

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Game dan Video Game

Menurut kamus *Cambridge Advanced Learner Dictionary*, *game* adalah sebuah aktivitas menghibur dan menyenangkan yang dimainkan oleh anak – anak. Sedangkan *video game* adalah *game* yang *player*-nya mengontrol gambar bergerak dengan menekan tombol pada *joystick*.

Menurut Pedersen, Roger E. (2003), Sebuah *game* memiliki beberapa konsep penting seperti:

- *game* tidak bersifat linier seperti buku dan film
- *game* mempunyai tujuan (*goal*)
- *game* harus bisa dimenangkan oleh *player*
- *game* memiliki awal (*start*) dan akhir (*ending*)

#### 2.2 Game Genre dan Real Time Strategy

Sebagaimana dikemukakan oleh Rabin, Steve (2010), banyak dari *game* modern bisa dikategorikan kedalam *genre* tertentu, atau gabungan dari dua atau lebih *genre* tertentu. *Genre* ini muncul setelah evolusi pada pengembangan *game*. Kemudian terdapat sebuah *genre Real-Time Strategy* pada keragaman *genre game*. *Genre game* ini mempunyai tujuan (*goal*) agar *player* mengumpulkan dan mengelola *resource*, membangun armada perang, dan mengontrol unit yang telah dimiliki untuk menyerang musuh. *Action* pada *game* ini sangatlah cepat, dan karena *genre* ini bersifat kontinyu, keputusan strategis harus dibuat secepat mungkin. Contoh *game* yang populer dari *genre* ini antara lain: *Command and*

*Conquer dan World of Warcraft.*

Ridwan Fadjar Septian, 2014

**IMPLEMENTASI FUZZY LOGIC METODE MAMDANI UNTUK PENGEMBANGAN INTELLIGENT NON-PLAYER CHARACTER PADA GAME STRATEGY**

Universitas Pendidikan Indonesia | [repository.upi.edu](http://repository.upi.edu) | [perpustakaan.upi.edu](http://perpustakaan.upi.edu)

### 2.3 Non Player Character

Umumnya tujuan pengembangan AI pada game adalah menciptakan agen cerdas yang sering juga disebut sebagai *nonplayer character (NPC)*. Agen ini berperilaku sebagai musuh atau kawan, atau dapat sebagai sosok netral di dunia *game*. Agen mempunyai tiga komponen penting yang terus diproses dalam sebuah siklus. Komponen tersebut antara lain siklus *sense-think-act*. Sebagai tambahan, terdapat juga komponen *learning* dan *remembering* yang dapat meningkatkan tantangan bermain dan kemenarikan *game* tersebut. (Rabin, Steve, 2010)

### 2.4 Reasoning

*Reasoning* adalah teknik penyelesaian masalah dengan cara merepresentasikan masalah ke dalam basis pengetahuan (*knowledge base*) menggunakan *logic* atau bahasa formal (bahasa yang dipahami komputer). *Reasoning* mempunyai beberapa contoh logika untuk merepresentasikan pengetahuan dan melakukan penalaran, diantaranya: *propositional logic*, *first order logic* atau *predicate calculus*, dan *fuzzy logic*. Dua logika pertama digunakan untuk masalah yang memiliki kepastian. Sedangkan *fuzzy logic* digunakan untuk menyelesaikan masalah yang memiliki ketidakpastian (Suyanto, 2011).

### 2.5 Fuzzy Logic

*Fuzzy logic* didefinisikan sebagai suatu jenis logika yang bernilai ganda dan berhubungan dengan ketidakpastian dan kebenaran parsial. *Fuzzy logic* teknik matematis yang didesain untuk bekerja pada area abu – abu. Tidak seperti *traditional logic* yang bekerja pada area “*true*” dan “*false*” (Millington, Ian, 2006).

**Tabel 2.1 Tabel kebenaran AND (Kusumadewi, Sri. 2003)**

A	B	A AND B
false	false	false
false	true	false
true	false	false
true	true	true

Ridwan Fadjar Septian, 20  
IMPLEMENTASI FUZZY LOGIC  
PLAYER CHARACTER PADA  
Universitas Pendidikan Indc

ANGAN INTELLIGENT NON-  
in.upi.edu

Pada *traditional logic* kita menggunakan tabel kebenaran, yang menunjukkan kepada kita seberapa benarkah sebuah pernyataan berdasarkan kemungkinan kebenaran yang ada (Millington, Ian, 2006). Sebagai contoh tabel kebenaran AND dinyatakan sebagai berikut:

Pada *fuzzy logic*, setiap operator memiliki aturan numerik yang bisa digunakan untuk menghitung derajat kebenaran berdasarkan derajat kebenaran dari setiap input. *Fuzzy rule* untuk AND adalah:

$$m_{(A \text{ AND } B)} = \min(m_A, m_B)$$

kemudian *fuzzy rule* untuk OR adalah:

$$m_{(A \text{ OR } B)} = \max(m_A, m_B)$$

sedangkan untuk NOT adalah:

$$m_{(\text{NOT } A)} = 1 - m_A$$

dari contoh diatas bisa kita dapat beberapa operator lain yang terdapat pada *fuzzy logic* seperti pada tabel berikut:

**Tabel 2.2 Operator Fuzzy Logic (Kusumadewi, Sri. 2003)**

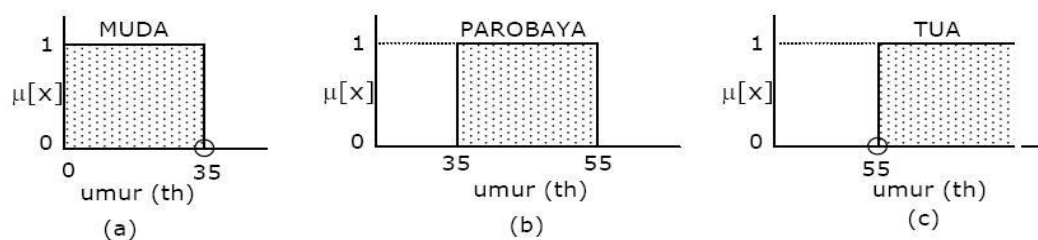
Expression	Equivalent	Fuzzy Equation
NOT A		$1 - m_A$
A AND B		$\min(m_A, m_B)$
A OR B		$\max(m_A, m_B)$
A XOR B	NOT(B) AND A	$\min(m_A, 1 - m_B)$
	NOT(A) AND B	$\min(1 - m_A, m_B)$
A NOR B	NOT(A OR B)	$1 - \max(m_A, m_B)$
A NAND B	NOT(A AND B)	$1 - \min(m_A, m_B)$

## 2.6 Fuzzy Set

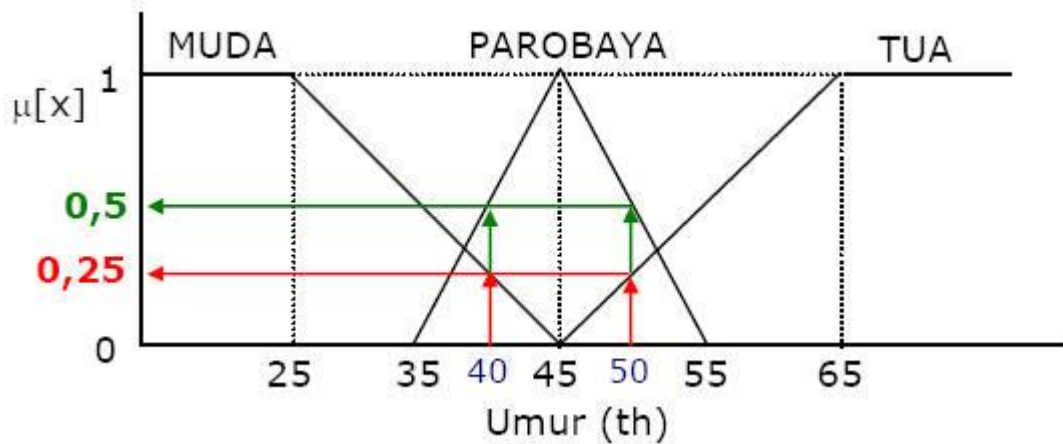
*Fuzzy Set* adalah cara menentukan seberapa baik sebuah objek memenuhi deskripsi yang tidak jelas (*vague description*). Sebagai contoh, terdapat sebuah kalimat “Nate itu tinggi”. Apakah benar jika Nate mempunyai tinggi 5’10”?, Kebanyakan orang akan ragu untuk menjawab “benar” atau “salah” (“*true*” or “*false*”), umumnya akan menjawab “kira – kira” atau “sepertinya”. Permasalahannya adalah istilah “tinggi” tidak memenuhi batas yang jelas (*sharp demarcation*) dari objek tersebut. *Fuzzy Set* menganggap “tinggi” sebagai *fuzzy predicate* dan kebenarannya dinyatakan dengan angka antara 0 dan 1, ketimbang sebagai “benar” atau “salah”. *Fuzzy Set* bisa disebut sebagai himpunan yang tidak memiliki batasan yang jelas (*sharp boundaries*) (Russel, Stuart dan Norvig, Peter, 2010).

Menurut Kusumadewi, Sri (2003) pada himpunan tegas (*crisp*), nilai keanggotaan suatu *item*  $x$  dalam suatu himpunan  $A$ , yang sering ditulis dengan  $\mu_A[x]$ , memiliki dua kemungkinan, yaitu:

- ⤴ satu (1), yang berarti bahwa suatu *item* menjadi anggota dalam suatu himpunan, atau
- ⤴ nol (0), yang berarti bahwa suatu *item* tidak menjadi anggota dalam suatu himpunan.



**Gambar 2.1 Himpunan: MUDA, PAROBAYA, dan TUA**  
(Kusumadewi, Sri. 2003)



Pada himpunan *fuzzy*, nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila  $x$  memiliki nilai keanggotaan *fuzzy*  $\mu_A[x] = 0$ , berarti  $x$  tidak menjadi anggota himpunan  $A$ . Demikian pula jika  $x$  memiliki nilai keanggotaan *fuzzy*  $\mu_A[x] = 1$ , berarti  $x$  menjadi anggota himpunan penuh  $A$ . Hal ini berbeda dengan derajat keanggotaan himpunan *crisp*. Pada himpunan *crisp* nilai keanggotaan hanya ada d

**Gambar 2.2 Himpunan fuzzy untuk variabel umur**  
(Kusumadewi, Sri. 2003)

Himpunan *fuzzy* memiliki dua atribut, yaitu:

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

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*,

yaitu:

1. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh : umur, temperatur, permintaan, dsb.

2. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

Contoh:

▲ Variabel umur, terbagi menjadi tiga himpunan *fuzzy*, yaitu: MUDA, PAROBAYA, dan TUA

▲ Variabel temperatur, terbagi menjadi 5 himpunan *fuzzy*, yaitu: DINGIN, SEJUK, NORMAL, HANGAT, dan PANAS

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contoh:

▲ Semesta pembicaraan variabel umur:  $[0 +\infty)$

▲ Semesta pembicaraan untuk variabel temperatur:  $[0 40]$

4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan 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. Nilai domain dapat berupa bilangan positif maupun negatif. Contoh domain himpunan *fuzzy*:

▲ MUDA =  $[0 45]$

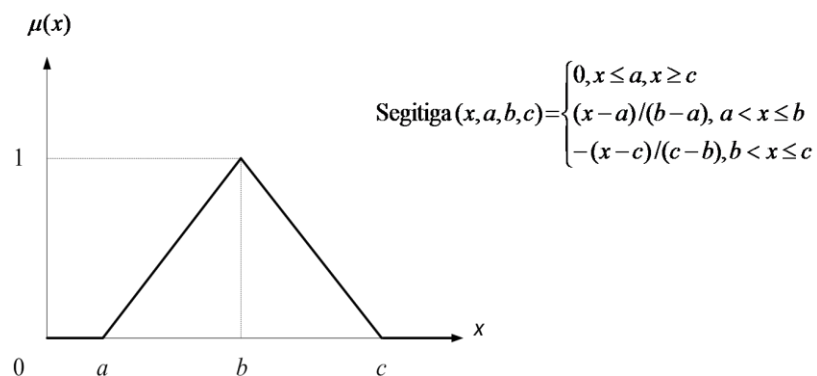
- ♣ PAROBAYA = [35 55]
- ♣ TUA = [45 tak hingga]
- ♣ DINGIN = [0 20]
- ♣ SEJUK = [15 25]
- ♣ NORMAL = [20 30]
- ♣ HANGAT = [25 35]
- ♣ PANAS = [30 40]

## 2.7 Fungsi Keanggotaan

Di dalam *fuzzy system*, fungsi keanggotaan digunakan untuk merepresentasikan masalah dan menghasilkan keputusan yang akurat. Terdapat banyak sekali fungsi keanggotaan yang bisa digunakan (Suyanto, 2011). Beberapa fungsi keanggotaan yang sering digunakan, antara lain:

### 1. Fungsi Segitiga

Pada fungsi ini terdapat nilai  $x$  yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika  $x = b$ . Tetapi, nilai – nilai di sekitar  $b$  memiliki derajat keanggotaan yang turun cukup tajam. Grafik dan notasi matematika dari fungsi ini terlihat pada gambar dibawah ini:

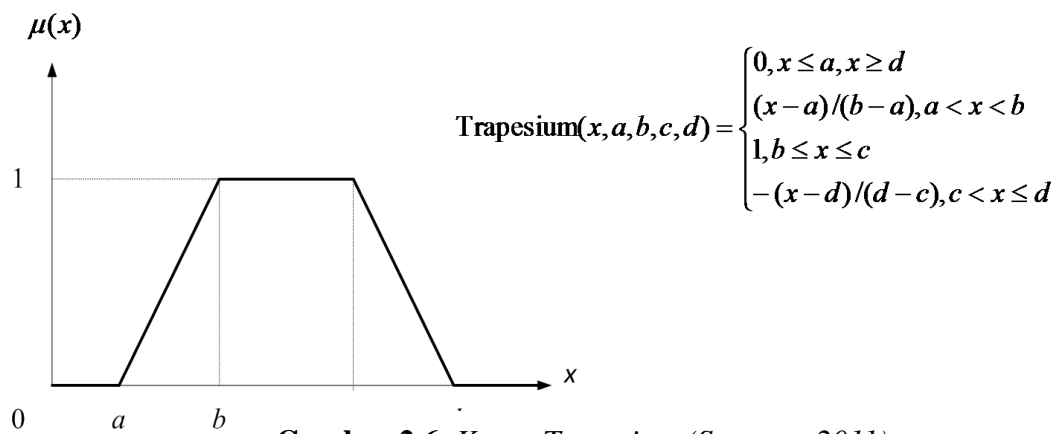


**Gambar 2.3 Kurva Segitiga (Suyanto. 2011)**



## 2. Fungsi Trapesium

Pada fungsi ini terdapat beberapa nilai  $x$  yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika  $b \leq x \leq c$ . Tetapi, derajat keanggotaan untuk  $a < x < b$  dan  $c < x \leq d$  memiliki karakteristik yang sama dengan fungsi segitiga. Grafik dan notasi matematika dari fungsi trapesium adalah sebagai berikut:

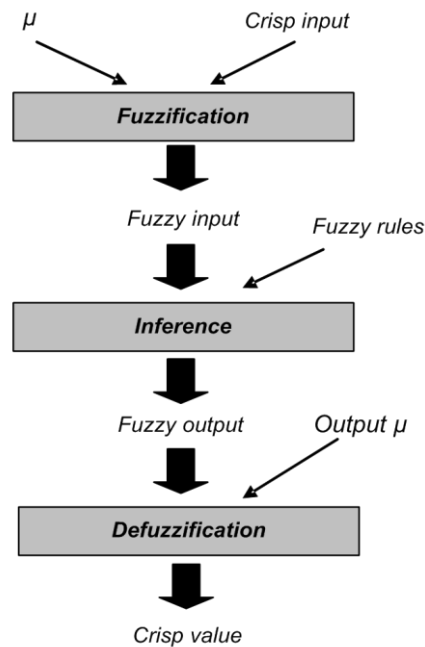


**Gambar 2.6** Kurva Trapesium (Suyanto. 2011)

## 2.8 Fuzzy System

Suatu sistem berbasis aturan *fuzzy* yang lengkap terdiri dari tiga komponen utama: *fuzzification*, *inference*, dan *defuzzification*. Coba lihat gambar dibawah ini.

**Gambar 2.4** Kurva Trapezoid (Suyanto. 2011)



**Gambar 2.5 Proses Fuzzy (Suyanto. 2011)**

*Fuzzification* mengubah masukan – masukan yang nilai kebenarannya bersifat pasti (*crisp input*) ke dalam bentuk *fuzzy input*, yang berupa nilai linguistik yang semantiknya ditentukan berdasarkan fungsi keanggotaan tertentu. *Inference* melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rule* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Sedangkan *defuzzification* mengubah *fuzzy output* menjadi *crisp value* berdasarkan fungsi keanggotaan yang telah ditentukan (Suyanto, 2011).

## 2.9 Fuzzy Logic Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan

output, diperlukan 4 tahapan: (1) Pembentukan himpunan *fuzzy*; (2) Aplikasi fungsi implikasi (aturan); (3) Komposisi aturan; (4) Penegasan (*defuzzy*) (Kusumadewi, Sri, 2003).

#### 1. Pembentukan himpunan *fuzzy*

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

#### 2. Aplikasi fungsi implikasi

Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

#### 3. Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri-dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu: *max*, *additive* dan probabilitistik OR (probor).

##### 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:

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

dengan:

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

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

b) Metode *Additive (Sum)*

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

$$\mu_{sf}[xi] \leftarrow \min(1, \mu_{sf}[xi] + \mu_{kf}[xi])$$

dengan:

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

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

c) Metode Probabilistik OR (probor)

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

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

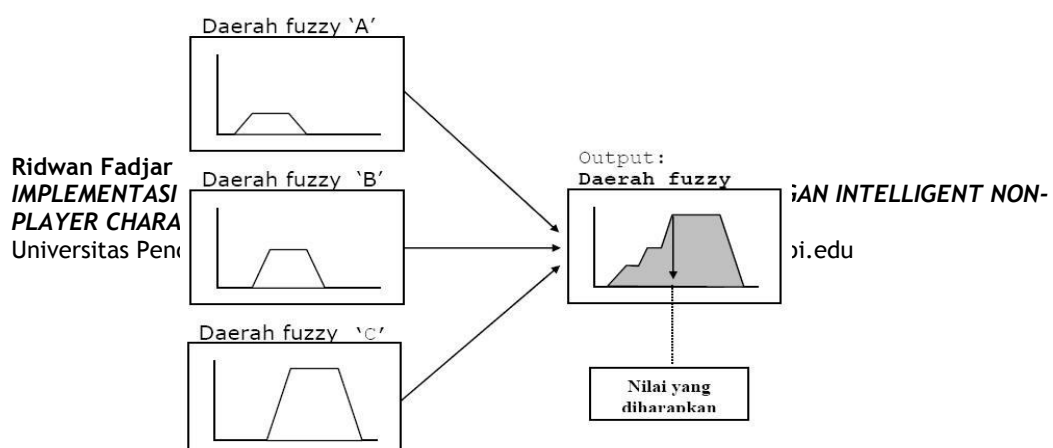
dengan:

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

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

4. Penegasan (*defuzzy*)

Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crsip* tertentu sebagai output seperti terlihat pada



gambar berikut.

Ada beberapa metode defuzzifikasi pada komposisi aturan MAMDANI (Kusumadewi, Sri, 2003), antara lain:

1. Metode *Centroid (Composite Moment)*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat ( $z^*$ ) daerah *fuzzy*. Secara umum dirumuskan:

**z Gambar 2.6 Proses Defuzzifikasi (Kusumadewi, Sri. 2003)**

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

2. Metode Bisektor

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan separuh dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:

$$z_p \text{ sedemikian hingga } \int_{\mathfrak{R}1}^p \mu(z) dz = \int_p^{\mathfrak{R}n} \mu(z) dz$$

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* (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.