

BAB 2

LANDASAN TEORI

2.1 Definisi Robot

Istilah “robot” berasal dari kata “robota” (bahasa Czech) yang berarti “kerja”. Kamus besar Webster memberikan definisi mengenai robot, yaitu “sebuah peralatan otomatis yang melakukan pekerjaan seperti yang dilakukan oleh manusia”.

Robot Institute of America memberikan definisi robot adalah ”Manipulator dengan fungsi ganda dan dapat diprogram kembali, didesain untuk dapat memindahkan komponen, peralatan-peralatan khusus melalui pergerakan yang diprogram agar dapat melakukan berbagai kegiatan (Schllusel, 1985).”

Robot adalah sebuah alat yang dapat diprogram kembali, dibuat untuk memanipulasi dan memindahkan komponen, peralatan dengan cara pergerakan yang diprogram untuk tugas-tugas tertentu (Koren,1985).

Service Robot (Robot Layanan) merupakan suatu robot yang digunakan untuk membantu atau melayani manusia dalam kehidupan sehari-hari.

Singkatannya Robot adalah manipulator serba guna yang dapat diprogram kembali, mempunyai eksternal sensor dan dapat melakukan berbagai tugas untuk membantu tugas manusia.

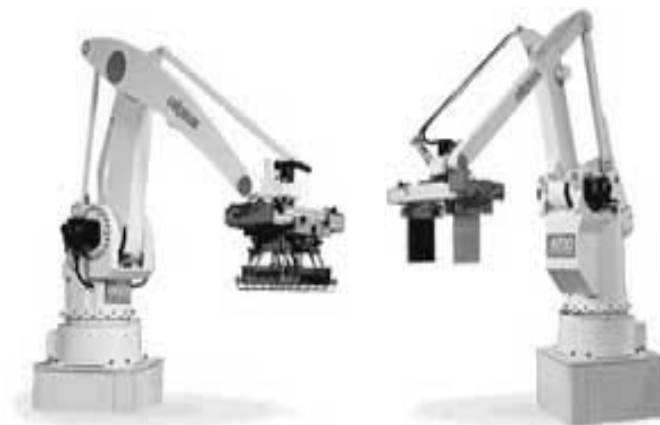
2.1.1 Klasifikasi Umum dari Robot

Berdasarkan sifat mobilitasnya, robot dapat diklasifikasikan kedalam 2 bagian besar, yaitu:

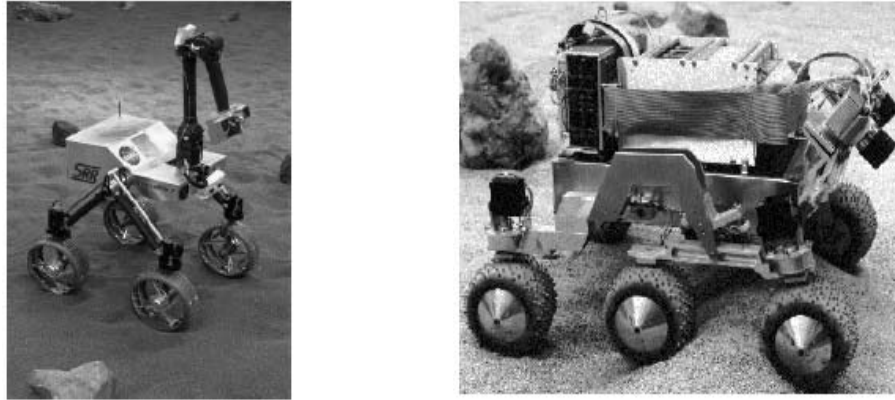
- a. *Fixed robot*, robot yang memiliki ruang kerja (*spatial space*) yang terbatas, dimana bagian dasarnya (*base*) dilekatkan pada sebuah benda tetap seperti panel atau meja.
- b. *Mobile robot*, robot yang memiliki ruang kerja (*spatial space*) yang cukup luas, dimana bagian dasarnya pada sebuah alat gerak seperti roda/ban atau kaki. Beberapa macam *mobile robot* antara lain :
 - *Wheeled robot* (robot beroda), robot yang bergerak dengan menggunakan perputaran roda/ban.
 - *Walking/legged robot* (robot berkaki), robot yang bergerak dengan menggunakan kaki dalam perpindahannya.

Pembahasan lebih lanjut akan berfokus pada jenis robot mobil dengan alat gerak roda (*wheeled robot*), khususnya tipe *differential drive*. Untuk tipe *fixed robot* dan *legged robot* tidak akan dibahas lebih lanjut.

Di bawah ini adalah beberapa contoh gambar robot :



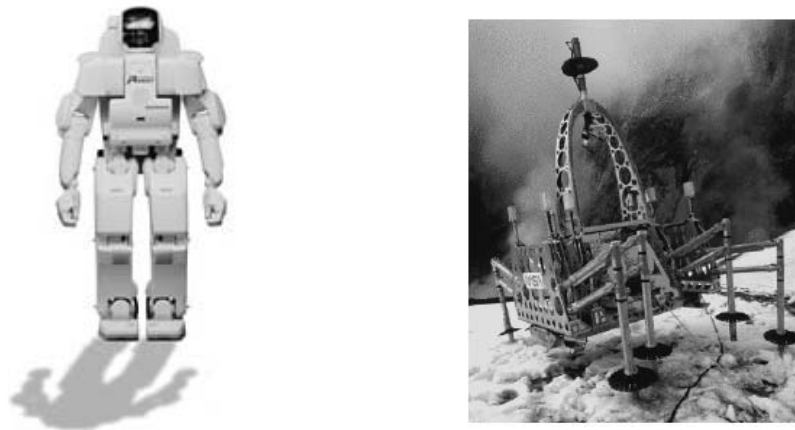
Gambar 2.1 Fixed Robot



Gambar 2.2 Robot Beroda

(http://robots.mit.edu/people/Karl/SPIE_00.pdf)

(<http://www.frc.ri.cmu.edu/~alonzo/course/course.html>)



Gambar 2.3 Robot Berkaki

(http://www.honda-robots.com/index_ns.html)

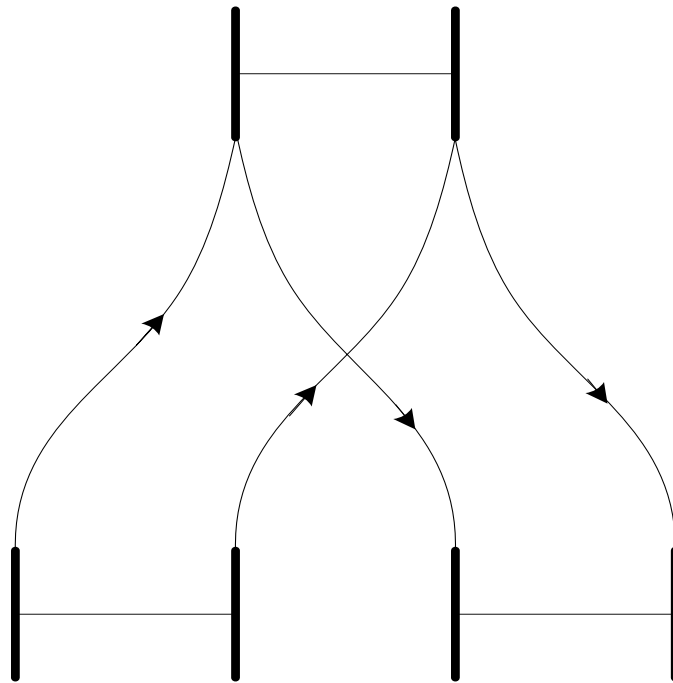
(<http://www.frc.ri.cmu.edu/~alonzo/course/course.html>)

2.1.2 Robot Mobil Beroda

Beberapa jenis robot mobil beroda adalah *differential drive*, *omni directional*, *syncho drive*. Jenis *differential drive* akan digunakan untuk

simulasi kinematika robot mobil karena jenis ini lebih mudah digunakan dan telah banyak diimplementasikan robot mobil.

Robot mobil yang menggunakan roda ada yang bersifat *holonomic* dan *non-holonomic*, robot mobil yang akan disimulasikan menggunakan roda yang bersifat *non-holonomic*, yaitu masing-masing roda kiri dan roda kanan dapat bergerak bebas. Contoh aplikasi yang menggunakan model ini adalah kursi roda. Perbedaan antara *holonomic* dengan *non-holonomic* adalah pada sistem *holonomic* kembali ke posisi awal berarti kembali ke posisi sistem semula, sedangkan pada sistem *non-holonomic* kembali ke posisi awal roda tidak berarti kembali ke posisi sistem semula. Seperti terlihat pada gambar 2.4.



Gambar 2.4 Pergerakan Roda pada Kinematika Non-holonomic

2.1.3 Differential Drive

Differential Drive adalah suatu sistem penggerak beroda dua dengan aktuator yang independen untuk masing-masing roda. Roda penggerak biasanya terletak pada masing-masing sisi samping robot. Robot yang menggunakan dua roda membutuhkan suatu alat tambahan sehingga robot dapat menyeimbangkan diri, biasanya menggunakan roda pendukung yang diletakkan di depan atau di belakang robot, seperti terlihat pada gambar 2.5 robot yang menggunakan tipe *differential drive* dengan menggunakan roda pendukung di belakang. Penambahan roda ini diharapkan tidak mengganggu pergerakan robot mobil sehingga roda pendukung ini ukurannya harus lebih kecil dari roda penggerak dan bergerak bebas pada axis vertikal.

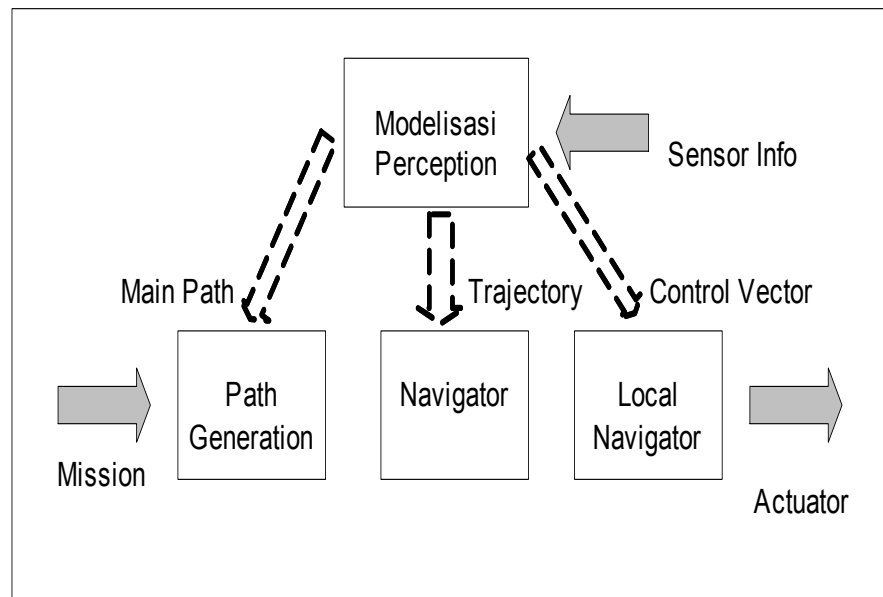


Gambar 2.5 Robot Differential Drive

<http://www.isr.ist.utl.pt/~mir/cadeiras/robmove1/Kinematics.pdf>

2.2 Navigasi Robot Mobil

Proses yang digunakan oleh robot mobil untuk bergerak dari posisi awal ke posisi tujuan dengan memperhatikan *initial frame*.



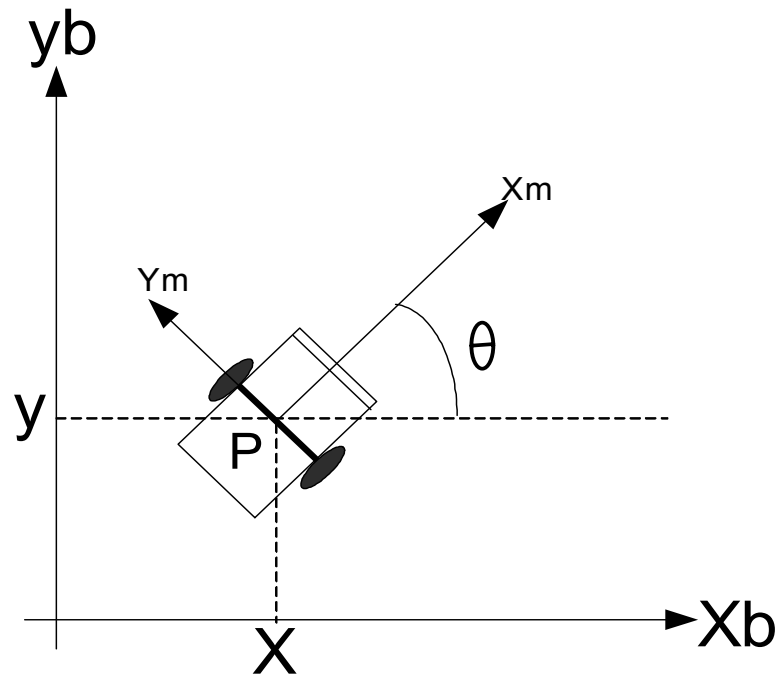
Gambar 2.6 Diagram Blok Navigasi Robot Mobil

(Gernot Kroneif and Probst, 1997)

Gambar 2.6 di atas merupakan blok diagram navigasi robot mobil, robot menggunakan sensor sebagai input bagi robot untuk mengenali lingkungan sekitarnya melalui *perception*. Pada *perception* terdapat sensor interpretasi dan sensor integrasi yang digunakan oleh robot mobil untuk melihat keadaan sekitar selama perjalanan dari posisi awal ke posisi tujuan sehingga robot mobil dapat mengetahui bila ada rintangan atau halangan di depannya. Untuk dapat mencapai posisi tujuan yang diinginkan harus dibuat *path planning* (denah) dan misi apa yang harus dilakukan oleh

2.3 Kinematika Robot Mobil

Kinematika adalah ilmu yang mempelajari pergerakan tanpa memperhitungkan gaya yang menyebabkan pergerakan tersebut. Kinematika robot mobil adalah ilmu yang mempelajari pergerakan robot mobil tanpa memperhitungkan gaya yang menyebabkan pergerakan robot mobil.

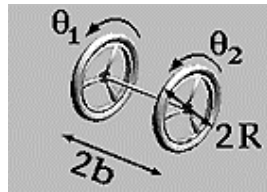


Gambar 2.8 Kinematika Robot Mobil

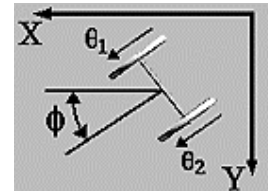
- $\{Y_b, X_b\}$ adalah *base frame*, merupakan sumbu kartesian yang mempresentasikan ruang dalam robot.
- $[Y_m, X_m]$ adalah *moving frame*, merupakan sumbu kartesian yang mempresentasikan robot terhadap ruang.
- θ adalah sudut *moving* robot mobil terhadap ruang.
- P adalah titik pusat robot mobil

2.3.1 Persamaan Kinematika Differential Drive

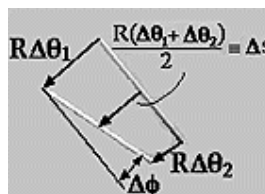
Persamaan 2.1, 2.2, 2.3 dan 2.4 adalah persamaan kinematika *non-holonomic* dimana hubungan antara perputaran roda dan posisi/orientasi pada base (<http://www.nd.edu/NDInfo/Research/sskaar/KinematicEquations.html>).



Gambar 2.9a Jarak Antara 2 Roda



Gambar 2.9b Sudut Roda



Gambar 2.10 Jarak dan Sudut Antara 2 Roda dengan Kecepatan Berbeda

$$\Delta s = \frac{R(\Delta\theta_1 + \Delta\theta_2)}{2} \dots\dots\dots(2.1)$$

Jika $\Delta\Phi$ mendekati 0 maka $\text{tg } \Delta\Phi = \Delta\Phi$,

$$\Delta\phi \approx \frac{R(\Delta\theta_1 - \Delta\theta_2)}{2b} \dots\dots\dots(2.2)$$

$$\Delta X \approx \Delta s \cos \phi \dots\dots\dots(2.3)$$

$$\Delta Y \approx \Delta s \sin \phi \dots\dots\dots(2.4)$$

dimana :

θ_1 = lintasan angular roda 1

θ_2 = lintasan angular roda 2

$2R$ = diameter roda

$2b$ = jarak antara roda 1 dan roda 2

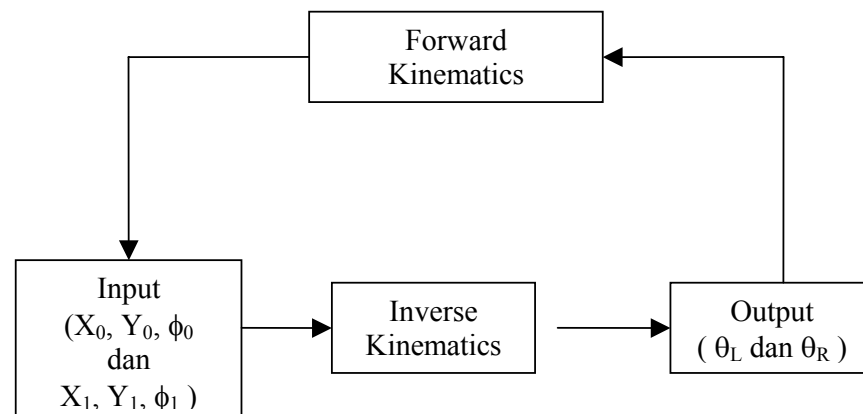
X = sumbu x

Y = sumbu y

Terdapat dua permasalahan dalam kinematika yaitu *Forward kinematics* dan *Inverse kinematics* [Lee, Fu, Gonzalez, 1987].

2.3.2 Forward Kinematics

Forward kinematics berhubungan dengan mencari solusi untuk mendapatkan posisi dan orientasi dari robot dimana sudah diketahui titik awal posisi robot dan nilai-nilai θ_L dan θ_R yang akan dijalankan sehingga dapat dicari titik akhir posisi robot akan berhenti.



Gambar 2.11 Diagram Blok Hubungan Inverse Kinematics dan Forward Kinematics

Dari Gambar 2.11 dapat dilihat hubungan antara *inverse kinematics* dan *forward kinematics*, output dari *inverse kinematics* dapat digunakan

untuk membuktikan kebenaran solusinya yaitu dengan memberikan output *inverse kinematics* menjadi masukan untuk *forward kinematics*.

2.3.3 Inverse Kinematics

Proses *Inverse kinematics* digunakan untuk mencari nilai perputaran roda kiri dan kanan (θ_L dan θ_R) dari robot mobil, untuk mencapai posisi dan orientasi yang diinginkan. Nilai perputaran roda kiri dan kanan (θ_L dan θ_R) didapat dengan menggunakan metode fuzzy, artinya θ_L dan θ_R merupakan output dari fuzzy ($\dot{\theta}_L$ dan $\dot{\theta}_R$) yang dikalikan dengan waktu sampling. Metode fuzzy ini akan dijelaskan pada subbab 2.4.

2.4 Fuzzy Logic

Logika fuzzy pertama kali diperkenalkan oleh Profesor Lotfi Zadeh dari Universitas California di Berkeley pada tahun 1962. Logika fuzzy merupakan suatu teori himpunan logika yang dikembangkan untuk mengatasi konsep nilai yang terdapat diantara kebenaran (*truth*) dan kesalahan (*false*). Dengan menggunakan logika fuzzy nilai yang dihasilkan bukan hanya ya (1) atau tidak (0) tetapi seluruh kemungkinan diantara 0 dan 1.

2.4.1 Konsep Fuzzy Logic

2.4.1.1 Perbedaan Antara Himpunan Crips dan Himpunan Fuzzy

Untuk membedakan antara himpunan klasik (himpunan crips) dan himpunan fuzzy diberikan beberapa contoh dibawah ini :

1. Himpunan Mahasiswi Cantik di Universitas Bina Nusantara Jurusan Sistem Komputer

a. Himpunan Crips

$$A = \{ \text{Dewi, Florentina, Tina, Luvita} \}$$

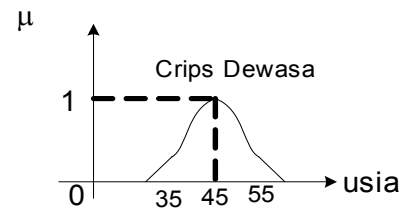
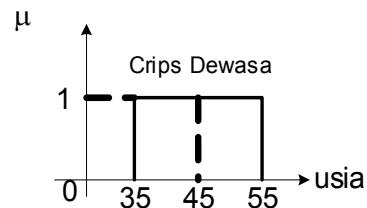
Keanggotaan Dewi, Florentina, Tina dan Luvita adalah 1, sedangkan keanggotaan mahasiswa yang lain adalah 0.

b. Himpunan Fuzzy

$$B = \{ (\text{Dewi}, 0.6), (\text{Florentina}, 0.7), (\text{Tina}, 0.8), (\text{Luvita}, 0.9) \}$$

Angka 0.6 – 0.9 menunjukkan tingkat keanggotaan (degree of membership).

2. Ekspresi Usia Dewasa



Gambar 2.12a Crips Dewasa Non Fuzzy Gambar 2.12b Crips Dewasa Fuzzy

Pada Himpunan Crips gambar 2.12a di atas dapat dilihat bahwa orang yang usianya 34,5 tahun tidak termasuk dalam anggota himpunan usia dewasa tetapi masuk dalam himpunan anggota usia muda, hal ini terasa tidak natural karena perbedaannya kecil sekali dengan orang yang usianya 35 tahun. Dengan menggunakan konsep fuzzy usia ini dihubungkan dengan tingkat keanggotaan sehingga lebih natural dan sesuai dengan cara berpikir manusia (gambar 2.12b).

2.4.1.2 *Membership Function* (Fungsi Keanggotaan)

Fungsi Keanggotaan menghubungkan suatu elemen x pada suatu himpunan dengan tingkat keanggotaannya.

Gambar 2.13 dibawah adalah model fungsi keanggotaan (sumber : <http://faculty.petra.ac.id/resmana/private/fuzzy>), beberapa istilah dari gambar tersebut adalah :

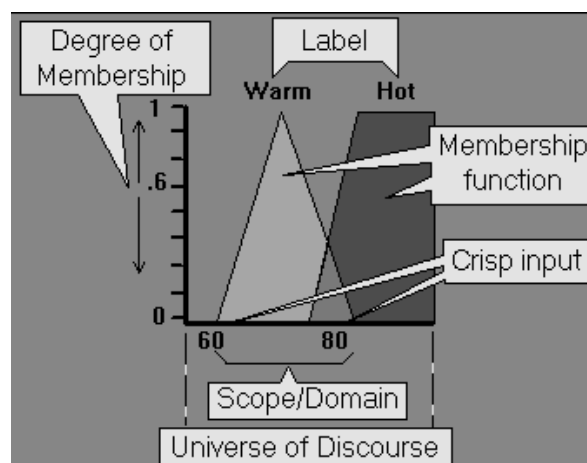
LABEL : Menentukan nama deskripsi untuk mengidentifikasi himpunan fuzzy

UNIVERSE Of DISCOURSE : Merupakan jarak semua nilai kemungkinan yang akan digunakan dalam variabel sistem.

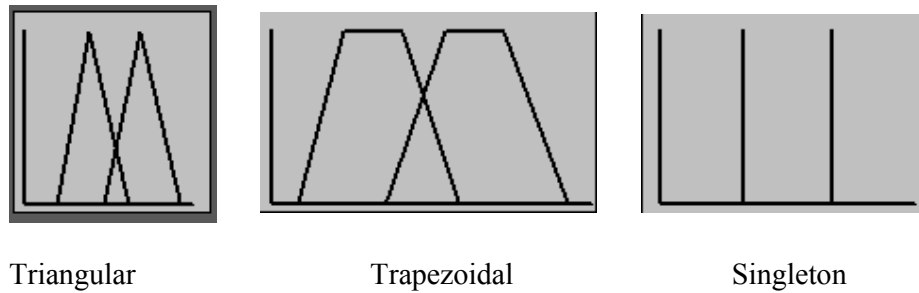
SCOPE/DOMAIN : merupakan lebar dari fungsi keanggotaan.

CRIPS INPUT : digunakan untuk menghasilkan fuzzy input, contoh: suhu, kecepatan.

DEGREE Of MEMBERSHIP (derajat keanggotaan) : merupakan nilai yang berkisar antara 0 sampai 1.



Gambar 2.13 Fungsi Keanggotaan



Gambar 2.14 Bentuk-bentuk Fungsi Keanggotaan

Gambar 2.14 di atas adalah beberapa bentuk dari fungsi keanggotaan antara lain *Triangular*, *Trapezoidal*, *Singleton*, selain itu juga ada bentuk *bell*, *haversine*, *exponential*. Bentuk *Triangular* dan *Trapezoidal* biasanya yang paling banyak digunakan sebagai fungsi keanggotaan untuk fuzzy input sedangkan untuk fuzzy output biasanya digunakan bentuk *Singleton* karena bentuk ini mudah untuk direpresentasikan di komputer dan algoritma defuzzifikasinya lebih mudah.

2.4.1.3 Operasi Himpunan Fuzzy

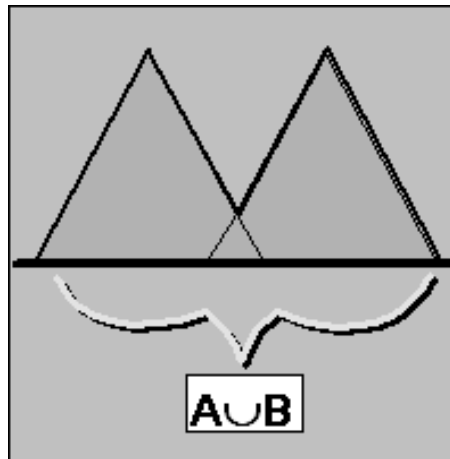
Beberapa operasi himpunan fuzzy yang digunakan untuk menentukan tingkat derajat keanggotaan, antara lain :

a. Operasi Union

Union dari himpunan fuzzy A dan himpunan fuzzy B adalah himpunan fuzzy $A \cup B$, dengan fungsi keanggotaan :

$$\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)] \text{ untuk setiap } x \in X$$

Derajat keanggotaan setiap himpunan fuzzy $A \cup B$ adalah derajat keanggotaannya pada himpunan fuzzy A atau B yang memiliki nilai lebih besar.



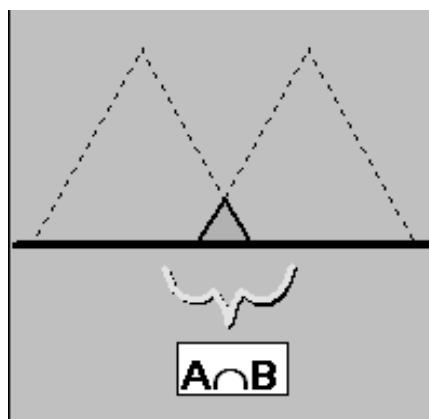
Gambar 2.15 Operasi Union

c. Operasi Intersection

Intersection dari himpunan fuzzy A dan himpunan fuzzy B adalah himpunan fuzzy $A \cap B$, dengan fungsi keanggotaan :

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)] \text{ untuk setiap } x \in X$$

Derajat keanggotaan setiap himpunan fuzzy $A \cap B$ adalah derajat keanggotaannya pada himpunan fuzzy A atau B yang memiliki nilai lebih kecil.

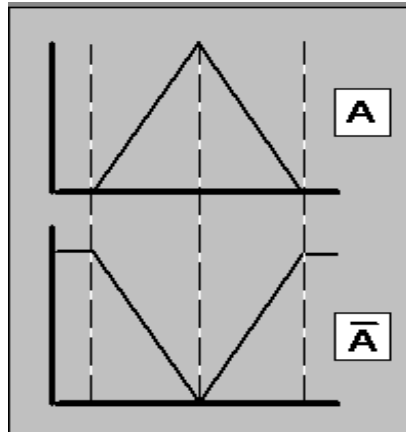


Gambar 2.16 Operasi Intersection

d. Operasi Complement

Complement dari fuzzy set A adalah A^c dengan fungsi keanggotaan :

$$\mu_{A^c}(x) = 1 - \mu_A(x) \text{ untuk setiap } x \in X$$



Gambar 2.17 Operasi Complement

e. Set Inclusion

Himpunan fuzzy A dan B dikatakan sama (equal) ($A=B$) jika dan hanya jika

$$\mu_A(x) = \mu_B(x) \text{ untuk setiap } x \in X$$

Himpunan Fuzzy A merupakan propersubset dari fuzzy set B ($A \subset B$) jika dan hanya jika :

$$\mu_A(x) \leq \mu_B(x) \text{ untuk setiap } x \in X$$

2.4.2 Fuzzy Logic Controller (FLC)

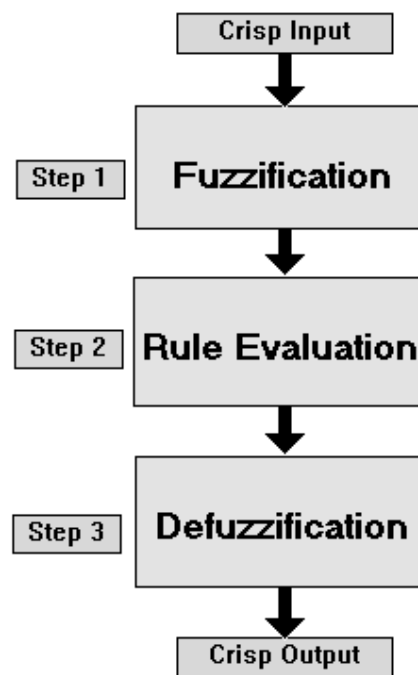
Fuzzy Logic Controller digunakan untuk melakukan pengontrolan terhadap suatu alat atau sistem. Cara kerja Fuzzy Logic Controller adalah seperti pola pikir manusia, FLC akan membuat keputusan berdasarkan *fuzzy rules* (aturan fuzzy) yang dibuat dengan menggunakan syntax :

If antecedent 1 AND antecedent 2 ... THEN

Consequent 1 AND consequent 2...

AND merupakan salah satu operasi logika yang dipakai dalam operasi logika fuzzy, *Antecedent* merupakan bentuk dari variabel input = label, contoh : temperature = hot. *Consequent* merupakan bentuk dari variabel output = label, contoh : watering duration = long.

Gambar 2.18 merupakan blok diagram yang digunakan untuk merancang suatu *Fuzzy Logic Controller*.

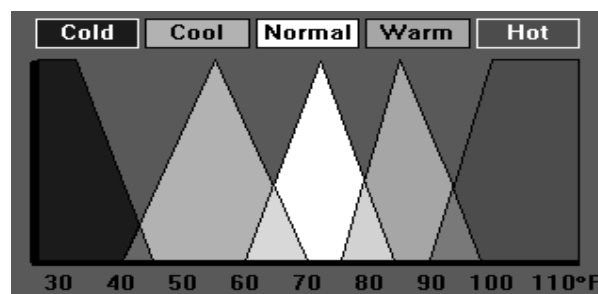


Gambar 2.18 Fuzzy Logic Controller

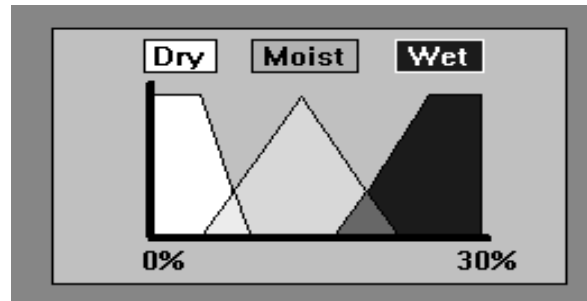
Dari gambar 2.18 di atas terdapat tiga step untuk membuat *fuzzy logic controller* antara lain :

- Step 1 *Fuzzification* (Fuzzifikasi) : Meliputi pengambilan nilai crisp input yang kemudian dengan fungsi keanggotaan menghasilkan fuzzy input.
- Step 2 *Rule evaluation* (Penggunaan aturan logika) : Pada tahap ini fuzzy input dibandingkan dengan aturan-aturan yang telah dibuat dan akan menghasilkan fuzzy output.
- Step 3 *Defuzzification* (Defuzzifikasi) : Tahap terakhir ini akan melakukan perubahan hasil fuzzy output menjadi crisp output.

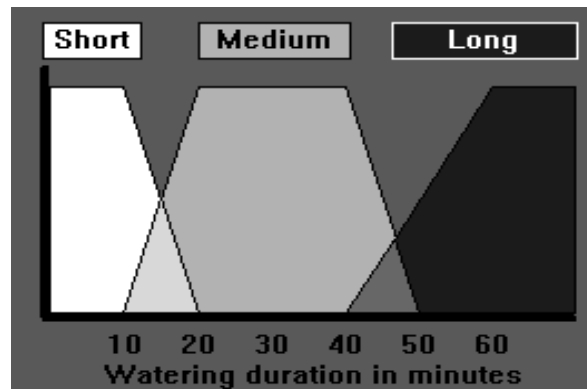
Untuk dapat lebih mengerti cara kerja setiap step di atas, berikut adalah contoh sederhana bagaimana menggunakan *Fuzzy Logic Controller*. Sistem penyiram rumput, langkah pertama adalah menentukan fungsi keanggotaan dengan menggunakan 2 crisp input yaitu *out door temperature* dan *soil moisture*. Sedangkan untuk crisp outputnya adalah *watering duration*.



Gambar 2.19 Fungsi Keanggotaan dari Input Out Door Temperature



Gambar 2.20 Fungsi Keanggotaan dari Input Soil Moisture



Gambar 2.21 Fungsi Keanggotaan untuk Output Watering Duration

Tabel 2.1 Label Dan Domain

LABEL	DOMAIN
Cold	30° F – 47° F
Cool	40° F – 70° F
Normal	60° F – 84° F
Warm	75° F – 98° F
Hot	90° F – 110° F

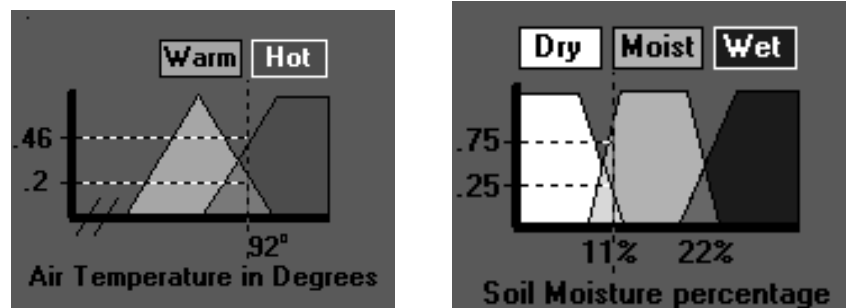
Tabel 2.1 merupakan nama label dan domain untuk Fungsi Keanggotaan dari input out door temperature, sehingga dapat dilihat pada gambar 2.19 fungsi keanggotaannya.

Langkah kedua adalah menentukan aturan fuzzy, aturan fuzzy menggunakan logika **IF-THEN**. Beberapa aturan yang dibuat berdasarkan gambar 2.22 dibawah adalah *IF temperature is hot AND soil is dry THEN watering duration is long*, *IF temperature is cold AND soil is wet THEN watering duration is short*. Rule yang dibuat dapat direpresentasikan dalam bentuk tabel, sehingga mudah untuk membacanya.

Untuk lebih lengkapnya aturan yang dapat dibuat dapat dilihat pada gambar 2.22.

		Temperature				
		Cold	Cool	Normal	Warm	Hot
Moisture	Wet	short	short	short	short	short
	Moist	short	med.	med.	med.	med.
	Dry	long	long	long	long	long

Gambar 2.22 Rule (Aturan) Sistem Penyiram Rumput



Gambar 2.23 Fungsi Keanggotaan pada Air Temperature dan Soil Moisture

Untuk input *air temperature* pada 92° , pada gambar 2.23 dapat dilihat nilai keanggotaan untuk *warm* 0.2 dan *hot* 0.46. Sedangkan untuk input *soil moisture* pada 11%, nilai keanggotaan untuk *dry* 0.25 dan *Moist* 0.75. Dengan

menggunakan operasi AND (intersection), dapat ditentukan *rule strength* (kekuatan aturan) untuk setiap aturan :

IF temperature is hot (0.46) AND soil is dry (0.25) THEN watering duration is long (rule strength 0.25)

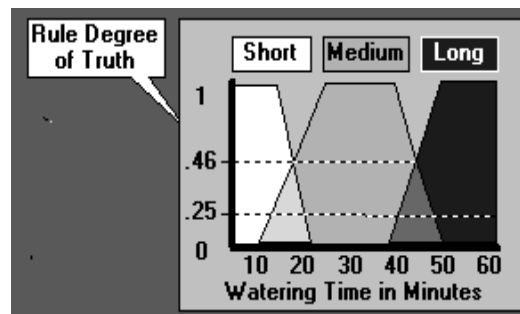
IF temperature is warm (0.2) AND soil is moist (0.75) THEN watering duration is medium (rule strength 0.2)

IF temperature is warm (0.2) AND soil is dry (0.25) THEN watering duration is long (rule strength 0.2)

IF temperature is hot (0.46) AND soil is moist (0.75) THEN watering duration is medium (rule strength 0.46)

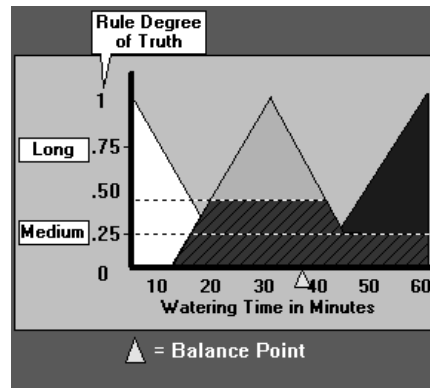
Langkah ketiga adalah defuzzifikasi, untuk mendapatkan crisp output biasanya menggunakan teknik *Center of Gravity (COG)*.

Untuk contoh sistem penyiram rumput dengan *temperature* 92° F dan *Soil moisture* 11% didapat *rule strength* yang terbesar adalah 0.25 dan 0.46, sehingga dapat digambarkan fungsi keanggotaan outputnya seperti pada gambar 2.24.



Gambar 2.24 Fungsi Keanggotaan Output

Dengan menggunakan teknik *Center of Gravity (COG)*, sehingga fungsi keanggotannya seperti gambar 2.25, dengan mencari '*balancing point*' dari *Center of Gravity* didapat craps output untuk watering duration adalah 38 menit.

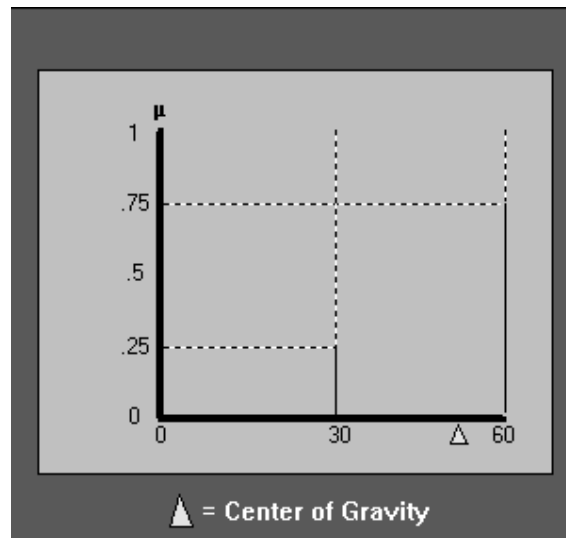


Gambar 2.25 Craps Output untuk Watering Duration

Rumus mencari craps output menggunakan *Center of Gravity*, yaitu :

$$\boxed{\text{COG}} = \frac{\int_a^b \mu(x) \cdot x \, dx}{\int_a^b \mu(x) \, dx}$$

Center of Gravity dapat juga digunakan untuk fungsi keanggotaan dengan output tipe singleton seperti gambar 2.26 dibawah :



Gambar 2.26 Crips Output Tipe Singleton

Untuk menghitung besarnya *Center of Gravity* dengan menggunakan rumus :

$$\text{Crisp Output (Y)} = \frac{\sum_i (\text{fuzzy output}_i) \times (\text{Singleton position on x axis}_i)}{\sum_i (\text{fuzzy output}_i)}$$