

## BAB II

### LANDASAN TEORI

#### 2.1 Logika *Fuzzy*

Logika *Fuzzy* pertama kali dikembangkan oleh Lotfi A. Zadeh pada tahun 1965. Teori ini banyak diterapkan di berbagai bidang, antara lain merepresentasikan pikiran manusia kedalam suatu sistem. “Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*” (Kusuma Dewi, 2004). Banyak alasan mengapa penggunaan logika *fuzzy* ini sering dipergunakan antara lain, konsep logika *fuzzy* yang mirip dengan konsep berpikir manusia. Sistem *fuzzy* dapat merepresentasikan pengetahuan manusia ke dalam bentuk matematis dengan lebih menyerupai cara berpikir manusia. Pengontrol dengan logika *fuzzy* mempunyai kelebihan yaitu dapat mengontrol sistem yang kompleks, *non-linier*, atau sistem yang sulit direpresentasikan kedalam bentuk matematis. Selain itu, informasi berupa pengetahuan dan pengalaman mempunyai peranan penting dalam mengenali perilaku sistem di dunia nyata.

Logika *fuzzy* juga memiliki himpunan *fuzzy* yang mana pada dasarnya, teori himpunan *fuzzy* merupakan perluasan dari teori himpunan klasik. Dimana dengan logika *fuzzy*, hasil yang keluar tidak akan selalu konstan dengan input yang ada. Cara kerja logika *fuzzy* secara garis besar terdiri dari *input*, proses dan *output*. Logika *fuzzy* merupakan suatu teori himpunan logika yang dikembangkan untuk mengatasi konsep nilai yang terdapat diantara kebenaran (*true*) dan

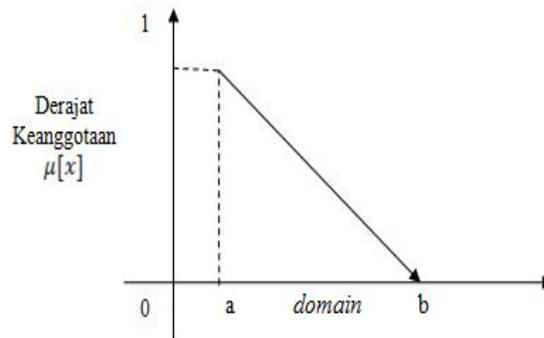
kesalahan (*false*). Dengan menggunakan *fuzzy logic* nilai yang dihasilkan bukan hanya “ya” (1) atau “tidak” (0) tetapi seluruh kemungkinan diantara 0 dan 1.

## 2.2 Komponen Fuzzy

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* (Kusuma Dewi, 2003), yaitu :

1. Variabel *fuzzy* yaitu variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contohnya: umur, temperatur, permintaan.
2. Himpunan *fuzzy* yaitu suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Himpunan *Fuzzy* memiliki 2 atribut (Kusuma Dewi, 2003), yaitu :
  - a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA.
  - b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50.
3. Semesta Pembicaraan yaitu keseluruhan nilai yang diperoleh untuk dioperasikan dalam suatu variabel *fuzzy*, semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri kekanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contohnya semesta pembicaraan untuk variabel suhu :  $[0 \ 40]$ .

4. Domain adalah keseluruhan nilai yang diijinkan dalam semesta pembicara dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan seperti pada gambar berikut.



Gambar 2.1 Domain himpunan fuzzy (Sumber: Kusuma Dewi, 2003)

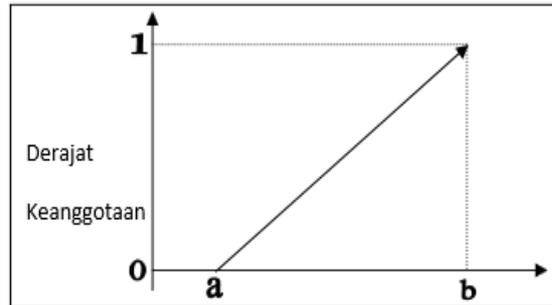
### 2.3 Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan merupakan suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaanya (disebut juga dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Untuk mendapatkan nilai keanggotaan dapat menggunakan cara pendekatan fungsi. Ada beberapa fungsi keanggotaan yang digunakan dalam teori himpunan fuzzy adalah :

1. Representasi Linier

Pada representasi linear, 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. Ada 2 keadaan himpunan fuzzy yang linear. Pertama, 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 seperti pada Gambar 2.2.

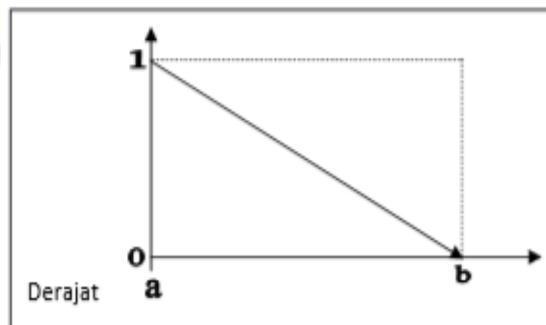


Gambar 2.2 Representasi Linear Naik (Sumber: Kusuma Dewi, 2003)

Persamaan fungsi keanggotaan linear naik:

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

Kedua, merupakan kebalikan yang pertama. 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 seperti pada Gambar 2.3.



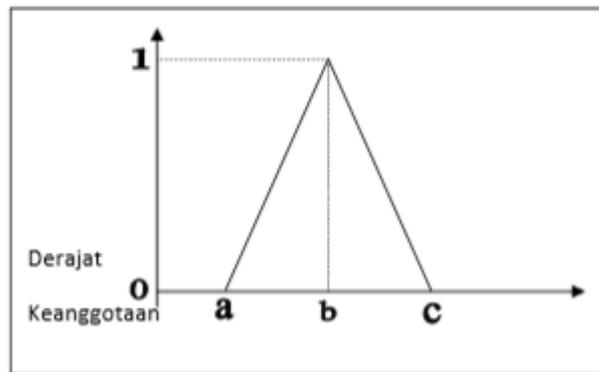
Gambar 2.3 Representasi Linear Turun (Sumber: Kusuma Dewi, 2003)

Fungsi keanggotaan linear turun :

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

## 2. Representasi Kurva Segitiga

Pada dasarnya merupakan gabungan antara 2 garis (linear), dimana nilai domain diantara a dan b atau diantara b dan c, seperti terlihat pada Gambar 2.4



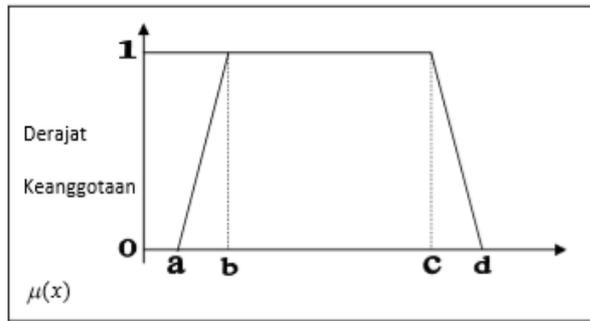
Gambar 2.4 Representasi Kurva Segitiga (Sumber: Kusuma Dewi, 2003)

Persamaan fungsi keanggotaan kurva segitiga:

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

## 3. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 seperti pada Gambar 2.5



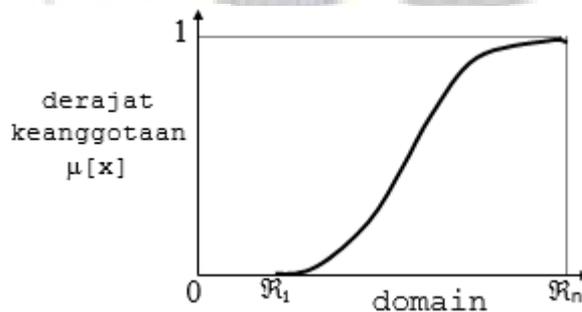
Gambar 2.5 Representasi Kurva Trapesium (Sumber: Kusuma Dewi, 2003)

Persamaan fungsi keanggotaan Kurva Trapesium :

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

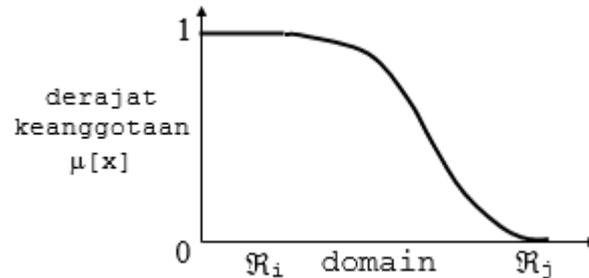
#### 4. Representasi Kurva-S

Kurva-S (*Sigmoid*) ada dua jenis yaitu kurva pertumbuhan dan penyusutan yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linear. Kurva-S untuk pertumbuhan akan bergerak dari sisi paling kiri (nilai keanggotaan = 0) ke sisi paling kanan (nilai keanggotaan = 1). Fungsi keanggotaannya akan tertumpu pada 50% nilai keanggotaannya yang sering disebut dengan titik infleksi (Gambar 2.6).



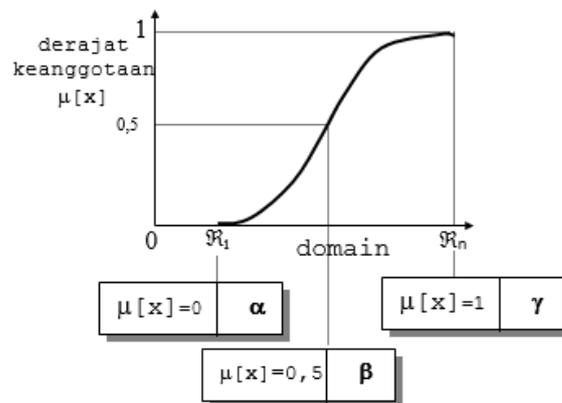
Gambar 2.6 Representasi Kurva-S pertumbuhan (Sumber: Kusuma Dewi, 2003)

Kurva-S untuk penyusutan akan bergerak dari sisi paling kanan (nilai keanggotaan = 1) ke sisi paling kiri (nilai keanggotaan = 0) seperti terlihat pada Gambar 2.7.



Gambar 2.7 Kurva-S penyusutan (Sumber: Kusuma Dewi, 2003)

Kurva-S didefinisikan dengan menggunakan 3 parameter, yaitu: nilai keanggotaan nol ( $\alpha$ ), nilai keanggotaan lengkap ( $\gamma$ ), dan titik infleksi atau crossover ( $\beta$ ) yaitu titik yang memiliki domain 50% benar. Gambar 2.8 menunjukkan karakteristik kurva-S dalam bentuk skema.



Gambar 2.8 Representasi Kurva-S (Sumber: Kusuma Dewi, 2003)

Persamaan fungsi keanggotaan Kurva-S untuk pertumbuhan :

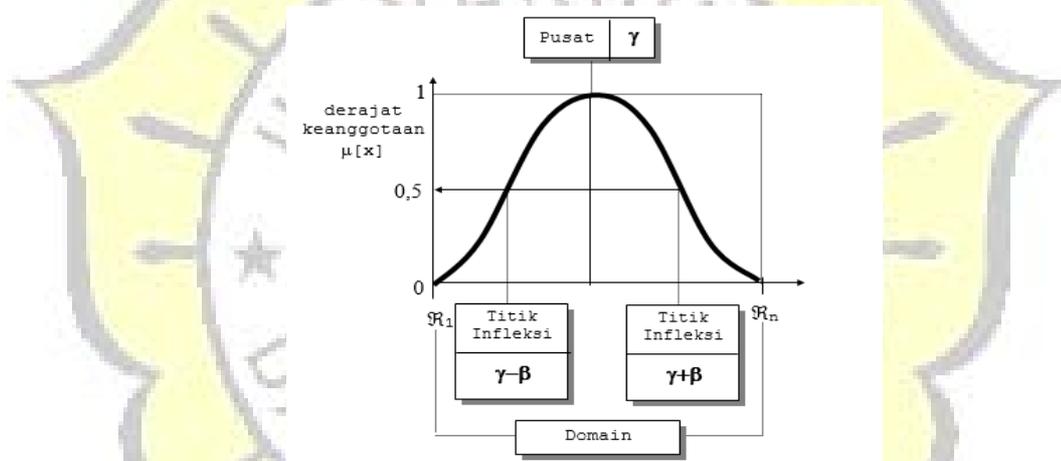
$$\mu(x; \alpha, \beta, \gamma) = \begin{cases} 0 & ; x \leq \alpha \\ 2((x - \alpha)/(\gamma - \alpha))^2 & ; \alpha \leq x \leq \beta \\ 1 - 2((\gamma - x)/(\gamma - \alpha))^2 & ; \beta \leq x \leq \gamma \\ 1 & ; x \geq \gamma \end{cases}$$

Sedangkan fungsi keanggotaan untuk penyusutan adalah :

$$\mu(x; \alpha, \beta, \gamma) = \begin{cases} 1 & ; x \leq \alpha \\ 1 - 2((x - \alpha)/(\gamma - \alpha))^2 & ; \alpha \leq x \leq \beta \\ 2((\gamma - x)/(\gamma - \alpha))^2 & ; \beta \leq x \leq \gamma \\ 0 & ; x \geq \gamma \end{cases}$$

### 5. Reperesentasi Kurva Beta

Seperti halnya kurva PI, kurva BETA juga berbentuk lonceng namun lebih rapat. Kurva ini juga didefinisikan dengan 2 parameter, yaitu nilai pada domain yang menunjukkan pusat kurva ( $\gamma$ ), dan setengah lebar kurva ( $\beta$ ) seperti terlihat pada Gambar 2.9. Nilai kurva untuk suatu nilai domain x diberikan sebagai:



Gambar 2.9 Kurva Beta (Sumber: Kusuma Dewi, 2003)

Persamaan fungsi keanggotaan kurva beta :

$$\mu(x; \gamma, \beta) = \frac{1}{1 + \left(\frac{x - \gamma}{\beta}\right)^2}$$

## 2.4 Jenis Metode *Fuzzy*

Ada tiga metode dalam sistem inferensi fuzzy yang dapat digunakan untuk menentukan jumlah produksi, yaitu: metode Tsukamoto, metode Mamdani, dan metode Sugeno (Setiadji, 2009). Penjelasan mengenai ketiga metode tersebut adalah sebagai berikut:

### 1. Metode Tsukamoto

Pada metode Tsukamoto, setiap aturan direpresentasikan menggunakan himpunan-himpunan fuzzy, dengan fungsi keanggotaan yang monoton. Untuk menentukan nilai *output crisp*/ hasil yang tegas ( $Z$ ) dicari dengan cara mengubah input (berupa himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy) menjadi suatu bilangan pada domain himpunan fuzzy tersebut. Cara ini disebut dengan metode defuzzifikasi (penegasan). Metode defuzzifikasi yang digunakan dalam metode Tsukamoto adalah metode defuzzifikasi rata-rata terpusat (Center Average Defuzzifier).

### 2. Metode Mamdani (Min-Max)

Untuk metode ini, pada setiap aturan yang berbentuk implikasi (“sebab-akibat”) anteseden yang berbentuk konjungsi (AND) mempunyai nilai keanggotaan berbentuk minimum (min), sedangkan konsekuen gabungannya berbentuk maksimum (max), karena himpunan aturan-aturannya bersifat independen (tidak saling bergantung).

### 3. Metode Takagi-Sugeno

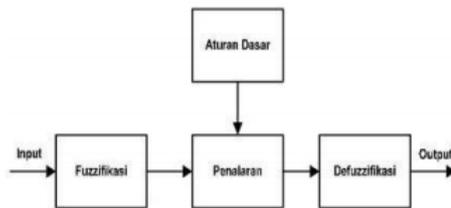
Fuzzy metode sugeno merupakan metode inferensi fuzzy untuk aturan yang direpresentasikan dalam bentuk IF – THEN, dimana output (konsekuen) sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear (KUS : 2002). Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Model Sugeno menggunakan fungsi keanggotaan Singleton yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai crisp tunggal dan 0 pada nilai crisp yang lain. Untuk Orde 0 dengan rumus : IF (x1 is A1) ° (x2 is A2) °...°(xn is An) THEN z= k, dengan Ai adalah himpunan fuzzy ke i sebagai antaseden (alasan), ° adalah operator fuzzy (AND atau OR) dan k merupakan konstanta tegas sebagai konsekuen (kesimpulan). Sedangkan rumus Orde 1 adalah: IF (x1 is A1) ° (x2 is A2) °...°(xn is An) THEN z = p1\*x1+...+pn\*xn+q, dengan Ai adalah himpunan fuzzy ke i sebagai antaseden, ° adalah operator fuzzy (AND atau OR), pi adalah konstanta ke i dan q juga merupakan konstanta dalam konsekuen.

## 2.5 Cara Kerja Logika Fuzzy

Dalam sistem logika fuzzy terdapat beberapa tahapan operasional meliputi:

1. Fuzzifikasi.
2. mesin penalaran atau inference engine.
3. aturan dasar (fuzzy rule).
4. defuzzifikasi.

Berikut blok diagram logika fuzzy ditunjukkan pada gambar 2.10.

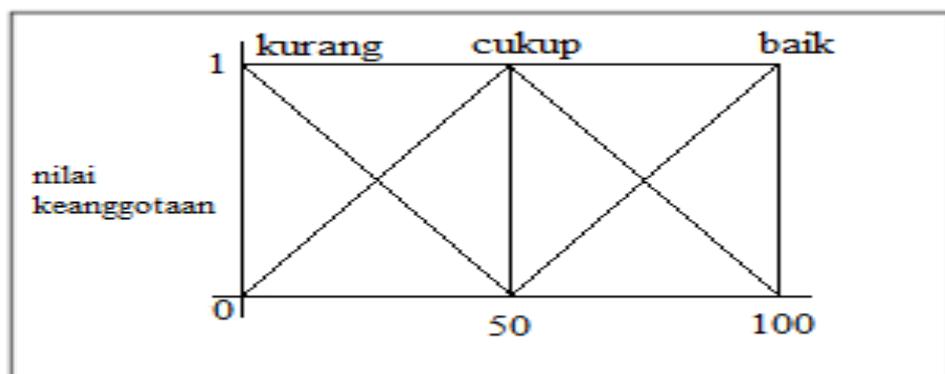


Gambar 2.10 Blok diagram logika fuzzy (Sumber: NUG, 2010)

Berdasarkan gambar 2.10 dalam *system* logika fuzzy terdapat beberapa tahapan operasional yang meliputi [NUG : 2010] :

1. Fuzzifikasi

Fuzzifikasi adalah suatu proses pengubahan nilai tegas yang ada ke dalam fungsi keanggotaan. Dalam hal ini fungsi keanggotaan yang digunakan adalah kurva segitiga. Dan berikut penjelasan fuzzifikasi pada input dan output :



Gambar 2.11 Fuzifikasi Pada Input output

a. Nilai Keanggotaan Kurang

$$\mu(x) = \begin{cases} 1; & x \leq 0 \\ \frac{50 - x}{50 - 0}; & 0 < x < 50 \\ 0; & x \geq 50 \end{cases}$$

b. Nilai Keanggotaan Cukup

$$\mu(x) = \begin{cases} 0; & x \leq 0 \text{ atau } x \geq 100 \\ \frac{x - 0}{50 - 0}; & 0 \leq x \leq 50 \\ \frac{100 - x}{100 - 50}; & 50 \leq x \leq 100 \end{cases}$$

c. Nilai Keanggotaan Baik

$$\mu(x) = \begin{cases} 1; & x \leq 50 \\ \frac{x - 50}{100 - 50}; & 50 < x < 100 \\ 0; & x \geq 100 \end{cases}$$

2. Penalaran (*Inference Machine*)

Mesin penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluaran sebagai bentuk pengambilan keputusan. Salah satu model penalaran yang banyak dipakai adalah penalaran max-min. Dalam penalaran ini, proses pertama yang dilakukan adalah melakukan operasi min sinyal keluaran lapisan fuzzifikasi, yang diteruskan dengan operasi max untuk mencari nilai keluaran yang selanjutnya akan didefuzzifikasikan sebagai bentuk keluaran.

3. Aturan Dasar (*Rule Based*)

Aturan dasar (rule based) pada logika fuzzy merupakan suatu bentuk aturan relasi “Jika-Maka” atau “*if-then*” seperti berikut ini: if x is A then y is B dimana A dan B adalah linguistic values yang didefinisikan dalam rentang variabel X dan Y. Pernyataan “x is A” disebut *antecedent* atau premis. Pernyataan “y is B” disebut *consequent* atau kesimpulan.

Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy yaitu : Max, Additive dan Probabilistik OR :

a. Metode Max (Maximum)

Pada metode ini solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke output dengan menggunakan operator OR(union). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan fuzzy yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] \leftarrow \max (\mu_{sf}[x_i] , \mu_{kf}[x_i])$$

dengan :

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

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

b. Metode Additive (Sum)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan *bounded-sum* terhadap semua output daerah fuzzy. Secara umum dituliskan :

$$\mu_{sf}[x_i] \leftarrow \max ( 1, \mu_{sf}[x_i] + \mu_{kf}[x_i] )$$

dimana :

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

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

#### c. Metode Probabilistik OR

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan product terhadap semua output daerah fuzzy. Secara umum dituliskan :

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

dimana :

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

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

#### 4. 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. Ada beberapa **metode defuzzifikasi** pada komposisi aturan MAMDANI (Kusuma Dewi, 2003), antara lain:

1. Metode *Centroid* (Composite Moment).

Pada metode centroid solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy*. Secara umum dapat dituliskan :

a. Untuk semesta kontinu

$$Z^* = \frac{\int_a^n z\mu(z)dz}{\int_a^n \mu(z)dz}$$

b. Untuk semesta diskrit

$$Z^* = \frac{\sum_{j=1}^n z_j\mu(z_j)}{\sum_{j=1}^n \mu(z_j)}$$

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 berikut :

$$\int_{\alpha}^z \mu(z) dz = \int_z^{\beta} \mu(z) dz$$

Dimana

:

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

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



## 2.6 HTML

HTML kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah file teks murni yang dapat di buat dengan *editor* teks sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam *browser web server*. Dokumen ini umumnya berisi informasi atau *interface* aplikasi di dalam internet. Ada dua cara untuk membuat sebuah *web page* yaitu dengan HTML *editor* atau dengan *editor* teks biasa seperti notepad.

Dokumen HTML disusun oleh elemen-elemen. “Elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen seperti *head*, *body*, *table*, paragraf, dan list. Elemen dapat berupa teks murni, atau bukan teks, atau keduanya.(Abdul Kadir, 2008)

## 2.7 CSS

CSS merupakan singkatan dari *Cascading Style Sheet*, merupakan fitur baru dari HTML 4.0 hal ini diperlukan setelah melihat perkembangan HTML menjadi kurang praktis karena *web pages* terlalu banyak dibebani hal-hal yang berkaitan dengan faktor tampilan seperti font dan lain-lain. Untuk itu jika kumpulan isi gaya (*style*) tersebut dikelola secara terpisah maka manajemen *pages* menjadi lebih mudah dan efisien. (Bambang Hariyanto, 2008)

## 2.8 PHP

PHP merupakan salah satu bahasa pemrograman berbentuk skrip yang ditempatkan dalam server dan diproses di server (Budi Raharjo, 2011). Hasilnya lah yang dikirim ke *client*, tempat pemakai menggunakan *browser*. Secara khusus,

PHP dirancang untuk membentuk aplikasi *web* dinamis. Maksudnya, PHP mampu menghasilkan *website* yang secara terus-menerus hasilnya bisa berubah-ubah sesuai dengan pola yang diberikan. Hal tersebut tergantung pada permintaan *client browse*-nya (bisa menggunakan *browser* Opera, Internet Explorer, Mozilla, dan lain-lain). Umumnya, pembuatan *web* dinamis berhubungan erat dengan *Database* sebagai sumber data.

PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *Command Line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Model kerja HTML diawali dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web browser*, mengidentifikasi alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari *file* yang diminta dan memberikan isinya ke *web browser* (atau yang biasa disebut *browser* saja). *Browser* yang mendapatkan isinya segera melakukan proses penerjemah kode HTML dan menampilkan ke layar pemakai.

## **2.9 Basisdata**

Data adalah rekaman mengenai fenomena/ fakta yang ada atau yang terjadi (Bambang Hariyanto, 2008). Data pada pokoknya adalah refleksi fakta yang ada. Data mengenai fakta-fakta penting organisasi harus direkam dan dikelola secara baik sehingga dapat dipakai/ diakses secara efisien sehingga efektif mendukung

operasi dan pengendalian organisasi. Data merupakan sumber daya penting pada manajemen modern. Untuk itu, organisasi perlu melakukan penataan dan manajemen data yang baik agar data yang dimiliki organisasi dapat berdaya guna secara maksimal.

Basisdata adalah kumpulan data (elementer) yang secara logic berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu (Bambang Hariyanto, 2008). Basisdata adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Basisdata mendeskripsikan *state* organisasi/perusahaan/sistem. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahan pun harus dilakukan terhadap data yang disimpan di basisdata. Basisdata merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di basisdata. Pengelolaan basisdata yang buruk dapat mengakibatkan ketidakterediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

## **2.10 MySQL**

Menurut Abdul Kadir (2008) MySQL (baca: mai-se-kyu-el) merupakan software yang tergolong sebagai DBMS (Database Management System) yang bersifat Open Source. Open Source menyatakan bahwa software ini dilengkapi dengan source code (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk sistem operasi, dan bisa diperoleh dengan cara mendownload (mengunduh) di Internet secara gratis. MySQL adalah salah satu jenis *database*

*server* yang sangat terkenal kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*.)

MySQL dalam operasi klien-server melibatkan server daemon MySQL di sisi server dan berbagai macam program serta pustaka yang berjalan di sisi klien. MySQL mampu menangani data yang cukup besar, Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 database, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data.

### **2.11 *Unified Modeling Language***

Menurut Munawar (2005) *Unified Modeling Language* (UML) adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek, UML menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Menurut Romi Satria Wahono dan Sri Dharyanti *Unified Modelling Language* merupakan sebuah bahasa yang menjadi bahan standar dalam dunia industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML menyediakan beberapa notasi dan *artifact* standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. *Artifact* didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan untuk atau dihasilkan dalam proses pengembangan perangkat.

Ada tiga modeling yang terdapat dalam UML, yaitu:

- a. *Function Modelling* merupakan diagram-diagram yang menyatakan bagaimana suatu sistem itu bekerja dan yang termasuk dalam *function modelling* adalah *usecase description*, *usecase diagram*, dan *activity diagram*.
- b. *Struktural Modelling* terdiri dari beberapa diagram yang memberikan *snapshot* dari elemen-elemen yang berperilaku. *Structural Modelling* mempunyai beberapa model diantaranya *Class Diagram*.
- c. *Behavior Modelling* mendefinisikan bagaimana suatu objek bertindak dan berinteraksi, dan berhubungan dengan fungsi diterapkan pada suatu atribut. Mempunyai beberapa model yaitu *sequence diagram*.

### **2.11.1 Function Modelling**

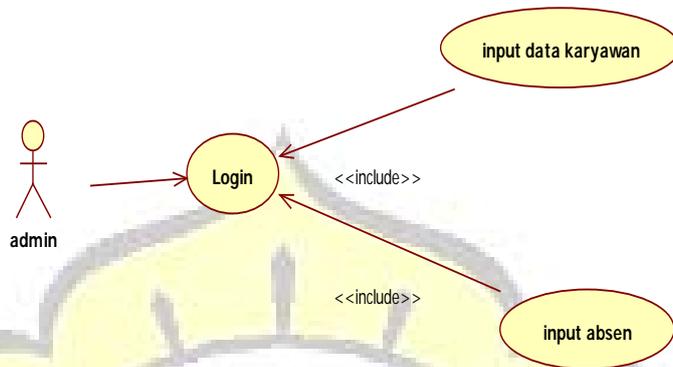
#### **a. Use Case Description**

*Usecase* Deskripsi merupakan tekstual tentang kejadian dan bagaimana pengguna berinteraksi dengan sistem, *Usecase* deskripsi juga berfungsi sebagai ilustrasi narasi dalam menjelaskan permasalahan

#### **b. Use Case Diagram**

Munawar (2005) menjelaskan Use case diagram merupakan deskripsi fungsi dari sebuah sistem perspektif pengguna. *Usecase* bekerja dengan mendeskripsikan fungsi tipikal interaksi antar *user* (pengguna) sebuah sistem dengan sistem-nya sendiri bagaimana sistem tersebut

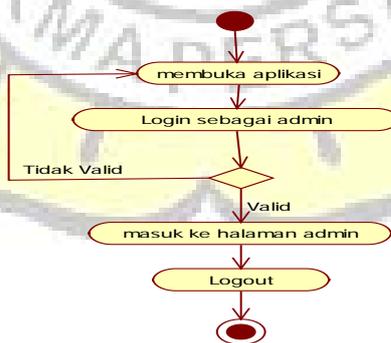
digunakan atau “apa” yang diperbuat sistem dan bukan “bagaimana”.  
*Usecase* dapat dipresentasikan sebuah interaksi antar *actor* dengan sistem.  
 Contoh *use case diagram* sebagai berikut.



Gambar 2.12 Contoh *use case diagram* (Sumber: Munawar,2005)

**c. Activity Diagram**

Bersifat dinamis. Diagram activity adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Menekankan pada aliran kendali antar objek.



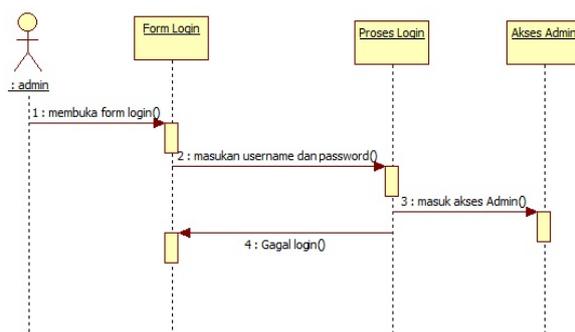
Gambar 2.13 Contoh *Activity Diagram* (Sumber: Munawar, 2005)

## 2.11.2 Behavior Modeling

### a. Sequence Diagram

*Sequence diagram* dilihat dari kacamata Romi Satria Wahono dan Sri Dharyanti menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

Munawar mengatakan *sequence diagram* biasa digunakan untuk menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.



Gambar 2.14 Contoh *sequence diagram* (Sumber: Munawar, 2005)