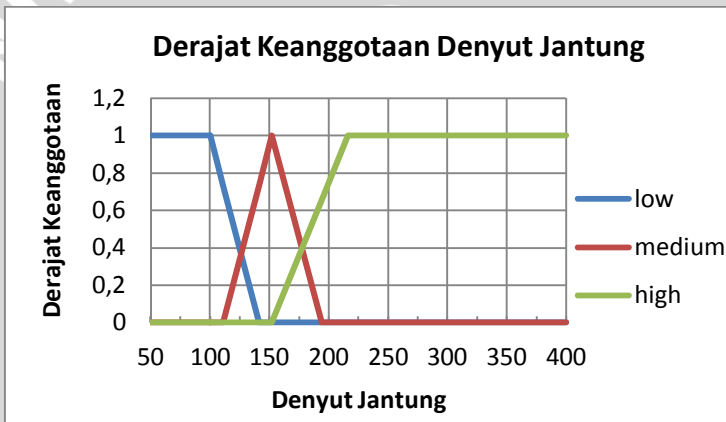
Atribut denyut jantung dibagi menjadi 3 variabel linguistik yaitu *low* (denyut jantung kurang dari 141), *medium* (diantara denyut jantung 111 – 194) dan *high* (denyut jantung lebih dari 152). Dari pembagian itu dapat ditentukan derajat keanggotaan masing-masing variable linguistik menggunakan persamaan 2.6 berikut ini.

$$\mu_{low}(x) = \begin{cases} 1; & x < 100 \\ \frac{141-x}{41}; & 100 \leq x < 141 \\ 0; & x \geq 141 \end{cases} \dots\dots (2.45)$$

$$\mu_{medium}(x) = \begin{cases} 0 ; x < 111 \text{ atau } x \geq 194 \\ \frac{x-111}{41} ; 111 \leq x < 152 \\ 1 ; x = 152 \\ \frac{194-x}{42} ; 152 \leq x < 194 \end{cases} \dots \dots (2.46)$$

$$\mu_{high}(x) = \begin{cases} 0 ; x < 152 \\ \frac{x-152}{64} ; 152 \leq x < 216 \\ 1 ; x \geq 216 \end{cases} \dots \dots (2.47)$$

Himpunan fuzzy untuk setiap variabel linguistik atribut denyut jantung menggunakan kurva berbentuk segitiga seperti pada Gambar 2.10.



Gambar 2.10 Fungsi keanggotaan Atribut Denyut Jantung

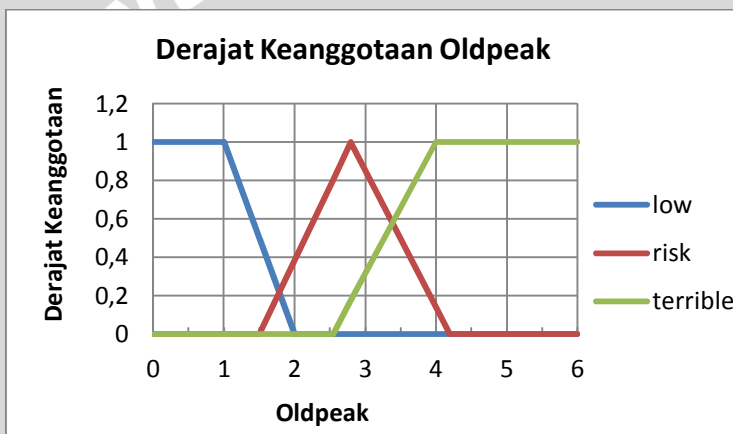
Atribut *oldpeak* dibagi menjadi 3 kelompok atau *linguistic term* yaitu *low* (*oldpeak* kurang dari 2), *risk* (diantara *oldpeak* 1.5 – 4.2) dan *terrible* (*oldpeak* lebih dari 2.55). Dari pembagian itu dapat ditentukan derajat keanggotaan masing-masing variable linguistik menggunakan persamaan 2.6 berikut ini.

$$\mu_{low}(x) = \begin{cases} 1 ; x < 1 \\ \frac{2-x}{1} ; 1 \leq x < 2 \\ 0 ; x \geq 2 \end{cases} \dots \dots (2.48)$$

$$\mu_{risk}(x) = \begin{cases} 0; & x < 1.5 \text{ atau } x \geq 4.2 \\ \frac{x-1.5}{1.3}; & 1.5 \leq x < 2.8 \\ 1; & x = 2.8 \\ \frac{4.2-x}{1.4}; & 2.8 \leq x < 4.2 \end{cases} \dots\dots (2.49)$$

$$\mu_{terrible}(x) = \begin{cases} 0; & x < 2.55 \\ \frac{x-2.55}{1.45}; & 2.55 \leq x < 4 \\ 1; & x \geq 4 \end{cases} \dots\dots (2.50)$$

Himpunan fuzzy setiap variabel linguistik atribut denyut jantung menggunakan kurva berbentuk segitiga seperti pada Gambar 2.11.



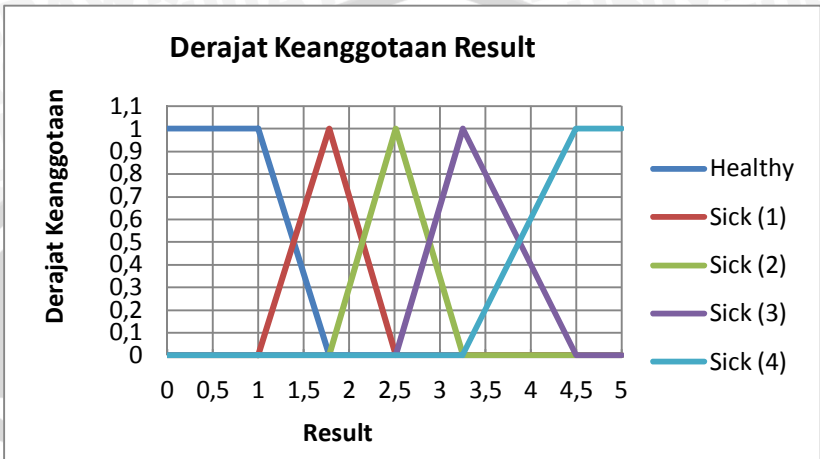
Gambar 2.11 Fungsi keanggotaan Atribut Oldpeak

Variabel output dibagi menjadi 5 kelas yaitu *healty* (sehat), *s1*(kategori ringan), *s2*(kategori sedang), *s3*(kategori berat) dan *s4*(kategori sangat berat). Fuzzy set dan range untuk variabel output dapat dilihat pada Tabel 2.1.

Tabel 2.1. Klasifikasi variabel output

Output Field	Range	Fuzzy Sets
Result	<1.78	Healthy
	1-2.51	Sick (s1)
	1.78-3.25	Sick (s2)
	2.51-4.5	Sick (s3)
	3.25>	Sick (s4)

Derajat keanggotaannya dapat dilihat pada Gambar 2.12.



Gambar 2.11 Fungsi keanggotaan Variabel Output

2.8 Akurasi

Akurasi adalah nilai derajat kedekatan dari pengukuran kuantitas untuk nilai sebenarnya (*true*). Nilai akurasi didapatkan dari hasil *rule* yang dihasilkan dari perhitungan *decision tree* kemudian di uji coba kan pada data testing dan menghasilkan derajat keakuratan dari *rule* tersebut setelah di uji coba kan pada data testing. Tingkat akurasi diperoleh dengan perhitungan sesuai dengan persamaan 2.51 (Nugraha, 2006).

$$Akurasi (\%) = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100\% \quad \dots \dots (2.51)$$

UNIVERSITAS BRAWIJAYA

