



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Pakar

Sistem pakar berasal dari bahasa Inggris *expert system* yang memiliki arti sistem informasi yang berisi dengan pengetahuan dari pakar sehingga dapat digunakan untuk konsultasi. Sistem pakar juga berarti “*a computer system that emulates the decision-making ability of a human expert*” (sistem komputer yang dapat atau memiliki kemampuan untuk mengambil sebuah keputusan seperti seorang ahli) (Jackson, Peter (1998)). Sistem pakar pun dibuat untuk menyelesaikan masalah yang sangat kompleks dengan memberikan penjelasan seperti seorang ahli dan tidak mengikuti prosedur pengembangannya seperti pada pemrograman konvensional lainnya. Sistem pakar pertama kalinya dibuat pada tahun 1970 dan berkembang pada tahun 1980. Sistem pakar merupakan *artificial intelligence* yang sukses dalam pengembangannya. Sistem pakar ini memiliki struktur yang unik dibandingkan dengan pemrograman *respond* yang biasa karena dibagi menjadi dua bagian, yang pertama adalah independen dari sistem pakar yaitu mesin inferensi dan bagian keduanya adalah *respond* yang berisi tentang pengetahuan dasar dari sistem pakar tersebut. Pada tahun 80an muncul bagian yang ketiga adalah antarmuka yang berfungsi untuk berkomunikasi dengan pengguna.

Untuk membangun sistem pakar yang baik diperlukan beberapa komponen, antara lain :

1. Antar Muka Pengguna ( *User Interface* )
2. Memori Kerja ( *Working Memory* )
3. Basis Pengetahuan ( *Knowledge Base* )
4. Mekanisme Inferensi ( *Inference Machine* )
5. Subsistem Penjelasan ( *Explanation Facility* )

Antar Muka Pengguna, sistem pakar digunakan untuk menggantikan seorang pakar dalam situasi tertentu, maka sistem harus dibuat sehingga orang awam pun dapat menggunakan sistem tersebut. Sistem pakar sendiri juga menyediakan komunikasi antar sistem dan pemakainya yang sering disebut sebagai antar muka. Antar muka haruslah *user friendly* karena bagi pengguna yang tidak ahli pada bidang yang diterapkan di sistem pakar akan sangat membantu mereka untuk melakukan atau setidaknya mengerti alur dari sistem pakar tersebut dan menggunakan sistem pakar tersebut sesuai dengan fungsinya.

Basis pengetahuan, merupakan kumpulan pengetahuan bidang–bidang tertentu pada tingkatan pakar dengan format tertentu. Pengetahuan ini didapat dengan cara mengumpulkan data–data dari beberapa pakar dan sumber –sumber pengetahuan lainnya. Pada sistem pakar, basis pengetahuan terpisah dengan mesin inferensi dimana

pemisahan ini membantu agar dapat lebih mudah dalam mengembangkan sistem pakar sesuai dengan perkembangan pengetahuan.

Mesin inferensi, merupakan otak dari sistem pakar yang merupakan perangkat lunak yang berguna untuk melakukan tugas inferensi penalaran sistem pakar atau sering disebut sebagai mesin pemikir. Mesin inilah yang bertugas untuk mencari solusi dari permasalahan.

Mesin inferensi adalah program respond yang menyediakan metodologi untuk melakukan penalaran suatu informasi yang diinput dalam sistem pakar pada basis pengetahuan dan memori kerja untuk mencapai kesimpulan–kesimpulan. Mesin inferensi ini memberikan arahan–arahan tentang bagaimana menggunakan pengetahuan dari sistem dengan membangun agenda yang mengelola dan mengontrol langkah–langkah yang diambil untuk menyelesaikan masalah ketika konsultasi sedang berlangsung.

Memori kerja, merupakan bagian dari sistem pakar yang menyimpan fakta–fakta yang terjadi pada saat konsultasi sedang berlangsung. Fakta–fakta inilah yang akan diproses oleh mesin inferensi untuk mendapatkan hasil atau keputusan untuk pemecahan masalah.

Subsistem penjelasan, digunakan untuk mencari respond dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif.

### 2.1.1 Keuntungan Sistem Pakar

Terdapat beberapa keuntungan dalam sistem pakar sebagai berikut:

(Kusumadewi)

1. Dapat menyimpan pengetahuan dan keahlian para pakar
2. Meningkatkan realibilitas dari sistem tersebut
3. Meningkatkan kualitas
4. Memiliki realibilitas
5. Memiliki kemampuan untuk mengakses pengetahuan

### 2.2 Fuzzy Logic

*Fuzzy Logic* adalah sebuah bentuk yang terdiri dari banyak sekali nilai logika atau yang berisi tentang logika probabilistik. *Fuzzy Logic* lebih memberikan alasan yang tepat dibandingkan memberikan jawaban yang tepat atau pasti. *Fuzzy Logic* memiliki kumpulan biner yang menentukan nilai benar antara 0 dan 1. *Fuzzy Logic* dikembangkan juga untuk menangani konsep kebenaran parsial yang memiliki nilai sangat benar atau sangat salah. *Fuzzy Logic* ini pertama kali diperkenalkan pada dunia pada tahun 1965 oleh Lotfi A. Zadeh dengan judul proposalnya “*Fuzzy Set Theory*”. *Fuzzy Logic* sudah banyak sekali di pakai dalam banyak bidang seperti untuk teori kontrol sampai pada kecerdasan buatan. *Fuzzy Logic* sebenarnya telah mulai dipelajari pada tahun 1920 sebagai “*infinite – valued logics notably* “ oleh *Lukasiewicz* dan *Tarski*(Francis Jeffry Pelletier,2002).

Teori *Fuzzy Logic* mendefinisikan kumpulan *fuzzy*. Terdapat masalah pada saat ingin mengaplikasikan *Fuzzy Logic* karena operan dari *Fuzzy Logic* tidak dikenali secara luas sehingga *Fuzzy Logic* biasanya menggunakan IF-THEN *rules* yang sudah dikenal secara umum.

Contoh penggunaan IF-THEN *rules Fuzzy Logic* adalah :

*IF temperature IS very cold THEN stop fan*

*IF temperature IS cold THEN turn down fan*

*IF temperature IS normal THEN maintain level*

*IF temperature IS hot THEN speed up fan*

Selain IF-THEN *rules* juga terdapat beberapa operator yang dapat digunakan pada *Fuzzy Logic* seperti *and*, *or* dan *not*.

Pada *Fuzzy Logic* sangat memungkinkan untuk membangun *database* relasional pada *fuzzy*. *Database* relasional *Fuzzy Logic* yang pertama dikembangkan oleh Maria Zemankova's. Setelah itu terdapat beberapa *database* yang dibuat untuk *Fuzzy Logic* seperti *Buckles-Perty Model*, *the Prade-Testemale Model*, *the Umano-fukami Model* yang dikembangkan oleh J.M. Medina, M.A. Vila dkk. Dalam *database Fuzzy Logic* pun terdapat beberapa bahasa *query* yang digunakan seperti SQLf yang dikembangkan oleh P. Bosc dkk dan bahasa lainnya adalah FSQL yang dikembangkan

oleh J. Galindo dkk. Bahasa–bahasa ini digunakan untuk memasukkan aspek dari *Fuzzy Logic* pada pernyataan SQL.

Ada beberapa alasan mengapa *Fuzzy Logic* digunakan untuk sistem pakar :

- a. Konsep *Fuzzy Logic* mudah dipahami. Konsep matematis yang mendasari penalaran *Fuzzy Logic* sangat sederhana dan mudah dimengerti
- b. *Fuzzy Logic* memiliki tingkat fleksibel yang tinggi
- c. *Fuzzy Logic* memiliki toleransi terhadap data–data yang tidak tepat
- d. *Fuzzy Logic* dapat memodelkan fungsi–fungsi non-linear yang sangat kompleks
- e. *Fuzzy Logic* dapat membangun dan mengaplikasikan pengetahuan para pakar secara langsung tanpa proses pelatihan
- f. *Fuzzy Logic* didasarkan pada bahasa alami

Terdapat beberapa hal yang perlu diketahui pada system *Fuzzy Logic* :

a. *Variabel Fuzzy*

*Variable fuzzy* merupakan *variable* yang hendak dibahas dalam suatu *system Fuzzy*. Seperti : *temperature*, kondisi.

b. Himpunan *Fuzzy*

Himpunan *Fuzzy* merupakan grup yang mewakili suatu kondisi dari *variable fuzzy*

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu *variable fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilainya dapat berupa *negative* ataupun positif.

d. Domain

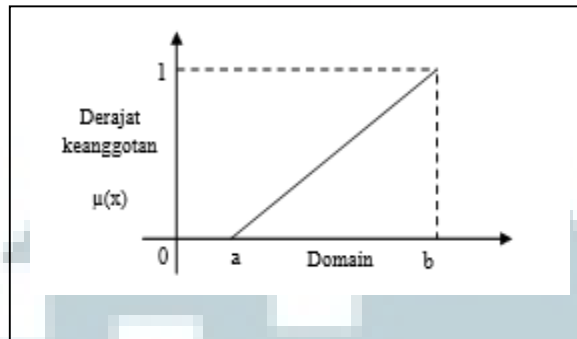
Domain himpunan *fuzzy* merupakan keseluruhan nilai yang diizinkan dalam semesta pembicara dan boleh dioperasikan dalam suatu himpunan *fuzzy*.

Fungsi Keanggotaan adalah kurva yang menunjukkan titik–titik input dalam nilai keanggotaan yang memiliki nilai interval 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan pendekatan fungsi. Terdapat beberapa fungsi yang dapat digunakan:

- Representasi linier

Pada representasi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai garis lurus. Pada representasi linier terdapat dua kemungkinan yaitu linier naik maupun turun



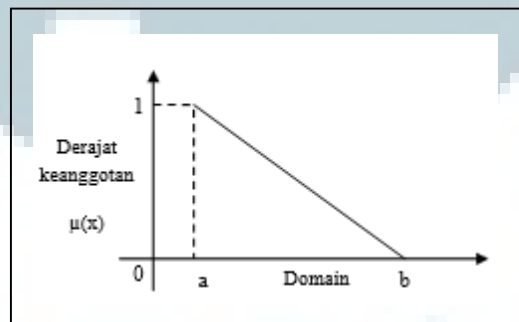


Gambar 0.1 Representasi Linier Naik

(<http://yusronrijal.wordpress.com/category/artificial-intelligence/fuzzy-logic/>)

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots\dots (2.1)$$



Gambar 0.2 Representasi Linier Turun

(<http://yusronrijal.wordpress.com/category/artificial-intelligence/fuzzy-logic/>)

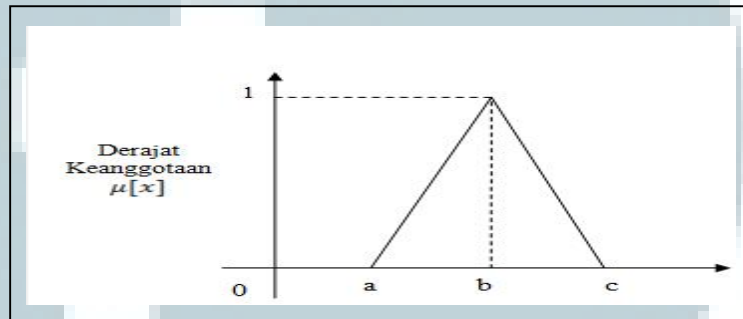
Fungsi Keanggotaan :

Fungsi Keanggotaan :

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

- Representasi kurva segitiga

Kurva segitiga merupakan gabungan dari dua garis linier. Nilai-nilai di sekitar  $b$  memiliki derajat keanggotaan yang turun menjauhi derajat keanggotaan satu.



Gambar 0.3 Kurva Segitiga

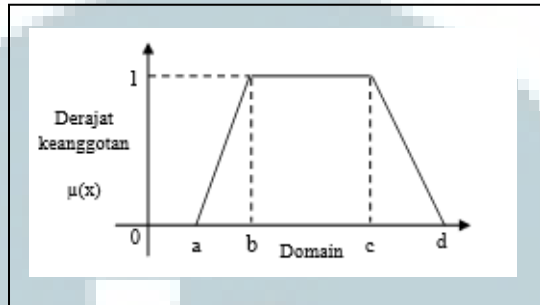
(<http://yusronrijal.wordpress.com/category/artificial-intelligence/fuzzy-logic/>)

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x-a)/(b-a); & a \leq x \leq b \\ (b-x)/(c-b); & b \leq x \leq c \end{cases} \dots\dots\dots(2.3)$$

- Representasi kurva trapesium

Kurva trapesium pada dasarnya sama seperti kurva segitiga tetapi pada rentang tertentu terdapat beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 0.4 Representasi Kurva Segitiga  
 (<http://yusronrijal.wordpress.com/category/artificial-intelligence/fuzzy-logic/>)

Fungsi keanggotaan :

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

Pada himpunan *fuzzy logic*, satu himpunan fuzzy dapat digabungkan dengan himpunan *fuzzy logic* yang lainnya dengan menggunakan operator. Ada tiga operator dasar yang diciptakan oleh Zadeh, yaitu :

a. Operator AND

Hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan–himpunan yang bersangkutan

$$\mu_{AB} = M(\mu_A(x), \mu_B(y)) \dots\dots\dots(2.5)$$

b. Operator OR

Hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan–himpunan yang bersangkutan.

$$\mu_{AB} = \text{MAX}(\mu_A(x), \mu_B(y)) \dots\dots\dots(2.6)$$

c. Operator Komplemen (*Complement*)

Hasil operasi dengan operator komplemen diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dengan 1.

$$\mu_{A^c} = 1 - \mu_A(x) \dots\dots\dots(2.7)$$

Terdapat beberapa metode sistem inferensi pada logika *fuzzy*. Salah satu yang paling sering dipakai adalah metode Mamdani. Metode Mamdani sering digunakan karena strukturnya yang sederhana yang menggunakan operasi MIN-MAX.

Untuk mendapatkan *output* diperlukan empat tahapan (Kusumadewi,2003) :

a. Pembentukan Himpunan *Fuzzy*

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

b. Aplikasi fungsi implikasi

Fungsi implikasi yang digunakan pada metode ini adalah MIN. Secara umum dapat digambarkan sebagai berikut

$$\mu A \cap B = \min(\mu a[X], \mu b[x]) \dots\dots\dots(2.8)$$

c. Komposisi Aturan

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

Tiga metode tersebut adalah :

1. Metode Max (*Maximum*)

Metode Max mengambil solusi dari himpunan *fuzzy* yang diperoleh dengan cara mengambil nilai maksimum aturan, yang kemudian diambil digunakan untuk mengubah daerah *fuzzy*, dan mengaplikasikan ke *output* dengan menggunakan operator *OR* (*union*), maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proporsi. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] \leftarrow \text{MAX}(\mu_{sf}[x_i], \mu_{kf}[x_i]) \dots\dots\dots(2.9)$$

Keterangan rumus :

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

$\mu_{sf}[x_i] = \text{nilai keanggotaan solusi fuzzy sampai ke aturan } i$

## 2. Metode Additive (Sum)

Metode *additive* mengambil solusi dari himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua output daerah *fuzzy*. Dapat dituliskan :

$$\mu_{sf}[x_i] \leftarrow \text{MIN}(1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \dots\dots\dots(2.10)$$

Keterangan :

$\mu_{sf}[x_i] = \text{nilai keanggotaan solusi fuzzy sampai ke aturan } i$

$\mu_{kf}[x_i] = \text{nilai keanggotaan solusi fuzzy sampai ke aturan } i$

## 3. Metode Probabilistik OR

Metode *Probabilistik OR* mengambil solusi himpunan *fuzzy* dengan cara melakukan *product* terhadap semua *output* daerah *fuzzy*. Dapat dituliskan dengan cara:

$$\mu_{sf}[x_i] \leftarrow (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] * \mu_{kf}[x_i]) \dots\dots\dots(2.11)$$

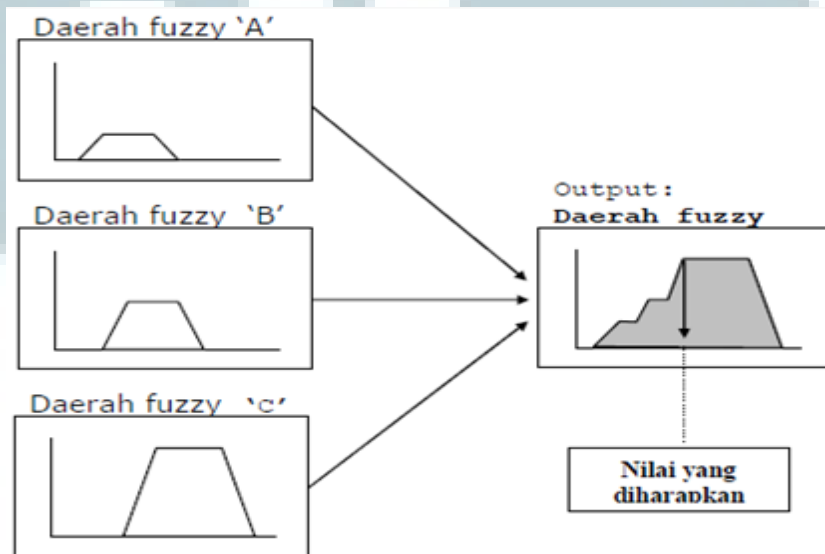
Keterangan :

$\mu_{sf}[x_i] = \text{nilai keanggotaan solusi fuzzy sampai ke aturan } i$

$\mu_{kf}[x_i] = \text{nilai keanggotaan solusi fuzzy sampai ke aturan } i$

d. Penegasan (*defuzzyfikasi*)

Input dari proses *defuzzyfikasi* 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 kan menghasilkan gambar sebagai berikut.



Gambar 0.5 Defuzzyfikasi

(<http://informatika.web.id/metode-mamdani.htm>)

Terdapat beberapa metode *defuzzyfikasi* pada komposisi aturan Mamdani (Kusumadewi), antara lain:

1. Metode *Centroid*, pada metode ini solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy*. Secara umum dapat digambarkan.

$$Z^* = \frac{\int_z Z\mu(z)dz}{\int_z \mu(z)dz} \rightarrow \text{untuk semesta kontinu}$$

$$Z^* = \frac{\sum_{j=1}^n Z_j\mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \rightarrow \text{untuk semesta diskret}$$

2. Metode Bisektor, pada metode ini solusi *crisp* diperoleh dengan cara mengambil nilai pada domain yang memiliki nilai keanggotaan separuh dari jumlah total nilai keanggotaan pada daerah *fuzzy*.
3. Metode *Mean of Maximum* (MOM), pada metode ini solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.
4. Metode *Largest of Maximum*, pada metode ini solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.
5. Metode *Smallest of Maximum*, pada metode ini solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

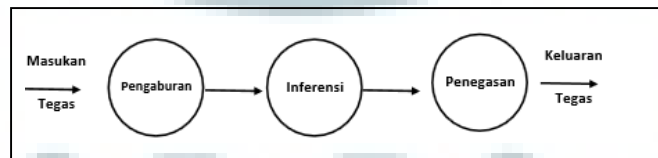
Sistem Inferensi *fuzzy* merupakan kerangka komputasi yang didasarkan pada teori himpunan *fuzzy* berbentuk IF-THEN dan penalaran *fuzzy*. Sistem inferensi *fuzzy* menerima *input crisp* (himpunan tegas). *Input* yang diterima ini kemudian dikirim ke basis pengetahuan yang berisi  $n$  aturan *fuzzy* dalam bentuk IF-THEN. Hasil-hasil tersebut akan dicari untuk setiap aturan dan jika ternyata terdapat jumlah aturan lebih



dari satu maka, akan dilakukan agregasi dari semua aturan. Selanjutnya hasil dari agregasi akan dilakukan *defuzzyfikasi* untuk mendapatkan nilai *crisp* (himpunan tegas) sebagai *output* sistem.

Penerapan *fuzzy logic* dapat meningkatkan kinerja sistem kendali dengan menekan munculnya fungsi–fungsi liar pada keluaran yang disebabkan oleh fluktuasi pada *variable* masukannya. Pendekatan pada *Fuzzy Logic* dikelompokkan dalam tiga tahap yaitu :

1. Tahap pengaburan (*fuzzification*) yaitu perubahan atau pemetaan dari masukan tegas menjadi himpunan kabur.
2. Tahap inferensi yaitu untuk membangkitkan aturan – aturan yang ada.
3. Tahap penegasan (*defuzzification*) yaitu transformasi keluaran dari nilai kabur ke nilai tegas.



Gambar 0.6 Tahapan Proses dalam Logika Kabur

### 2.3 Mencari Nilai Kesesuaian

Untuk mencari hubungan antara dua *fuzzy set*, yang berbeda dengan *universal set*  $U$ , maka diperlukan perbandingan tiap elemen dari masing-masing *fuzzy set*

tersebut. Untuk membandingkan nilai *membership degree* antara kedua *fuzzy set* tersebut dengan definisi (Sugianto,2005) :

$$R:[0,1] \times [0,1] \rightarrow [0,1] \dots\dots\dots(2.13)$$

Dapat dimisalkan bahwa U merupakan suatu *knowledge base* gejala dari suatu penyakit yang dinyatakan sebagai sebuah *fuzzy set* terhadap gejala Z dan Y adalah gejala yang diinputkan oleh *user* yang dinyatakan sebagai suatu *fuzzy set* terhadap gejala A dan X adalah gejala yang diinput oleh *user* yang dinyatakan sebagai suatu *fuzzy set* terhadap A, dimana  $A = \{a_1, a_2, a_3, a_4, \dots, a_n\}$  sedangkan  $U = \{\mu_j(a_1)/a_1, \mu_j(a_2)/a_2, \mu_j(a_3)/a_3, \mu_j(a_4)/a_4\}$  dan  $x = \{\mu_x(a_1)/a_1, \mu_x(a_2)/a_2, \mu_x(a_3)/a_3, \mu_x(a_4)/a_4\}$ . Selanjutnya dicari seberapa besarkah perbedaan nilai antara  $\mu_j$  dengan  $\mu_x$  yang merupakan nilai *fuzzy set* menurut *knowledge-based* dengan nilai *fuzzy set* gejala  $a_1$  menurut masukan dari pasien dibandingkan dengan nilai  $\mu_j(a_1)$ . Jadi formula untuk mencari nilai kesesuaian tersebut adalah (Sugianto,2005) :

$$R(X(a_i), U_j(a_i)) = \text{MAX}(0, 1 - \frac{c|\mu_x(a_i) - \mu_j(a_i)|}{\mu_j(a_i)}) \text{ dengan syarat } \mu_j(a_i) \neq 0 \dots\dots\dots(2.14)$$

Keterangan rumus :

$a_i$  = gejala yang ke-i dari *set* A

$i = 1, 2, 3, \dots, k$  dimana  $k =$  banyaknya gejala yang diinputkan *user*

$j = 1, 2, 3, \dots, l$  dimana  $l =$  banyaknya penyakit pada *knowledge-based*.

$C =$  konstanta yang dimasukkan oleh pembuat aplikasi dimana  $C$  terletak di antara interval  $0 < C \leq 1$ . Semakin  $C$  mendekati nilai 1, maka penilaian kesesuaian antara elemen *fuzzy set* yang satu dengan yang lain semakin baik

$A =$  *universal set* dari gejala di *knowledge-based*

$U =$  *fuzzy set knowledge-based* gejala terhadap  $A$  dari suatu penyakit

$\mu_x(a_i) =$  nilai *membership degree attribute* ke- $i$  dari *fuzzy set*  $A$ . Jika pada sistem ini  $A$  berisi *membership degree* dari gejala yang dimasukkan *user*.

$\mu_j(a_i) =$  nilai *membership degree attribute* ke- $i$  dari *fuzzy set*  $U$ . Jika pada pembuatan sistem ini dibaca nilai *membership degree* gejala ke- $i$  dari penyakit ke- $j$ .

$|\mu_x(a_i) - \mu_j(a_i)| =$  merupakan nilai mutlak dari selisih antara nilai *fuzzy set*.

## 2.4 Fuzzy Conditional Probability

Dari perhitungan nilai kesesuaian yang didapat antara gejala yang berasal dari *user* dengan gejala yang ada di *knowledge-based*, maka tahapan selanjutnya adalah melakukan penjumlahan nilai kesesuaian untuk penyakit, dan mencari nilai *Fuzzy Conditional Probability* untuk penyakit dengan banyaknya gejala yang dimiliki penyakit pada *knowledge-based*. Berikut merupakan rumus mencari nilai *Fuzzy Conditional Probability* (Mukaidono, 2002) :

$$P(X, U_n) = \sum_{k=1}^n \frac{R(X(ai), U_j)}{|U_n|a} \quad \text{dengan syarat } \mu_{U_n} > 0 |U_n| a \neq 0 \dots \dots \dots (2.15)$$

Keterangan rumus :

$\sum_{k=1}^n R(X(ai), U_j)$  = hasil penjumlahan dari kesesuaian antara X dengan gejala dari penyakit ke j

$|U_n|a$  = pada aplikasi ini menyatakan banyaknya gejala yang diderita oleh penyakit Un.

Dimana a dinyatakan gejala dan Un sebagai penyakit yang ke-n. Yang termasuk Un adalah nilai sampai  $a_8$  yang tidak bernilai 0

$i = 1, 2, 3, \dots, k$  dimana k = banyak gejala yang diinputkan oleh *user*

## 2.5 Demam Berdarah

Demam berdarah disebabkan oleh virus yang bernama virus dengue. Virus ini penyebab penyakit demam berdarah dan karena itulah penyakit ini diberi nama demam berdarah *dengue* yang lebih sering disingkat menjadi DBD. Terdapat empat jenis virus demam berdarah yang telah ditemukan. Oleh karena itu, pada beberapa kasus demam berdarah terdapat perbedaan gejala yang dialami penderita demam berdarah lainnya. Penyakit demam berdarah ini menyebar atau menular dari satu penderita ke penderita lainnya melalui nyamuk *aedes aegypti* atau *aedes albopictus betina*. Nyamuk ini tampak dan menggigit pada siang hari.

Demam berdarah adalah penyakit demam akut yang disebabkan oleh virus *dengue*, yang masuk ke peredaran darah manusia melalui gigitan nyamuk dari *genus Aedes*, misalnya *Aedes aegypti* atau *Aedes albopictus*. (Kristina,2004). Penyakit demam berdarah ditemukan di daerah tropis dan *subtropics* di berbagai belahan dunia, terutama pada musim hujan yang lembab. Organisasi Kesehatan Dunia (WHO) memperkirakan setiap tahunnya terdapat 50–100 juta kasus infeksi *dengue* di seluruh dunia. Terdapat juga empat jenis virus *dengue* yang berbeda, tetapi berelasi dekat yang dapat menyebabkan demam berdarah. Empat jenis tersebut adalah DEN-1, DEN-2, DEN-3, dan DEN-4. Gejala demam berdarah baru muncul saat seseorang yang pernah terinfeksi oleh salah satu dari empat jenis virus *dengue* mengalami infeksi oleh jenis virus *dengue* yang berbeda. Ini disebabkan oleh sistem imun yang sudah terbentuk di dalam tubuh setelah infeksi pertama justru akan mengakibatkan kemunculan gejala penyakit yang lebih parah saat terinfeksi untuk kedua kalinya.

Terdapat dua cara penyebaran virus *dengue* ini yang pertama adalah virus ditularkan dari nyamuk betina ke telurnya, yang nantinya akan menjadi nyamuk. Virus juga dapat ditularkan dari nyamuk jantan ke nyamuk betina melalui hubungan *sexual*. Cara kedua adalah virus ini masuk ke dalam tubuh manusia. Nyamuk yang menggigit manusia yang sudah terdapat virus *dengue* akan masuk sampai ke dalam lambung nyamuk dan mengalami replikasi atau memecah diri untuk berkembang biak kemudian akan pindah sampai di kelenjar ludah nyamuk. Penularan dari manusia kepada nyamuk dapat terjadi bila nyamuk menggigit manusia yang sedang mengalami *viremia* yaitu

dua hari sebelum panas sampai 5 hari setelah demam timbul. Pada tubuh manusia, virus memerlukan waktu masa tumbuh atau berkembang empat sampai enam hari sebelum menimbulkan penyakit. Replikasi virus *dengue* terjadi juga dalam limfosit yang bertransformasi dengan akibat terdapatnya virus dalam jumlah banyak. Hal ini mengakibatkan terbentuknya virus kompleks antigen–antibody yang selanjutnya akan mengakibatkan aktivasi sistem komplemen pelepasan C3a dan C5a akibat aktivasi C3 dan C5 menyebabkan peningkatan *permeabilitas* dinding pembuluh darah. *Patofisiologi* utama DBD atau DSS adalah kebocoran plasma yang disebabkan oleh peningkatan *permeabilitas* pembuluh darah, sehingga perpindahan plasma ke dalam ruang *ekstravaskuler*. Jika tidak ditangani lebih lanjut maka dapat terjadi komplikasi lebih parah yang dapat berujung pada kematian. Terdapat dua perubahan *patofisiologis* utama yang terjadi pada DBD, yang pertama adalah peningkatan permeabilitas pembuluh darah yang meningkatkan hilangnya *plasma* dari *kompartmenten vaskular*. Situasi ini mengakibatkan *hemokonsentrasi*, tekanan nadi rendah, dan tanda–tanda lain dari syok. Perubahan kedua adalah gangguan yang mencakup perubahan dalam *hemostatis vaskular*, *trombositopenia*. Kerusakan trombosit terjadi dalam kualitatif dan kuantitatif, jumlah trombosit selama fase akut DBD dapat habis.

Demam berdarah sendiri dapat dibagi menjadi dalam empat derajat :

1. Derajat satu

Demam disertai dengan gejala-gejala yang tidak khas untuk demam berdarah dan satu-satunya manifestasi adalah dengan melakukan uji bendung.

2. Derajat dua

Memiliki gejala seperti derajat satu tetap disertai pendarahan spontan dikulit dan atau pendarahan lain.

3. Derajat tiga

Memiliki gejala nadi menurun (20 mmHg atau kurang) atau hipotensi, sianosis di sekitar mulut, kulit dingin dan lembab.

4. Derajat empat

Sudah terjadi syok berat, nadi sudah tidak dapat diraba dan tekanan darah tidak terukur.

UMMN