

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Berdasarkan penelitian sebelumnya banyak peneliti yang sudah membahas tentang sistem kendali motor BLDC dengan menggunakan pengendali diantaranya, meneliti tentang Pengendali Kecepatan Motor *Brushless* Dc (BLDC) Menggunakan Metode *Fuzzy*[1]. pada penelitian ini menggunakan kendali fuzzy untuk mengendalikan kecepatan motor *brushless* dc. Dari hasil penelitian yang dilakukan, dapat disimpulkan bahwa metode defuzzifikasi yang dilakukan mampu mengikuti setting kecepatan yang diberikan adalah metode COA. Pengujian perubahan kecepatan dari 1000 rpm sampai 2000 rpm dihasilkan karakteristik tanggapan sistem kendali PID konvensional dengan rata rata waktu kenaikan (t_r) 0.29 *second*, rata rata waktu tunak (t_s) 0.9 *second*, rata rata overshoot sebesar 8.63%, dan persentase ISE sebesar 98,19%, sedangkan pada *fuzzy* dihasilkan rata rata waktu kenaikan (t_r) 0.25 *second*, waktu tunak (t_s) 0.27 *second*, rata rata overshoot 0.15% dan persentase ISE 99.36%. sistem kendali *fuzzy* yang di implementasikan untuk mengatur motor BLDC dapat memperbaiki kinerja dari PID konvensional.

Penelitian selanjutnya meneliti tentang Sistem Kendali Kecepatan Motor BLDC Menggunakan Metode Algoritma *Hybrid* PID *Fuzzy*[7]. Pada penelitian ini dapat disimpulkan proses kendali dengan sistem penalaan parameter kendali PID dengan logika *fuzzy* yang di aplikasikan untuk mengatur motor BLDC dapat memperbaiki kinerja kendali PID konvensional. Pengujian perubahan set poin dan perubahan beban, dihasilkan karakteristik tanggapan sistem kendali PID konvensional dengan nilai rata rata yaitu waktu naik (t_s) 0.025 detik, waktu penetapan (t_s) 0.1625 detik, *overshoot* 15.98%, sedangkan kendali *hybrid* PID *fuzzy* dihasilkan rata rata waktu naik (t_r) 0.0025 detik, waktu penetapan (t_s) 0.057 detik, *overshoot* sebesar 5.42%. Dapat disimpulkan bahwa kendali hybrid PID fuzzy mampu meningkatkan kinerja dari kendali PID konvensional.

Penelitian selanjutnya meneliti tentang desain kendali *fuzzy* PID *gain scheduling* untuk pengaturan kecepatan motor DC tanpa sikat[8]. Pada penelitian ini penulis merancang sebuah sistem yang memberikan pembebanan pada motor BLDC menggunakan rem magnetik agar sistem menjadi tidak linear. Pengendalian yang digunakan adalah PID dan *fuzzy* untuk mengatur K_p , K_i , dan K_d . Setelah diimplementasikan pada kendali di dapat

bahwa nilai *rise time* rata rata respon motor sebesar 2.6 detik dan nilai setting time rata rata sebesar 3.6 detik, selain itu didapat juga *overshoot* yang relatif kecil sekitar 1% .

Penelitian selanjutnya meneliti tentang Metode *Six Step Comutation* pada Perancangan Rangkaian Kendali *Sensored Motor Brushless Direct Current*[9]. Pada penelitian ini dilakukan perancangan dan pengujian inverter tiga fase dengan metode *sensored* menggunakan sensor *hall effect* sebagai penentu posisi rotor untuk mengendalikan motor *brushless direct current*. Topologi inverter *full – bridge* diaplikasikan dengan MOSFET kanal N IRFZ44. Rangkaian pengendali motor *brushless direct current* di uji dan dianalisis unjuk kerjanya. Hasil pengujian menunjukkan pengaruh kenaikan *duty – cycle* terhadap kecepatan motor, makin besar nilai *duty – cycle* yang di atur maka makin besar pula kecepatan motor.

Penelitian selanjutnya dilakukan meneliti tentang Perancangan dan Pengujian Awal Kendali Motor DC *Brushless Untuk Independent 4-Wheel Drive Platform Robot Rev-11*[10]. Pada penelitian ini sistem yang dibuat terdiri dari 2 bagian yaitu modul kendali motor BLDC dan kendali *supervisor* yang berfungsi untuk mengkoordinasi perintah ke modul modul kendali motor, kendali *supervisory* mengirim data referensi berupa kecepatan berupa kecepatan dan arah pada modul kendali motor sebagai referensi untuk mengendalikan kecepatan dan arah dari masing masing aktuator pada *platform REV 11*. Dari hasil pengujian disimpulkan bahwa sistem kendali yang didesain sudah mampu berfungsi dengan baik untuk mengkoordinasi dan mengendalikan kecepatan dan arah gerak motor aktuator *platform REV 11*.

Penelitian selanjutnya dilakukan meneliti tentang Aplikasi Kendali *Fuzzy Logic* Untuk Pengaturan Kecepatan Motor Universal[11]. Pada penelitian ini pengaturan kecepatan motor dilakukan dengan mengatur tegangan motor dan menggunakan metode *Pulse Wudth Modulation* (PWM). Metode Defuzzinifikasi yang digunakan adalah metode *mean of maximum* dan *center of area*. Jumlah rule bervariasi tergantung dari jumlah label yang digunakan, respon sistem ditampilkan dalam bentuk grafik kecepatan motor terhadap waktu. Hasil pengujian menunjukkan membership function dengan bentuk segitiga atau *trapezoid* tidak memberikan pengaruh yang cukup besar terhadap respon sistem. Metode defuzzinifikasi *center of area* lebih baik dibandingkan dengan metode *mean of maxima*. Respon sistem akan lebih baik bila menggunakan membership function dengan jumlah label yang lebih banyak.

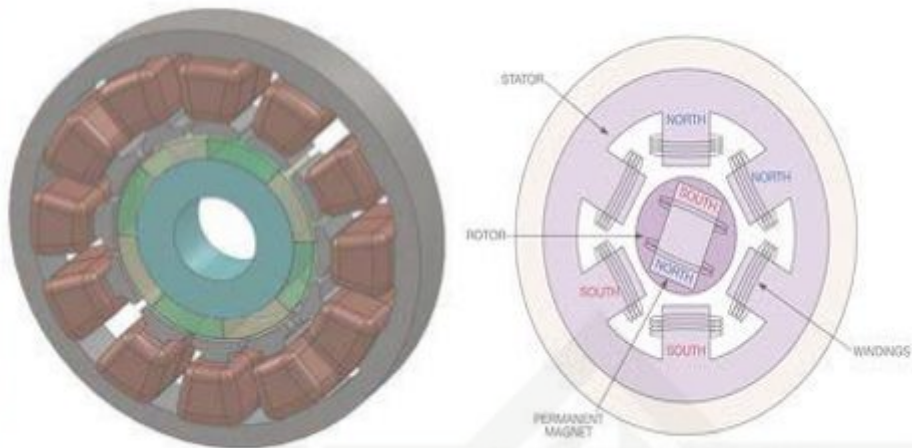
Berdasarkan referensi yang diuraikan di atas maka penulis menyimpulkan bahwa kendali *hybrid* *pid – fuzzy* adalah suatu kendali yang tidak memiliki pemodelan matematis kompleks, dengan mendefinisikan variabel masukan dan keluaran serta menetapkan setiap himpunan *fuzzy* pada variabel, dan memiliki kendali yang stabil dan memiliki risetime yang cepat, walaupun terdapat *overshoot* sedikit.

2.2 Motor BLDC

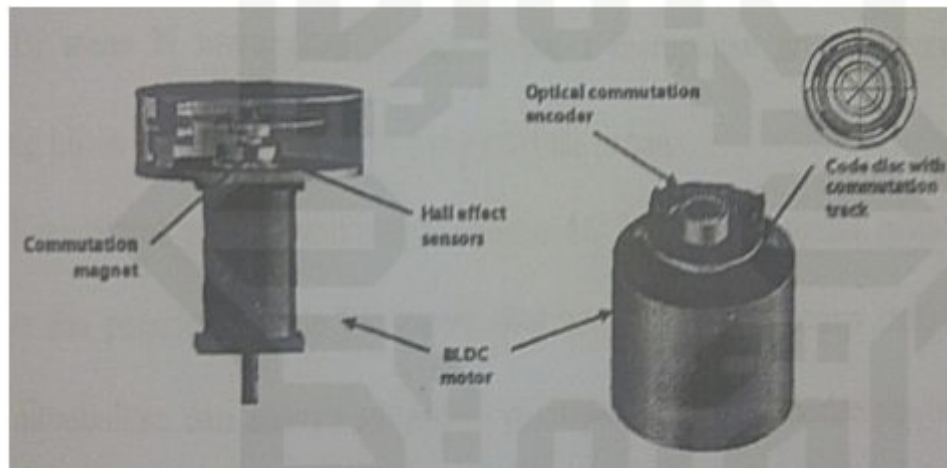
BLDC motor atau dapat disebut juga dengan BLAC motor merupakan motor listrik *synchronous* AC 3 fasa. Perbedaan pemberian nama ini terjadi karena BLDC memiliki BEMF berbentuk *trapezoid*, sedangkan BLAC memiliki BEMF berbentuk sinusoidal. Walaupun demikian keduanya memiliki struktur yang sama. Dibandingkan dengan motor DC, BLDC memiliki biaya perawatan yang lebih rendah dan kecepatan yang lebih tinggi akibat tidak digunakannya *brush*. Dibandingkan dengan motor induksi, BLDC memiliki efisiensi yang lebih tinggi karena rotor dan torsi awal yang lebih tinggi karena rotor terbuat dari magnet permanen. Walaupun memiliki kelebihan dibandingkan dengan motor DC dan motor induksi, pengendalian BLDC jauh lebih rumit untuk kecepatan dan torsi yang konstan karena tidak adanya *brush* yang menunjang proses komutasi dan harga BLDC jauh lebih mahal.[12]

Secara umum BLDC terdiri dari dua bagian, yakni rotor, bagian yang bergerak, yang terbuat dari permanen magnet dan stator, bagian yang tidak bergerak, yang terbuat dari kumparan 3 fasa. Walaupun merupakan motor listrik *synchronous* AC 3 fasa, motor ini tetap disebut dengan BLDC karena pada implementasinya BLDC menggunakan sumber DC sebagai sumber energi utama yang kemudian diubah menjadi tegangan AC dengan menggunakan inverter 3 fasa. Tujuan dari pemberian tegangan AC 3 fasa pada stator BLDC adalah menciptakan medan magnet putar stator untuk menarik magnet rotor.[12]

Oleh karena tidak adanya *brush* pada motor BLDC, untuk menentukan timing komutasi yang tepat pada motor ini sehingga didapatkan torsi dan kecepatan yang konstan, diperlukan 3 buah sensor *Hall* atau *encoder*. Pada sensor *Hall*, *timing* komutasi ditentukan dengan cara mendeteksi medan magnet rotor dengan menggunakan 3 buah sensor *Hall* untuk mendapatkan 6 kombinasi *timing* yang berbeda, sedangkan pada *encoder*, *timing* komutasi ditentukan dengan cara menghitung jumlah pola yang ada pada *encoder*. [12]



Gambar 2.1 Penampang motor DC[12]

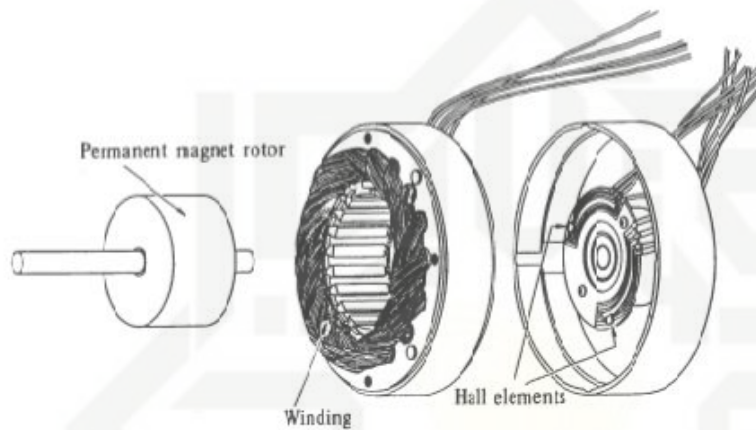


Gambar 2.2 Sensor *hall* dan *encoder* pada motor BLDC[12]

Pada umumnya encoder lebih banyak digunakan pada motor BLDC komersial karena *encoder* cenderung mampu menentukan timing komutasi lebih presisi dibandingkan dengan menggunakan sensor *hall*. Hal ini terjadi karena pada *encoder*, kode komutasi telah ditetapkan secara *fixed* berdasarkan banyak *pole* dari motor dan kode inilah yang digunakan untuk menentukan *timing* komutasi. Namun karena kode komutasi encoder untuk suatu motor tidak dapat digunakan untuk motor dengan jumlah pole yang berbeda. Hal ini berbeda dengan sensor Hall. Apabila terjadi perubahan pole rotor pada motor, posisi sensor *hall* dapat diubah dengan mudah. Hanya saja kelemahan dari sensor *hall* adalah apabila posisi sensor *hall* tidak tepat akan terjadi kesalahan dalam penentuan *timing* komutasi atau bahkan tidak didapatkan 6 kombinasi *timing* komutasi yang berbeda.

2.3 Kontruksi motor BLDC

Pembangunan motor BLDC ini sangat mirip dengan motor AC, gambar di bawah menggambarkan struktur dari motor tiga fase BLDC. Pada gambar dibawah menggambarkan struktur dari tiga fase motor BLDC, gulungan statornya sama dengan yang terdapat pada motor polyphase AC, dan rotornya terdiri dari satu atau lebih magnet permanen.[13]



Gambar 2.3 Motor BLDC sederhana. [13]

2.4 Prinsip Kerja Motor BLDC

Motor BLDC ini dapat bekerja ketika stator yang terbuat dari kumparan diberikan arus 3 phasa. Akibat arus yang melewati kumparan pada stator timbul medan magnet (B):

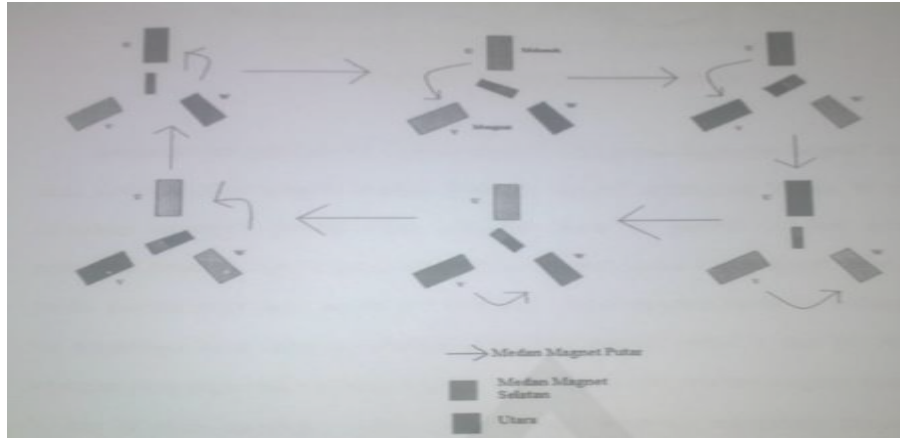
$$B = \frac{\mu Ni}{2l} \tag{2.1}$$

Di mana N merupakan jumlah lilitan, i merupakan arus, l merupakan panjang lilitan dan μ merupakan permeabilitas bahan.

Karena arus yang diberikan berupa arus AC 3 phasa sinusoidal, nilai medan magnet dan polarisasi setiap kumparan akan berubah-ubah setiap saat. Akibat yang ditimbulkan dari adanya perubahan polarisasi dan besar medan magnet tiap kumparan adalah terciptanya medan putar magnet dengan kecepatan.[12]

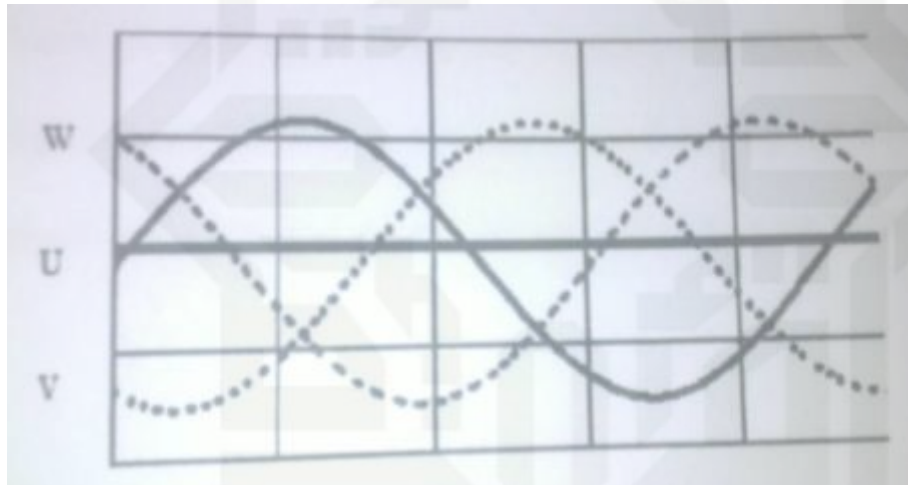
$$\eta_s = \frac{120f}{p} \tag{2.2}$$

Di mana f merupakan frekuensi arus input dan p merupakan jumlah *pole* rotor.



Gambar 2.4 Magnet putar stator dan perputaran rotor[12]

Berdasarkan gambar 2.4, medan putar magnet stator timbul akibat adanya perubahan polaritas pada stator U, V, dan W. Perubahan polaritas ini terjadi akibat adanya arus yang mengalir pada stator berupa arus AC yang memiliki polaritas yang berubah-ubah.



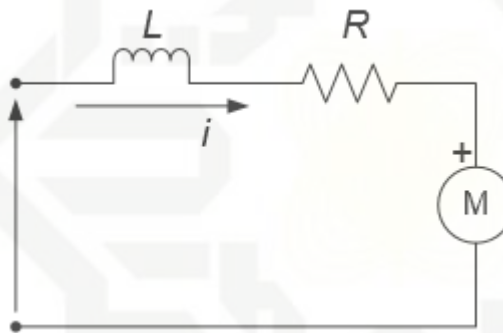
Gambar 2.5 Tegangan stator BLDC[12]

Berdasarkan gambar 2.5, ketika stator U diberikan tegangan negative maka akan timbul medan magnet dengan polaritas negative sedangkan V dan W yang diberikan tegangan positif akan memiliki polaritas positif. Akibat adanya perbedaan polaritas antara medan magnet kumparan stator dan magnet rotor, sisi positif magnet rotor akan berputar mendekati medan magnet stator U, sedangkan sisi negatifnya akan berputar mengikuti medan magnet stator V dan W. Akibat tegangan yang digunakan berupa tegangan AC sinusoidal, medan magnet stator U, V, dan W akan berubah-ubah polaritas dan besarnya mengikuti perubahan tegangan sinusoidal AC. Ketika U dan V memiliki medan magnet negatif akibat mendapatkan tegangan negatif dan W memiliki medan magnet positif akibat tegangan positif, magnet permanen rotor akan berputar menuju ke polaritas yang bersesuaian yakni bagian negatif akan berputar menuju medan magnet stator W dan sebaliknya

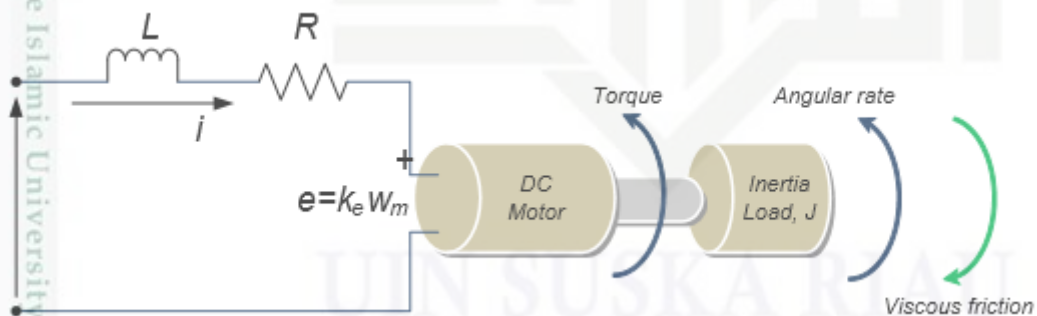
bagian postif akan berputar menuju medan magnet stator U dan V. Selanjutnya ketika V memiliki medan magnet negatif dan U serta W memiliki medan magnet postif, bagian postif bagian postif magnet permanen akan berputar menuju V dan bagian negatif akan menuju U dari kumparan W. Karena tegangan AC sinusoidal yang digunakan berlangsung secara *kontinu*, proses perubahan polaritas tegangan pada stator ini akan terjadi secara terus menerus sehingga menciptakan medan putar magnet stator dan magnet permanen rotor akan berputar mengikuti medan putar magnet stator ini. Hal inilah yang menyebabkan rotor pada BLDC dapat berputar.[12]

2.5 Model Matematika Motor BLDC

Pemodelan matematika dari motor BLDC tidak benar-benar berbeda dari motor DC konvensional. Untuk pemodelan motor DC dapat dilihat dari gambar 2.5



Gambar 2.6 rangkaian motor listrik sederhana[14]



Gambar 2.7 Diagram simetris motor DC[14]

Dari gambar diatas dapat dibuat persamaan menggunakan hukum kirchhoff,persamaan sebagai berikut:

$$V_s = R_{\phi}i + L \frac{di}{dt} + e$$

$$V_s = R_{\phi}i + e \tag{2.3}$$

Untuk mencari emf dari motor DC maka dari persamaan 2.3 dapat ditulis dengan persamaan berikut

$$e = -Ri - L \frac{di}{dt} + V_s \quad (2.4)$$

Dimana,

V_s = Tegangan motor DC

i = Angker saat ini

Mengingat sifat mekanik motor DC, tentang hukum kedua dari newton tentang gerak, sifat mekanik relatif terhadap torsi pengaturan sistem pada gambar 2.3 dan 2.4 akan menjadi hasil dari beban inersia, J adalah tingkat kecepatan sudut, ω_m sama dengan jumlah semua torsi, dapat dilihat dari persamaan 2.3 dan 2.4

$$J \frac{d\omega_m}{dt} = \sum T_i \quad (2.5)$$

$$T_e = k_f \omega_m + J \frac{d\omega_m}{dt} + T_L \quad (2.6)$$

Dimana,

T_e = Torsi listrik

k_f = Gesekan konstan

J = Rotor inersia

ω_m = Kecepatan sudut

T_L = Beban mekanik

Dimana torsi listrik dan emf dapat di tulis dengan

$$e = k_e \omega_m \text{ dan } T_e = k_t \omega_m \quad (2.7)$$

Dimana,

k_e = emf konstan

k_t = torsi konstan

Maka persamaan 2.2 dan 2.3 dapat di tulis sebagai berikut;

$$\frac{di}{dt} = -i \frac{R_\phi}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad (2.8)$$

$$\frac{d\omega_m}{dt} = i \frac{k_t}{J} - \frac{k_f}{J} \omega_m + \frac{1}{J} T_L \quad (2.9)$$

Menggunakan persamaan laplace maka persamaan 2.6 dan 2.7 dapat ditulis sebagai berikut;

Untuk Persamaan 2.8,

$$L \left\{ \frac{di}{dt} = -i \frac{R_\phi}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \right\} \quad (2.10)$$

$$s i = -i \frac{R_\phi}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad (2.11)$$

Untuk Persamaan 2.9,

$$L \left\{ \frac{d\omega_m}{dt} = -i \frac{k_t}{J} - \frac{k_e}{J} \omega_m + \frac{1}{J} T_L \right\} \quad (2.12)$$

Menjadi,

$$s \omega_m = i \frac{k_t}{J} - \frac{k_f}{J} \omega_m + \frac{1}{J} T_L \quad (2.13)$$

Tanpa beban untuk $T_L = 0$ persamaan dari 2.13 menjadi

$$s \omega_m = i \frac{k_t}{J} - \frac{k_f}{J} \omega_m \quad (2.14)$$

Dari persamaan 2.12, i pada persamaan 2.9 dapat diganti dengan menggunakan persamaan berikut;

$$i = \frac{s \omega_m + \frac{k_f}{J} \omega_m}{\frac{k_t}{J}} \quad (2.15)$$

$$\left(\frac{s \omega_m + \frac{k_f}{J} \omega_m}{\frac{k_t}{J}} \right) \left(s + \frac{R_\phi}{L} \right) = -\frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad (2.16)$$

Persamaan 2.16 menjadi;

$$\left\{ \left(\frac{s^2 J}{k_t} + \frac{s k_f}{k_t} + \frac{s R_\phi J}{k_t L} + \frac{k_f R_\phi}{k_t L} \right) + \frac{k_e}{L} \right\} \omega_m = \frac{1}{L} V_s \quad (2.17)$$

Dan persamaan 2.17 di sederhanakan pada persamaan 2.18

$$V_s = \left\{ \frac{s^2 J L + s k_f L + s R_\phi J + k_f R_\phi + k_e k_t}{k_t} \right\} \omega_m \quad (2.18)$$

Maka, transfer function dapat di cari dengan menggunakan rasio dan kecepatan sudut, ω_m kesumber tegangan V_s adalah;

$$G(s) = \frac{\omega_m}{V_s} = \frac{k_t}{s^2 J L + s k_f L + s R_\phi J + k_f R_\phi + k_e k_t} \quad (2.19)$$

Fungsi alih tersebut diturunkan lagi menjadi;

$$G(s) = \frac{\omega_m}{V_s} = \frac{k_t}{s^2 J L + (R_\phi J + k_e L) + k_f R_\phi + k_e k_t} \quad (2.20)$$

Mengingat asumsi sebagai berikut

1. Gesekan konstan kecil, yaitu k_f cenderung 0, ini berarti;
2. $R_\phi J \gg k_f L$ dan $k_e k_t \gg R k_f$

Setelah mengabaikan nilai – nilai tersebut, fungsi alih dapat di tulis sebagai berikut;

$$G(s) = \frac{\omega_m}{V_s} = \frac{k_t}{s^2 J L + R_\phi J s + k_e k_t} \quad (2.21)$$

Jadi, dengan kembali pengaturan manipulasi matematika pada JL dengan mengalikan atas dan bawah di persamaan 2.21

$$\frac{R_\phi}{k_e k_t} \times \frac{1}{R_\phi}$$

Maka di peroleh persamaan setelah manipulasi sebagai berikut;

$$G(s) = \frac{\frac{1}{k_e}}{\frac{R_\phi J}{k_e k_t} \cdot \frac{L}{R_\phi} \cdot s^2 + \frac{R_\phi J}{k_e k_t} \cdot s + 1} \quad (2.22)$$

Dari persamaan 2.15 konstanta berikut didapat

Ketetapan waktu mekanik;

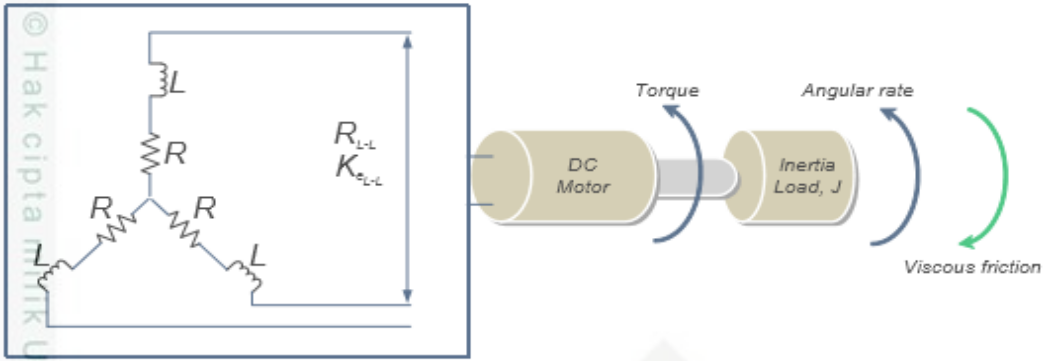
$$\tau_m = \frac{R_\phi J}{k_e k_t} \quad (2.23)$$

Ketetapan waktu konstan;

$$\tau_e = \frac{L}{R_\phi} \quad (2.24)$$

Pada pemodelan motor BLDC tidak jauh berbeda dengan motor DC hal yang membedakan pemodelannya terletak pada fase yang mempengaruhi hasil keseluruhan model BLDC. Fase khusus yang mempengaruhi resitif dan induktif dari susunan BLDC. Misalnya pengaturan sederhana dengan simetris 3-fase dan ini koneksi internal yang bisa memberikan gambaran singkat tentang konsep fase keseluruhan.[14]

Jadi disini ditampilkan perbedaan persamaan dari motor DC konvensional. Perbedaan ini akan mempengaruhi konstanta mekanik dan listrik karena mereka adalah bagian yang sangat penting dari parameter pemodelan.



Gambar 2.8 Diagram simetris BLDC[14]

Untuk ketetapan waktu mekanik menjadi:

$$\tau_m = \sum \frac{R_\phi J}{K_e K_t} = \frac{F \sum R_\phi}{K_e K_t} \quad (2.25)$$

$$\tau_m = \frac{J \cdot 3}{K_e K_t} \quad (2.26)$$

Untuk mengikat efek fase,

$$\tau_m = \frac{3 \cdot R_\phi \cdot J}{K_e K_t} \quad (2.27)$$

Ketetapan waktu konstan,

$$\tau_e = \sum \frac{L}{R_\phi} = \frac{L}{\sum R_\phi} \quad (2.28)$$

$$\tau_e = \frac{L}{3 \cdot R_\phi} \quad (2.29)$$

Dimana K_e dan K_t adalah nilai fase EMF konstan:

$$K_e = \frac{3 \cdot R_\phi \cdot J}{\tau_m K_t} \quad (2.30)$$

Dimana,

$$K_e = \left[\frac{V \cdot \text{secs}}{\text{rad}} \right] : \text{nilai fase listrik}$$

$$K_t = \left[\frac{N \cdot m}{A} \right] : \text{putaran konstan}$$

Oleh karena itu persamaan untuk BLDC sekarang dapat di peroleh sebagai berikut dengan mempertimbangkan efek dari konstanta dan fase yang sesuai :

$$G(s) = \frac{\frac{1}{K_e}}{\tau_m \cdot \tau_e \cdot S^2 + \tau_m \cdot S + 1} \quad (2.31)$$

Dimana:

- K_e = Nilai fase EMF
- τ_m = Ketetapan waktu mekanik
- τ_e = Ketetapan waktu konstan
- L = Terminal induksi fase ke fase
- R_ϕ = terminal resistansi fase ke fase
- J = Rotor inersia

Pada rancangan ini motor BLDC yang digunakan adalah type maxson EC flat ϕ 45mm, dimana spesifikasi motor yang di gunakan adalah sebagai berikut:

Tabel 2.1 Parameter Motor BLDC[15]

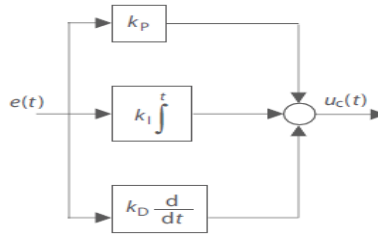
	Characteristics	Nilai	Satuan
1	Terminal resistansi fase ke fase (R)	1.1	Ω
2	Terminal induksi fase ke fase (L)	0.50	mH
3	Putaran konstan (K_t)	24.5	mNm/A
4	Kecepatan konstan (K_s)	35.4	rpm/V
5	Gradasi kecepatan / putaran (K_θ)	17.6	rpm/mNm
6	Ketetapan waktu mekanik (τ_m)	16.1	Ms
7	Rotor inersia (j)	82.5	gcm^2
8	Nomor fase	3	

2.6 PID

PID adalah salah satu pengendali otomatis yang terdiri dari tiga kendali, yaitu kendali Proporsional (P), Integral (I), Derivatif (D), dengan masing – masing memiliki kelebihan dan kekurangan. Tujuan dari penggabungan ketiga jenis kendali tersebut adalah untuk menutupi kekurangan dan menonjolkan kelebihan dari masing – masing jenis kendali[4].

Persamaan kendali PID dapat dituliskan sebagai berikut:

$$u(t) = K_p e(t) + K_i \int e dt + K_d \frac{de}{dt} \quad (2.32)$$



Gambar 2.9 Blok diagram kendali PID[4]

2.6.1 Kendali Proporsional

Kendali *Proporsional* memiliki berbagai keterbatasan karena sifat kendali yang tidak dinamik. Walaupun begitu kendali *Proporsional* cukup mampu untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Ciri – ciri dari kendali *Proporsional* adalah saat nilai K_p kecil, maka pengendali *Proporsional* hanya mampu menghilangkan *error* yang kecil, sehingga menghasilkan respon sistem yang lambat (menambah *rise time*). Namun nilai K_p dapat diatur sedemikian rupa sehingga mengurangi *steady state error*, tetapi tidak menghilangkannya.

2.6.2 Kendali Integratif

Pengendali integratif berfungsi menghasilkan respon sistem yang memiliki *error steady state* =0. Jika sebuah pengendali tidak memiliki unsur integrator, pengendali proporsional tidak akan mampu menjamin keluaran sistem dengan *error steady state* =0.

Keluaran pengendali ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadi perubahan masukan. Sinyal keluaran pengendali integral merupakan luas bidang yang dibentuk oleh kurva *error*. Ciri – ciri pengendali integral adalah *output* dari pengendali integral memerlukan selang waktu tertentu, sehingga pengendali *integral* cenderung memperlambat respon. Konstanta integral K_i yang berharga besar akan mempercepat hilangnya *offset*. Tetapi semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengendali.

2.6.3 Kendali Derivatif

Keluaran kendali diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengendali akan mengakibatkan

perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan, keluaran pengendali juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menarik, keluaran menghasilkan sinyal berbentuk impuls. Jika sinyal masukan berubah naik secara perlahan, keluaran justru merupakan fungsi *step* yang besar *magnitudenya* sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan faktor konstanta K_d . Dengan meningkatkan nilai K_d , dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*.

Dari karakteristik pengendali *derivatif*, pengendali ini umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. kerja pengendali diferensial hanya efektif pada lingkungan yang sempit, yaitu periode peralihan. Oleh karena itu pengendali diferensial tidak pernah digunakan tanpa ada kendali lain.

Transfer fungsi dari pengendali PID adalah :

$$U_c(s) = \left[k_p + \frac{k}{s} + k_D s \right] E(s) \quad (2.33)$$

Respon sistem kendali PID menurut parameter dapat dilihat pada tabel 2.1.

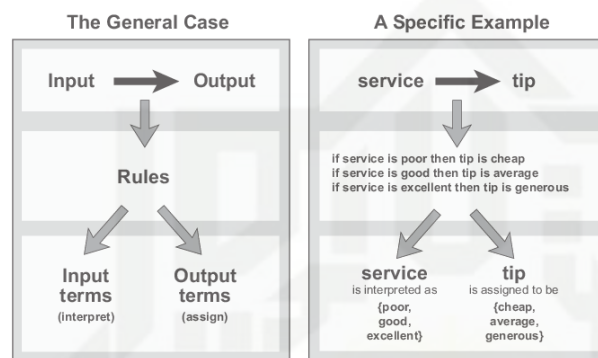
Tabel 2.2 : Respon sistem kendali PID menurut parameter[4]

Tanggapan Loop Tertutup	Waktu Naik	<i>Overshoot</i>	<i>Waktu Turun</i>	<i>Error Steady State</i>
<i>Proporsional (Kp)</i>	Menurun	Meningkat	Perubahan kecil	Menurun
<i>Integral (Ki)</i>	Menurun	Meningkat	Meningkat	Hilang
Tanggapan Loop Tertutup	Waktu Naik	<i>Overshoot</i>	Waktu Turun	<i>Error Steady State</i>
<i>Derivative</i>	Perubahan kecil	Menurun	Menurun	Perubahan kecil

2.7 Logika Fuzzy

Logika *fuzzy* merupakan sebuah logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar dan salah. Dalam teori logika *fuzzy* sebuah nilai bisa bernilai benar dan salah secara bersamaan namun berapa besar kebenaran dan kesalahan suatu nilai tergantung kepada bobot/derajat keanggotaan yang dimilikinya. Tujuan utama dari logika *fuzzy* adalah memetakan sebuah ruang input ke dalam sebuah ruang output, dan

mekanisme utama untuk melakukan ini adalah kumpulan dari pernyataan *if-then* yang disebut dengan *rules* (aturan). Semua *rule* dievaluasi secara paralel, dan orde dari *rule* tersebut tidaklah penting. *Rule* ini sangat berguna karena mengacu pada kata sifat yang menggambarkan variabel tersebut. Untuk membangun sistem yang menggunakan *rule* tersebut, maka harus didefinisikan terlebih dahulu semua istilah dan kata sifat yang menggambarkan variabel tersebut. Gambar 2.4 Merupakan diagram yang memperlihatkan proses *inference fuzzy* [16].



Gambar 2.10 Roadmap proses *inference fuzzy*[16]

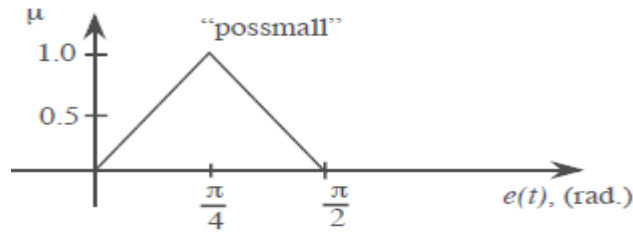
Gambar 2.9 memperlihatkan sistem logika *fuzzy* secara umum pada bagian kiri dan sistem logika *fuzzy* secara spesifik pada bagian kanan. Dari gambar diatas, dapat disimpulkan bahwa *inference fuzzy* adalah metode yang menginterpretasi nilai pada vektor input berdasarkan beberapa kumpulan *rules*, dan menetapkan nilai pada vektor *output*. [16]

2.8 Fuzzifikasi

Fuzzifikasi adalah proses mengubah nilai *crisp*/tegas variabel input menjadi nilai linguistik yang berupa derajat keanggotaan (*degree of membership*) dari setiap himpunan *fuzzy* (*fuzzy set*) variabel input. Untuk mengubah *crisp* input menjadi *fuzzy* input, terlebih dahulu harus menentukan fungsi keanggotaan (*membership function*) untuk *crisp* input, kemudian proses fuzzifikasi akan mengubah *crisp* input dan membandingkan dengan fungsi keanggotaan (*membership function*) yang telah ada untuk menghasilkan harga *fuzzy* input.

2.9 Himpunan Fuzzy

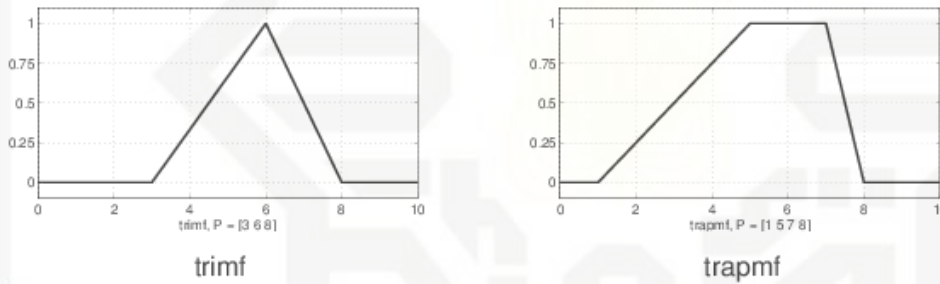
Himpunan fuzzy adalah sebuah himpunan yang anggotanya memiliki derajat keanggotaan tertentu. Setiap anggota memiliki derajat keanggotaan tertentu yang ditentukan oleh fungsi keanggotaan (*membership function*) tertentu atau disebut juga fungsi karakteristik (*characteristik function*). Berikut ini adalah contoh dari himpunan *fuzzy*:



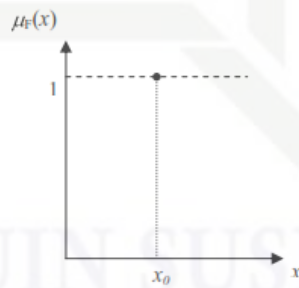
Gambar 2.11 Fungsi keanggotaan himpunan possmall[16]

2.10 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Representasi dari fungsi keanggotaan ini dapat digambarkan dengan dua bentuk umum yaitu *linear* atau garis lurus (*singleton*) dan kurva geometri diantaranya adalah kurva segitiga dan kurva trapesium yang diperlihatkan pada gambar 2. [16]



Gambar 2.12 Bentuk fungsi keanggotaan segitiga dan trapesium



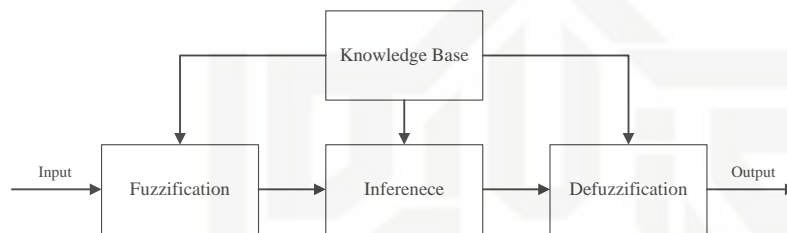
Gambar 2.13 Bentuk fungsi keanggotaan *singleton*

2.11 Pengendali Logika Fuzzy

Logika *fuzzy* merupakan cabang dari sistem kecerdasan buatan (*Artificial Intelligence*) yang mengevaluasi kemampuan manusia dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Sekitar tahun 1965, Profesor Lotfi A. Zadeh adalah guru besar pada *University of California* yang merupakan pencetus sekaligus yang memasarkan ide tentang cara

mekanisme pengolahan atau manajemen ketidak pastian yang kemudian dikenal dengan logika *fuzzy*. Teknik *fuzzy* dapat diimplementasikan dalam bidang sistem kendali, pengenalan pola, pemrosesan citra, analisis kuantitatif pada penelitian di bidang sosial, penarikan kesimpulan seperti pada *Expert System* (ES), perencanaan, prediksi, teknik pembuatan *software*, lain-lain. Dalam hal ini, akan dibahas tentang logika *fuzzy* sebagai pengendali, atau biasa disebut *Fuzzy Logic Controller* (FLC).

Secara umum, sistem pengendali logika *fuzzy* terdiri dari empat komponen utama yaitu Fuzzifikasi, *Knowledge base*, *Inference* dan Defuzzifikasi.[17]



Gambar 2.14 Blok diagram sederhana logika *fuzzy*[17]

Fuzzifikasi mengubah input yang bernilai numerik menjadi nilai linguistik variabel input *fuzzy*. *Knowledge base* terdiri dari *rule base* yang menghubungkan nilai linguistik variabel input dan *output* untuk memenuhi tujuan dari perancangan logika *fuzzy*. Proses *inference* menghubungkan input dan *output* berdasarkan *rule base* dan menghasilkan penarikan kesimpulan berupa nilai linguistik variabel *output*. Defuzzifikasi mengubah nilai linguistik variabel *output* menjadi nilai keluaran *crisp* hasil dari proses *inference*. [17]

2.12 Knowledge Base

Knowledge base merupakan komponen yang terpenting dalam merancang logika *fuzzy*. Secara umum, *knowledge base* hanya terdiri dari aturan – aturan dasar (*rule base*) yang berfungsi untuk memenuhi tujuan perancangan logika *fuzzy*. Pada pengendali logika fuzzy, *knowledge base* berkembang menjadi dua bagian yaitu: *data base* dan *fuzzy rules base*.

2.12.1 Data Base

Pada bagian *data base*, ada empat prinsip dalam mendesain parameter dari pengendali *fuzzy* yaitu : diskritisasi dan normalisasi dari *universe of discourses*, partisi input dan *output fuzzy*, dan fungsi keanggotaan dari himpunan *fuzzy*.

- a) Diskritisasi *universe of discourse*

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Universe of discourses dapat bersifat kontinyu atau diskrit. Jika *universe* bersifat kontinyu, maka diperlukan proses diskritisasi untuk menjadikan *universe* tersebut bersifat diskrit. Proses diskritisasi juga sering disebut dengan kuantisasi. Kuantisasi merupakan proses mengubah *universe* menjadi segmen – segmen penomoran. Jumlah *level* kuantisasi merupakan hal penting yang dapat mempengaruhi performansi logika *fuzzy*, sehingga diperlukan jumlah kuantisasi yang cukup besar untuk memberikan performansi yang memadai. Perlu pertimbangan dalam penentuan jumlah *level* kuantisasi dalam hal kualitas performansi *fuzzy* dan kapasitas memori pada komputer. Untuk proses diskritisasi, dibutuhkan skala pemetaan yang berfungsi untuk mengubah variabel yang diukur kedalam nilai diskrit. Pemetaan dapat berupa *uniform (linear)*, *nonuniform (nonlinear)*, atau keduanya.

b) Normalisasi *universe of discourse*

Proses ini merupakan proses diskritisasi kedalam rentang normalisasi. *Universe* normalisasi terdiri dari segmen – segmen yang mempunyai nilai terbatas.

c) Partisi ruang input dan *output fuzzy*

Sebuah variabel linguistik dalam *antecedent* dari sebuah *rule base* akan membentuk suatu ruang input *fuzzy*, sementara itu *consequent* dari sebuah *rule base* membentuk ruang *output fuzzy*. Secara umum, variabel linguistik dikaitkan dengan himpunan kata (*term set*). Partisi *fuzzy* berfungsi untuk menentukan berapa banyak kata yang seharusnya berada pada sebuah himpunan kata. Terdapat tujuh kata linguistik yang sering digunakan dalam *fuzzy inference* :

NB: *negative big*

NM: *negative medium*

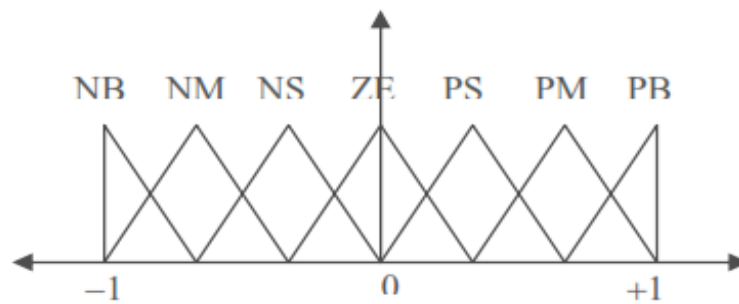
NS: *negative small*

ZE: *zero*

PS: *positive small*

PM: *positive medium*

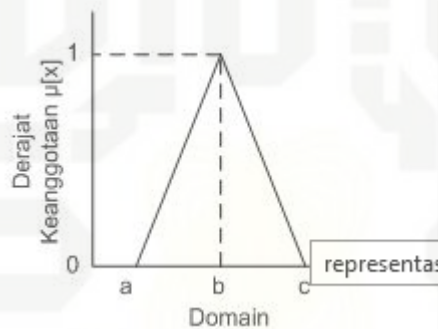
PB: *positive big*



Gambar 2.15 Contoh partisi *fuzzy* dengan kata linguistik

d) Fungsi keanggotaan dari himpunan *fuzzy*

Terdapat beberapa tipe fungsi keanggotaan seperti bentuk segitiga, trapesium dan bell. Sebagai contoh fungsi keanggotaan segitiga $\mu[x] = (2.1)$



Gambar 2.16 Contoh fungsi keanggotaan segitiga

2.12.2 Rule Base

Rule base merupakan sebuah cara untuk menyatakan suatu pernyataan kondisi berdasarkan pengetahuan manusia yang direpresentasikan dalam istilah linguistik. Pernyataan kondisi ini dinyatakan dalam bentuk :

IF premise (antecedent), THEN conclusion (consequent)

Bentuk ekspresi diatas lebih dikenal sebagai bentuk *IF-THEN rule base*. *Fuzzy rule* memiliki dua bagian yang berbeda : evaluasi *antecedent* dari sebuah *rule* (bagian IF dari *rule*) dan *implication* atau menerapkan hasil evaluasi ke *consequent rule* tersebut (bagian THEN dari *rule*).

Dalam menentukan *rule base* yang digunakan, maka diperlukan metode untuk menentukan *rule base* yang mampu memenuhi kriteria perancangan logika *fuzzy*. Terdapat dua cara pendekatan untuk menurunkan *rule base fuzzy*. Pertama, metode heuristik, yang mana *rule base* dibentuk dengan menganalisa perilaku dari proses yang dikendalikan. Penurunan ini mengandalkan pengetahuan kualitatif dari perilaku proses. Pendekatan kedua yaitu metode penentuan, yang mana metode ini

dapat menentukan secara sistematis struktur linguistik dari *rule base*. Berikut ini merupakan empat mode untuk menurunkan *rule base fuzzy*. Empat mode ini memang tidak begitu eksklusif, tetapi mode ini diperlukan untuk digabungkan sehingga didapatkan sebuah sistem yang mampu bekerja efektif.[17]

1. Pengalaman para pakar dan pengetahuan *control engineer*: pengoperasian manual dan metode kuisisioner
2. Berdasarkan aksi operator kendali: pengamatan terhadap manusia sebagai pengendali dalam hal pengoperasian input dan *output* data.
3. Berdasarkan model *fuzzy* dari sebuah proses: gambaran linguistik karakter dinamis dari suatu proses.
4. Berdasarkan pembelajaran: kemampuan untuk memodifikasi *rule base* seperti pengendali *self-organizing*.

2.13 Mekanisme Inference

Fuzzy inference dapat diartikan sebagai suatu proses pemetaan dari input yang diberikan kepada sebuah *output* menggunakan teori himpunan *fuzzy*. [19] *Fuzzy IF-THEN rule*, mengekspresikan hubungan implikasi antara premis (*antecedent*) dari himpunan *fuzzy* dan kesimpulan dari himpunan *fuzzy* (*consequent*). Berikut ini adalah penjelasan mengenai mekanisme *inference* :

1. Proses *Matching*, yaitu mengkombinasikan input yang telah difuzzifikasi terhadap masing – masing *antecedent rule base*, atau penentuan derajat keanggotaan / *degree of firing* (DOF) dari *antecedent*.
2. Jika *rule* memiliki beberapa *antecedent*, maka digunakan operator *fuzzy* AND atau OR untuk mendapatkan *single* DOF yang menyatakan hasil dari evaluasi *antecedent*.
3. *Implication*, yaitu mengaplikasikan hasil dari evaluasi *antecedent* terhadap fungsi keanggotaan dari *consequent* untuk menghasilkan sebuah *output*.

Terdapat dua metode *inference* yang sering digunakan yaitu *inference* model Mamdani dan model Sugeno. Tipe Mamdani lebih cocok digunakan pada sistem pakar dan tipe Sugeno lebih cocok digunakan untuk sistem pengendalian. Metode *inference* yang dirancang pada penelitian ini adalah metode *inference* tipe Sugeno, karena tipe Sugeno memiliki keunggulan dari segi performansi dan kemampuan adaptasinya. [19]

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.13.1 Metode Mamdani

Metode ini merupakan metode yang paling sering digunakan pada banyak literatur. Untuk dapat menggambarkan metode ini, maka dibuat sebuah sistem sederhana dengan dua *rule*, dimana masing – masing *rule* terdiri dari dua *antecedent* dan satu *consequent*. Sebuah sistem *fuzzy* dengan dua input x_1 dan x_2 (*antecedent*) dan sebuah *output* y (*consequent*) dideskripsikan oleh sejumlah r proposisi linguistik *IF-THEN* dalam bentuk Mamdani yaitu :

$$IF \ x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z \text{ is } C$$

Dimana A_1 dan A_2 adalah himpunan *fuzzy* yang menyatakan bagian *antecedent* ke- k dan B adalah himpunan *fuzzy* yang menyatakan bagian *consequent* ke- k . Pada metode *inference* Mamdani, terdapat dua tahap *inference* berbeda untuk sistem *fuzzy* yang memiliki dua input atau lebih yaitu metode *inference max-min* dan *max-product*. [19]

2.13.2 Metode Sugeno

Metode Sugeno sangat mirip dengan metode Mamdani, hanya saja metode ini menggunakan *singleton* sebagai fungsi keanggotaan pada *rule consequent*. *Fuzzy singleton* adalah himpunan *fuzzy* dengan fungsi keanggotaan berupa sebuah garis lurus tunggal pada daerah *fuzzy* dan bernilai nol pada daerah lain. Metode Sugeno sangat mirip dengan metode Mamdani, yang mana Sugeno hanya melakukan perubahan pada *rule consequent*. Metode Sugeno menggunakan fungsi matematika pada variabel input. Format *rule base* untuk *rule* dengan dua input x dan y dan sebuah *output* z pada metode Sugeno adalah :

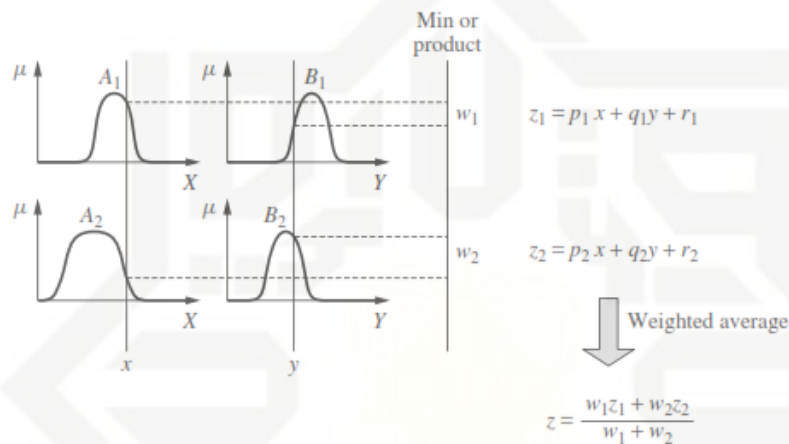
$$IF \ x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z \text{ is } f(x,y)$$

Dimana x, y dan z adalah variabel linguistik; A dan B adalah himpunan *fuzzy* X dan Y ; dan $f(x,y)$ adalah fungsi matematika. Ketika $f(x,y)$ merupakan sebuah konstanta, maka sistem *inference* disebut sebagai *zero order* Sugeno model. Ketika $f(x,y)$ merupakan fungsi linear x dan y , maka sistem *inference* disebut sebagai *first order* Sugeno model. Penelitian yang dilakukan oleh Jang dkk, menunjukkan bahwa *output* dari *zero order* Sugeno model akan menghasilkan fungsi yang halus dari variabel inputnya selama fungsi keanggotaan pada *antecedent* memiliki tumpang tindih (*overlap*) yang cukup. Pada model Mamdani, tumpang tindih fungsi

keanggotaan dalam *consequent* tidak memiliki efek yang pasti (*decisive effect*) pada perilaku *smoothness* sistem. Hanya *overlap* pada fungsi keanggotaan *antecedent* saja yang menentukan perilaku *smoothness* pada sistem.[20] Zero order Sugeno model memiliki bentuk *rule base* sebagai berikut :

$$IF\ x\ is\ A\ AND\ y\ is\ B\ THEN\ z\ is\ k$$

Dimana k adalah konstanta. Dalam hal ini, *output* dari *fuzzy rule* adalah nilai konstanta. Dengan kata lain, semua fungsi keanggotaan *consequent* direpresentasikan oleh *singleton*.0



Gambar 2.17 Model fuzzy Sugeno

Pada model Sugeno, masing – masing *rule* memiliki *output crisp* yang diberikan oleh sebuah fungsi. Oleh karena itu, seluruh *output* didapat melalui metode defuzzifikasi *weighted average*, seperti pada gambar 2.13 Proses ini bertujuan untuk menghindari konsumsi waktu defuzzifikasi yang diperlukan pada model Mamdani.

2.14 Defuzzifikasi

Defuzzifikasi merupakan proses konversi nilai *fuzzy* yang dengan fungsi keanggotaan hasil gabungan proses *inference* menjadi nilai numerik pada variabel *output*.

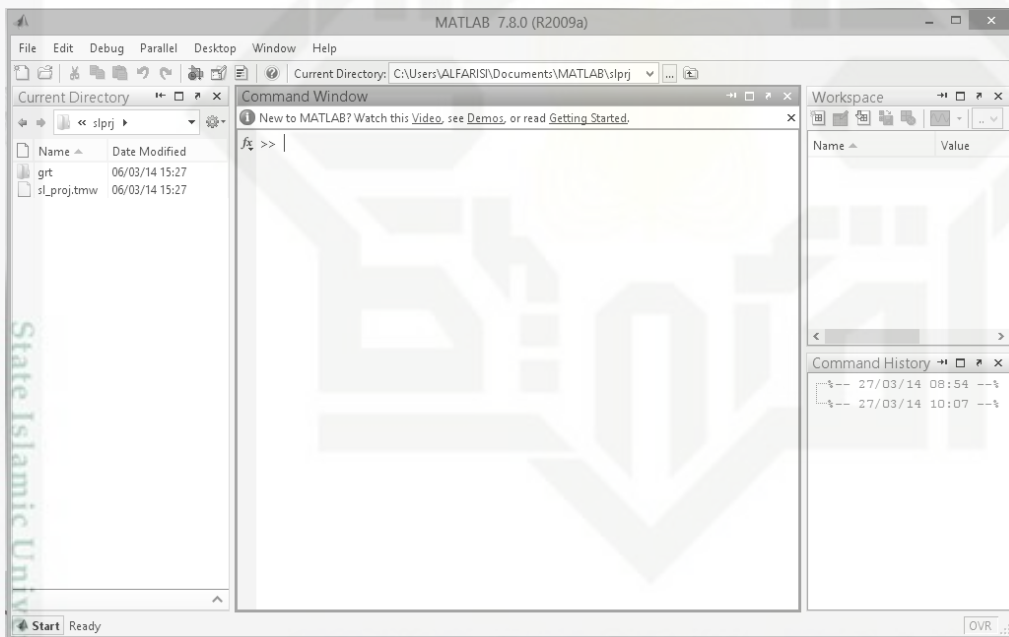
2.15 Program Matlab

Matlab merupakan singkatan dari *Matrix Laboratory* yang berarti bahasa pemrograman level tinggi (semakin tinggi level bahasa pemrograman maka semakin mudah cara menggunakannya) dengan kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan

tunggal. Matlab memberikan sistem interaktif yang menggunakan konsep *array/matrix* sebagai variabel elemennya tanpa membutuhkan pendeklarasian *array*.

Matlab dikembangkan oleh Mathwork pada tahun 1970. Aplikasi Matlab itu sendiri banyak digunakan dalam bidang yang membutuhkan perhitungan matematika yang rumit, dimana seluruh operasi perhitungan dalam Matlab berupa operasi matrik. Matlab dapat menampilkan hasil perhitungan dalam bentuk *plot* grafik dan dapat juga dirancang menggunakan GUI (*Graphical User Interface*) yang kita rancang. Pada *software* Matlab 7.8.0 (R2009a) terdapat beberapa bagian penting yang digunakan dalam menjalankan program yaitu:

1. *Command window* digunakan untuk mengetik fungsi yang diinginkan.
2. *Command history* berfungsi agar fungsi yang telah digunakan sebelumnya dapat digunakan kembali.
3. *Workspace* digunakan untuk membuat variabel yang ada dalam Matlab.



Gambar 2.18 Tampilan Matlab 7.8.0 (R2009a)

2.16 Simulink Matlab

Simulink adalah sebuah kumpulan aplikasi dalam Matlab untuk melakukan modeling, simulasi, dan untuk melakukan analisis dinamik pada suatu sistem. Program simulink memudahkan *user* untuk membuat suatu simulasi lebih interaktif. Tiruan sistem diharapkan mempunyai perilaku yang sangat mirip dengan sistem fisik. Jika digunakan dengan benar, simulasi akan membantu proses analisis dan desain sistem.

Simulink dalam *matlab* juga dapat menunjukkan performansi sistem dalam bentuk grafik dua dimensi ataupun tiga dimensi. Dalam perancangannya *user* dibantu oleh blok-blok diagram yang dapat dengan mudah diatur sedemikian rupa, sesuai dengan model matematis dari sistem atau *plant* yang yang dikendalikan.



Hak-hak Cipta Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.