

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Dalam penelitian tentang sistem kendali *take-off quadcopter* terdapat beberapa algoritma yang telah digunakan salah satunya seperti *full state backstepping* yang digunakan oleh Madani dan Benallegue (2006) untuk mengendalikan kestabilan *quadcopter*. Dengan menggunakan algoritma ini tidak ada *overshoot* pada kontrol ketinggian. Namun pada penelitian ini memiliki kelemahan yaitu waktu yang dibutuhkan *quadcopter* untuk *take-off* dan mencapai posisi *hover* adalah sebesar 11 detik. Selain algoritma *Full State backstepping* algoritma kontrol cerdas lainnya seperti jaringan saraf tiruan juga diterapkan seperti yang ditulis oleh Nicol et al (2008) untuk mengendalikan posisi ketinggian *quadcopter*. Dalam penelitiannya, *quadcopter* dapat melakukan posisi *hover* yang stabil, namun butuh waktu lama untuk mencapai posisi stabil hingga *hover*.

Algoritma kontrol cerdas lainnya adalah pengendali logika *fuzzy* yang digunakan oleh Varga dan Bogdan (2009) untuk mengendalikan posisi ketinggian *quadcopter*. Algoritma ini menggunakan *rule base fuzzy 3x3* yang terdiri dari dua input yaitu *error* dan *delta error* untuk menstabilkan *quadcopter* tanpa osilasi. Namun sistemnya cenderung lambat dengan setting time 17 detik dan ada *overshoot*. Peneliti lain juga telah mengembangkan algoritma *fuzzy* dengan mengubah *rule base* menjadi *7x7* untuk kontrol ketinggian *quadcopter* seperti yang dilakukan oleh Mehranpour et al (2013) di mana *quadcopter* mampu bergerak cepat untuk melakukan *take-off* dan *hover* namun masih terdapat *overshoot*.

Fuzzy MRAC juga diterapkan untuk menjaga kestabilan pada waktu *take-off* seperti ditulis oleh Prawirayuda (2012). Pada paper ini membahas mekanisme *take-off* menggunakan metode *fuzzy Model Reference Adaptive Control (MRAC)*, dimana yang dikontrol adalah ketinggian (*Z*) yang didapat dari sensor *ultrasonik* atau *ping*. Dengan kontroler *fuzzy Model Reference Adaptive Control (MRAC)*, *quadcopter* diharapkan dapat mencapai titik vertikal yang ditentukan, kestabilan pada sudut *roll* dan *pitch* dikontrol oleh kontroler *PD (Proportional Diferensial)*. Data yang diolah berupa *feedback* yang didapat dari *accelerometer* dan *gyroscope*. Selain menggunakan metode *fuzzy Model Reference Adaptive Control (MRAC)* ada pula yang menggunakan kontrol *fuzzy Proportional Integral Derivative controller (PID)* seperti yang dilakukan oleh Kusuma, Effendi, Iskandar (2012). Pada penelitian ini penulis menggunakan kontroler tipe *Proportional Integral Derivative controller (PID)*, dikarenakan sederhana, mudah dipelajari dan tentunya mudah diaplikasikan.

Adapula penelitian yang menggunakan *fuzzy Mamdani* seperti yang ditulis oleh Raharja et al. (2015) pada penelitian ini merupakan pengembangan dari beberapa penelitian sebelumnya yang bertujuan untuk mengoptimalkan kecepatan, meningkatkan waktu dan menghilangkan *overshoot* saat *quadcopter*

hover pada posisi ketinggian yang telah ditentukan. Pada penelitian yang dilakukan oleh Raharja et al (2015) kontrol *fuzzy* digunakan untuk mengendalikan tinggi *quadcopter* dengan *rule base 7x7*. Kontrol ketinggian dengan basis aturan *Fuzzy 7x7* adalah kontrol *fuzzy* dimana terdiri dari dua buah variabel *input* yaitu ketinggian dan perubahan ketinggian yang memiliki tujuh variabel anggota himpunan *fuzzy*. Dari hasil pengujian yang dilakukan oleh Raharja et al (2015) untuk basis aturan *7x7* di dapatkan kesimpulan bahwa algoritma *fuzzy* dengan metode *Mamdani* dapat diterapkan pada kontrol ketinggian *quadcopter*, sedangkan untuk kontrol ketinggian dengan algoritma logika *fuzzy* menggunakan *rule base 7x7* lebih baik dari *rule base 3x3* dan *5x5*. Dengan kontrol ketinggian menggunakan *rule base 7x7*, *quadcopter* dapat menjaga kesetabilan pada 6.089 detik dan waktu yang di perlukan sampai pada posisi *hover* sebesar 1,539 detik.

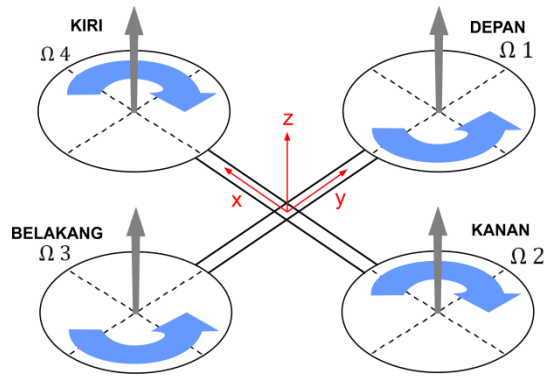
Berdasarkan beberapa penelitian yang telah dilakukan maka dapat disimpulkan bahwa algoritma *full state backstepping* mampu menghilangkan *overshoot* pada kontrol ketinggian namun kelemahan dari algoritma ini adalah waktu yang dibutuhkan untuk mencapai posisi *hover* masih lambat. Begitu pula dengan algoritma jaringan saraf tiruan, *quadcopter* bisa melakukan posisi *hover* yang stabil, namun butuh waktu lama untuk mencapai posisi stabil hingga *hover*. Sedangkan algoritma *fuzzy* yang menggunakan *rule base 3x3* waktu yang ditempuh cenderung lambat yaitu sebesar 17 detik dan ada *overshoot*. Peneliti lain juga telah mengembangkan algoritma *fuzzy* dengan mengubah *rule base* menjadi *7x7*, *quadcopter* mampu bergerak cepat untuk melakukan *take-off* dan *hover* namun masih terdapat *overshoot*. Penulis memilih algoritma *fuzzy Mamdani* dengan *rule base 7x7* di kadenakan dengan menggunakan *fuzzy Mamdani* dengan *rule base 7x7* diharapkan *quadcopter* dapat menempuh waktu yang cepat dalam mencapai posisi *hover* dan pengujian yang telah dilakukan oleh peneliti sebelumnya memberikan hasil bahwa tidak terjadi *overshoot* pada penerapan algoritma *fuzzy Mamdani*.

2.2 Dasar Teori

Pada dasar teori ini akan membahas tentang dasar-dasar penulis dalam membuat sistem. Adapun teori yang akan dijelaskan dalam bab ini antara lain *Quadcopter* dan Logika *fuzzy*.

2.2.1 Quadcopter

Quadcopter atau sering disebut '*quadrotor*' dapat dimodelkan sebagai sebuah konfigurasi empat buah motor pada sebuah kerangka kaku yang menyilang. Kerangka berbahan dasar fiber biasanya digunakan karena sifatnya yang ringan dan kuat. Pada masing masing motor terpasang baling-baling atau *propeller* untuk membuat aliran udara yang menghasilkan tekanan ke arah bawah sehingga timbul gaya angkat pada *quadcopter*. Gambar 2.1 Motor depan dan belakang memiliki arah putaran searah jarum jam. Sebaliknya motor kanan dan kiri berputar dengan arah berlawanan jarum jam (Kusuma, Effendi, Iskandar 2012).



Gambar 2.1 Model Quadcopter
Sumber : (Kusuma,et al., 2012)

Berikut merupakan gerakan yang dihasilkan oleh *quadcopter*

a. Gerakan *Throttle*

Gerakan ini dilakukan dengan menambah atau mengurangi kecepatan masing masing motor secara serempak.

b. Gerakan *Pitch*

Gerakan ini dihasilkan dari penambahan atau pengurangan kecepatan motor belakang sembari melakukan pengurangan atau penambahan kecepatan pada motor depan. Gerakan ini bertumpu pada sumbu X.

c. Gerakan *Roll*

Gerakan ini dilakukan dengan menambah atau mengurangi kecepatan motor kiri sembari mengurangi atau menambah kecepatan motor kanan. Gerakan ini bertumpu pada sumbu Y.

d. Gerakan *Yaw*

Gerakan ini dilakukan dengan menambah atau mengurangi kecepatan sepasang motor (kanan dan kiri) secara serempak sembari melakukan pengurangan atau penambahan kecepatan sepasang motor lainnya (depan dan belakang) secara serempak. Gerakan ini bertumpu pada sumbu Z.

2.2.2 Pemodelan *Quadcopter*

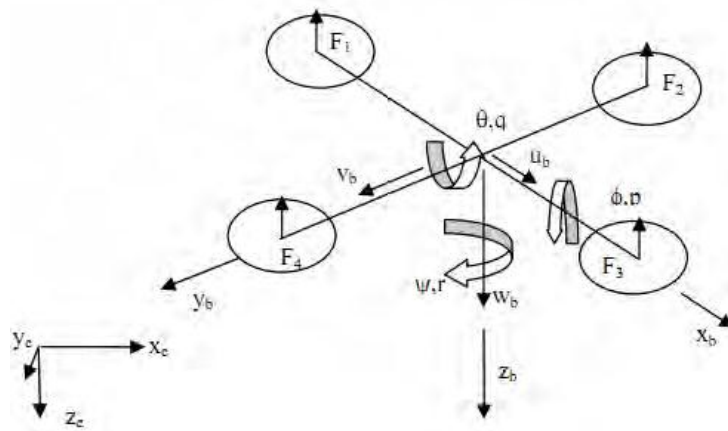
Perhitungan untuk mendapatkan pemodelan fisis yang tepat dapat dilakukan dengan memakai pendekatan asumsi mengenai kondisi fisik dari *quadcopter*. Hal ini bertujuan untuk menyederhanakan kompleksitas sistem yang dihasilkan dari dinamika *quadcopter*. Beberapa asumsi tersebut antara lain:

1. Struktur *body* dari *quadcopter* merupakan benda kaku
2. Struktur *body frame* dari *quadcopter* bersifat simetris sepanjang sumbu x dan y
3. Struktur dari *propeller* merupakan benda kaku(*rigid*)
4. Gaya *thrust* dan *drag* adalah proporsional dengan kuadrat dari kecepatan *propeller*

5. Keadaan model diasumsikan dalam keadaan *hovering*.
6. Aerodinamika seakan akan diabaikan
7. Nonlinieritas dari baterai diabaikan.
8. Tidak ada *slip* antara *propeller* dan motor.

Quadcopter memiliki 6 *degree of freedom* (DOF) yang menghasilkan keadaan (*state*) keluaran. Keadaan keluaran tersebut menghasilkan gerakan yang merepresentasikan *altitude* dari *quadcopter* yakni gerakan translasi (u, v, w) dan gerakan rotasi (ϕ, θ, ψ). Berikut variabel keluaran yang akan digunakan untuk menggambarkan posisi dan gerakan *quadcopter*.

- x = posisi *quadcopter* terhadap sumbu X_e
- y = posisi *quadcopter* terhadap sumbu Y_e
- z = posisi *quadcopter* terhadap sumbu Z_e
- u = kecepatan *quadcopter* yang diukur pada sumbu X_b
- v = kecepatan *quadcopter* yang diukur pada sumbu Y_b
- w = kecepatan *quadcopter* yang diukur pada sumbu Z_b
- ϕ = sudut *roll* terhadap sumbu X_e
- θ = sudut *pitch* terhadap sumbu Y_e
- ψ = sudut *yaw* terhadap sumbu Z_e
- p = kecepatan sudut *roll* yang diukur pada sumbu X_b
- q = kecepatan sudut *pitch* yang diukur pada sumbu Y_b
- r = kecepatan sudut *yaw* yang diukur pada sumbu Z_b



Gambar 2.2 Ilustrasi B-frame terhadap E-frame
Sumber : (Kusuma, et al., 2012)

Persamaan model dari *quadcopter* diperoleh melalui pemodelan fisik yang umum didapatkan dari beberapa penelitian. Persamaan yang digunakan adalah berdasarkan penelitian dari Bresciani(2015) telah dijelaskan sebelumnya bahwa dalam perancangan simulasi *quadcopter* digunakan 12 buah keluaran yang menentukan pola gerakan dari *quadcopter*. Persamaan model fisik yang merepresentasikan gerak translasi dan rotasi *quadcopter* tersebut dituliskan pada Persamaan 2.1 sampai dengan 2.6

$$\ddot{x} = (\sin\psi\sin\phi + \cos\psi\sin\vartheta\cos\phi) \frac{U1}{m} \quad (2.1)$$

$$\ddot{y} = (-\cos\psi\sin\phi + \sin\psi\sin\vartheta\cos\phi) \frac{U1}{m} \quad (2.2)$$

$$\ddot{z} = -g + (\cos\vartheta\cos\phi) \frac{U1}{m} \quad (2.3)$$

$$\dot{p} = \frac{Iyy-Izz}{Ixx} qr + \frac{jr}{Ixx} q\Omega + \frac{U2}{ixx} \quad (2.4)$$

$$\dot{q} = \frac{Izz-Ixx}{Iyy} pr + \frac{jr}{Iyy} q\Omega + \frac{U3}{iyy} \quad (2.5)$$

$$\dot{r} = \frac{Ixx-Iyy}{Izz} pq + \frac{U4}{izz} \quad (2.6)$$

2.2.3 Logika Fuzzy

Dalam bahasa Inggris, *fuzzy* mempunyai arti kabur atau tidak jelas. Jadi, logika *fuzzy* adalah logika yang kabur, atau mengandung unsur ketidakpastian. Pada logika biasa, yaitu logika tegas, kita hanya mengenal dua nilai, salah atau benar, 0 atau 1. Sedangkan logika *fuzzy* mengenal nilai antara benar dan salah. Kebenaran dalam logika *fuzzy* dapat dinyatakan dalam derajat kebenaran yang nilainya antara 0 sampai 1. Misalnya dalam kehidupan sehari-hari, dewasa didefinisikan dengan berusia 17 tahun ke atas. Jika menggunakan logika tegas, seseorang yang berusia 17 tahun kurang 1 hari akan didefinisikan sebagai tidak dewasa. Namun dalam logika *fuzzy*, orang tersebut dapat dinyatakan dengan hampir dewasa (Saelan 2009).

2.2.4 Himpunan Fuzzy

Himpunan *fuzzy* adalah pengelompokan sesuatu berdasarkan variabel bahasa (*linguistic variable*), yang dinyatakan dengan fungsi keanggotaan, dalam semesta U. Keanggotaan suatu nilai pada himpunan dinyatakan dengan derajat keanggotaan yang nilainya antara 0.0 sampai 1.0.

Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan *real* pada interval [0,1]. Nilai keanggotaannya menunjukkan bahwa suatu item tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah.

2.2.5 Fungsi Keanggotaan

Dalam logika tegas, fungsi keanggotaan menyatakan keanggotaan pada suatu himpunan. Fungsi keanggotaan $\chi_A(x)$ bernilai 1 jika x anggota himpunan A, dan bernilai 0 jika x bukan anggota himpunan A. Jadi, fungsi keanggotaan ini hanya bisa bernilai 0 atau 1 seperti pada Persamaan 2.7.

$$\chi_A : X \rightarrow \{0,1\} \quad (2.7)$$

Sedangkan dalam logika *fuzzy*, fungsi keanggotaan menyatakan derajat keanggotaan pada suatu himpunan. Nilai dari fungsi keanggotaan ini berada dalam selang [0,1], dan dinyatakan dengan μ_A seperti pada Persamaan 2.8.

$$\mu_A : \mathcal{X} \rightarrow [0,1] \quad (2.8)$$

Fungsi keanggotaan $\mu_A(x)$ bernilai 1 jika x anggota penuh himpunan A , dan bernilai 0 jika x bukan anggota himpunan A . Sedangkan jika derajat keanggotaan berada dalam selang $(0,1)$, misalnya $\mu_A(x) = \mu$, menyatakan x sebagian anggota himpunan A dengan derajat keanggotaan sebesar μ .

Ada 3 cara mendefinisikan himpunan *fuzzy*:

1. Sebagai himpunan pasangan berurutan

Misalkan himpunan *fuzzy* A didefinisikan dalam semesta $\mathcal{X} = \{\chi_1, \chi_2, \dots, \chi_n\}$, maka himpunan pasangan berurutan yang menyatakan himpunan *fuzzy*-nya adalah ditunjukkan pada Persamaan 2.9.

$$A = \{ (\chi_1, \mu_A(\chi_1)), (\chi_2, \mu_A(\chi_2)), \dots, (\chi_n, \mu_A(\chi_n)) \} \quad (2.9)$$

Cara ini hanya dapat digunakan pada himpunan *fuzzy* yang anggotanya bernilai diskrit.

2. Dengan menyebut fungsi keanggotaan

Misalkan himpunan *fuzzy* A didefinisikan dalam semesta X yang anggotanya bernilai kontinu, maka himpunan pasangan berurutan yang menyatakan himpunan *fuzzy*-nya adalah ditunjukkan pada Persamaan 2.10.

$$A = \{ (\chi, \mu_A(\chi)) \mid \mu_A(\chi) = \dots, \chi \in X \} \quad (2.10)$$

3. Menuliskan Persamaan 2.11 dan 2.12 sebagai

$$A = \{ \mu_A(\chi_1)/\chi_1 + \mu_A(\chi_2)/\chi_2 + \dots + \mu_A(\chi_n)/\chi_n = \{ \sum_{i=1}^n \mu_A(\chi_i)/\chi_i \} \quad (2.11)$$

untuk X diskrit, atau

$$A = \{ \int x \mu_A(\chi) / \chi \} \quad (2.12)$$

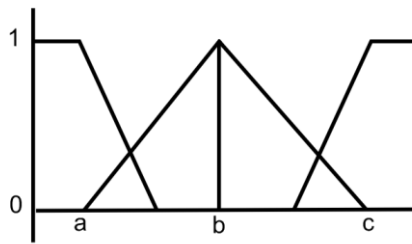
untuk X kontinu. Lambang \int bukan berarti integral.

Fungsi keanggotaan suatu himpunan *fuzzy* dapat ditentukan dengan fungsi segitiga (*triangel*), trapesium (*trapezoidal*), atau fungsi gauss (*Gaussian*).

Persamaan 2.13 merupakan persamaan dari fungsi keanggotaan segitiga:

$$\mu(x, a, b, c) = \left\{ \begin{array}{ll} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 1 & x > c \end{array} \right\} \quad (2.13)$$

Persamaan tersebut direpresentasikan dalam bentuk grafik sebagai berikut.



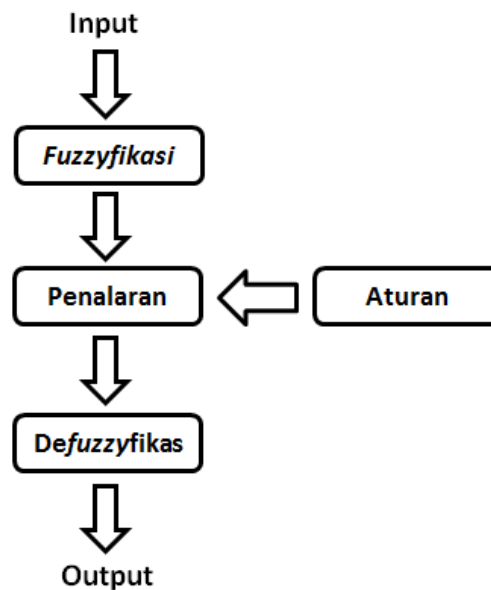
Gambar 2.3 Grafik fungsi keanggotaan segitiga
Sumber : (Saelan 2012)

2.2.6 Kendali Logika Fuzzy

Sistem kendali logika *fuzzy* disebut juga sistem Inferensi *Fuzzy* (*Fuzzy Inference System/FIS*) atau *fuzzy inference engine* adalah sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya.

Terdapat beberapa jenis FIS yang dikenal yaitu Mamdani, Sugeno dan Tsukamoto. FIS yang paling mudah dimengerti, karena paling sesuai dengan naluri manusia adalah FIS Mamdani. FIS tersebut bekerja berdasarkan kaidah-kaidah linguistik dan memiliki algoritma *fuzzy* yang menyediakan sebuah aproksimasi untuk dimasuki analisa matematik. Sistem kendali logika *fuzzy* terdiri dari beberapa tahapan yang dapat dilihat pada Gambar 2.6.

Input yang diberikan adalah berupa bilangan tertentu dan *output* yang dihasilkan juga harus berupa bilangan tertentu. Aturan-aturan dalam bahasa linguistik dapat digunakan sebagai *input* yang bersifat teliti harus dikonversikan terlebih dahulu, lalu melakukan penalaran berdasarkan aturan-aturan dan mengonversi hasil penalaran tersebut menjadi *output* yang bersifat teliti.



Gambar 2.4 Proses Kendali Logika Fuzzy
Sumber : (Saelan.A, 2012)

2.2.7 Fuzzyfikasi

Fuzzyfikasi adalah pemetaan nilai *input* yang merupakan nilai tegas ke dalam fungsi keanggotaan himpunan *fuzzy*, untuk kemudian diolah di dalam mesin penalaran seperti ditunjukkan pada Persamaan 2.16.

$$\text{fuzzyfikasi} : \chi \rightarrow \mu(\chi) \quad (2.16)$$

2.2.8 Aturan Dasar

Aturan dasar dalam kendali logika *fuzzy* adalah aturan implikasi dalam bentuk “jika ... maka ...”. Aturan dasar tersebut ditentukan dengan bantuan seorang pakar yang mengetahui karakteristik objek yang akan dikendalikan. Contoh bentuk implikasi yang digunakan adalah sebagai berikut.

Jika $X = A$ dan $Y = B$ maka $Z = C$.

2.2.9 Penalaran

Pada tahapan ini sistem menalar nilai masukan untuk menentukan nilai keluaran sebagai bentuk pengambil keputusan. Sistem terdiri dari beberapa aturan, maka kesimpulan diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu *max*, *additive* dan probabilistik OR.

Pada metode *max*, 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*). Secara umum dapat ditulis seperti Persamaan 2.17.

$$\mu_{df}(\chi_i) \leftarrow \max(\mu_{df}(\chi_i) \mu_{kf}(\chi_i)) \quad (2.17)$$

Selain itu, salah satu model penalaran yang banyak digunakan adalah max-min. Dalam penalaran ini, pertama-tama dilakukan proses operasi min sinyal keluaran lapisan *fuzzyfikasi*, kemudian diteruskan dengan operasi max untuk mencari nilai keluaran yang selanjutnya akan di *defuzzyfikasikan* sebagai bentuk keluaran pengendali. Operasi max-min tersebut dapat dinyatakan sebagai Persamaan 2.18 dan 2.19.

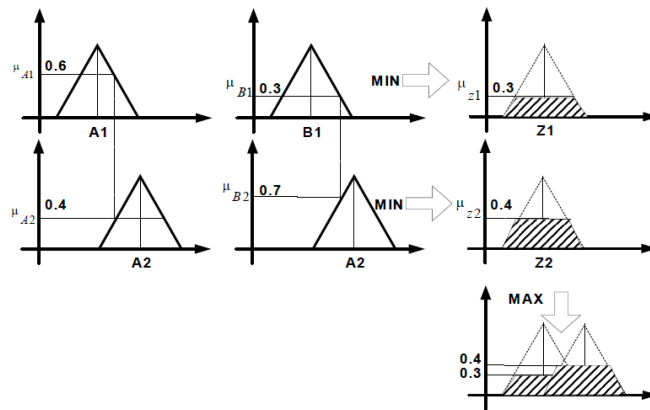
Operasi min atau irisan

$$\begin{aligned} a \wedge b &= \min(a, b) = a \text{ if } a \leq b \\ &= b \text{ if } a > b \end{aligned} \quad (2.18)$$

Operasi max atau gabungan

$$\begin{aligned} a \vee b &= \max(a, b) = a \text{ if } a \geq b \\ &= b \text{ if } a < b \end{aligned} \quad (2.19)$$

Proses penalaran max-min dijelaskan dalam grafik berikut.



Gambar 2.5 Proses Penalaran max-min
Sumber : (Saelan,A, 2012)

2.2.10 Defuzzyfikasi

Defuzzyfikasi merupakan kebalikan dari *fuzzyfikasi*, yaitu pemetaan dari himpunan *fuzzy* ke himpunan tegas. *Input* dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*. Hasil dari *defuzzyfikasi* ini merupakan *output* dari sistem kendali logika *fuzzy* seperti pada Persamaan 2.20.

Defuzzyfikasi dideskripsikan sebagai

$$Z^* = \text{defuzzyfier}(Z) \quad (2.20)$$

dengan

Z = hasil penalaran *fuzzy*

Z^* = keluaran kendali logika *fuzzy*

defuzzyfier = fungsi *defuzzyfikasi*

Metode *defuzzyfikasi* antara lain:

- Metode Maximum
- Metode ini juga dikenal dengan metode puncak, yang nilai keluarannya dibatasi oleh fungsi $\mu_C(z^*) > \mu_1(z)$.
- Metode titik tengah
- Metode titik tengah juga disebut metode pusat area. Metode ini lazim dipakai dalam proses *defuzzyfikasi*. Keluaran dari metode ini adalah titik tengah dari hasil proses penalaran.
- Metode rata-rata
- Metode ini digunakan untuk fungsi keanggotaan keluaran yang simetris. Keluaran dari metode ini adalah nilai rata-rata dari hasil proses penalaran.
- Metode penjumlahan titik tengah
- Keluarannya dari metode ini adalah penjumlahan titik tengah dari hasil proses penalaran.

- i. Metode titik tengah area terbesar
- j. Dalam metode ini, keluarannya adalah titik pusat dari area terbesar yang ada.