

BAB II

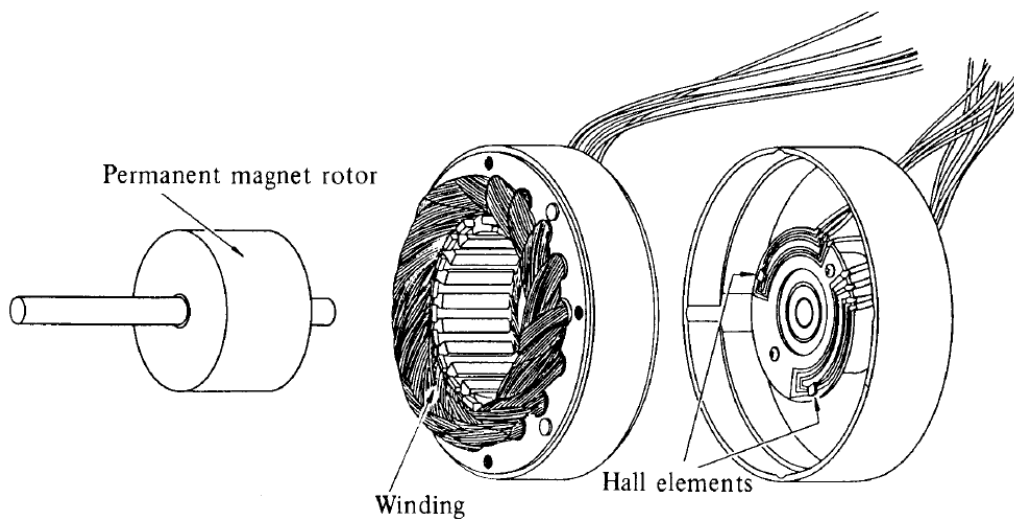
KAJIAN PUSTAKA

Pada bab ini dijelaskan teori penunjang yang digunakan dalam penulisan dengan tujuan untuk memudahkan dalam memahami dasar teori maupun dasar-dasar perancangan alat yang dijabarkan sebagai berikut:

2.1. Motor *Brushless Direct Current* (BLDC)

Motor arus searah tanpa sikat, atau dapat disebut motor *Brushless Direct Current* (BLDC) adalah salah satu jenis mesin elektrik yang berfungsi untuk mengubah energi elektrik menjadi energi kinetik. Motor BLDC menggunakan sumber arus searah atau *Direct Current* (DC) sebagai sumber energi utama yang kemudian diubah menjadi arus bolak – balik 3 (tiga) fasa atau *three-phase Alternating Current* (AC) dengan menggunakan inverter motor tiga fasa. Tujuan pemberian arus bolak – balik tiga fasa adalah untuk menciptakan medan magnet putar (Dharmawan, 2009).

Konstruksi motor BLDC terdiri dari beberapa bagian, yaitu kumparan yang dililitkan pada inti besi sebagai *stator* dan satu atau lebih magnet permanen sebagai *rotor*. Oleh karena tidak adanya *brush* pada motor BLDC, diperlukan 3 (tiga) buah sensor *hall* dan atau *encoder* untuk menentukan *timing* komutasi yang tepat pada motor BLDC, sehingga didapatkan torsi dan kecepatan yang konstan. Pada sensor *Hall*, *timing* komutasi ditentukan dengan cara mendeteksi medan magnet *rotor* dengan menggunakan 3 (tiga) buah sensor *hall* untuk mendapatkan 6 kombinasi *timing* yang berbeda, sedangkan pada *encoder*, *timing* komutasi ditentukan dengan cara menghitung pola yang ada pada *encoder* (Kenjo & Nagamori, 1985). Bagian – bagian dari konstruksi motor BLDC dapat dilihat dalam Gambar 2.1.



Gambar 2.1 Tampak Motor BLDC setelah dibongkar: rotor magnet permanen, kumparan, dan sensor hall.

Sumber: Kenjo & Nagamori, 1985

2.1.1. Cara Kerja Motor Brushless DC (BLDC)

Motor BLDC dapat bekerja ketika *stator* yang terbuat dari kumparan diberikan arus bolak – balik 3 (tiga) fasa. Arus yang melewati kumparan pada *stator* mengakibatkan timbulnya medan magnet (B):

$$B = \frac{\mu i N}{2l} \quad (2-1)$$

dengan N = jumlah lilitan

i = arus (A)

l = panjang lilitan (m)

μ = permeabilitas bahan ($\Omega\text{s/m}$)

Karena arus yang diberikan berupa arus bolak – balik 3 fasa 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 n_s :

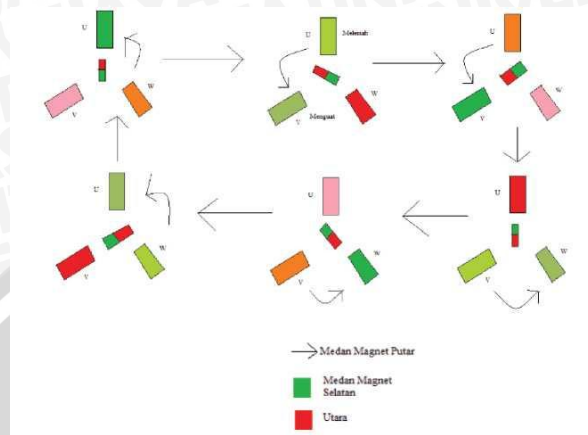
$$n_s = \frac{120f}{p} \quad (2-2)$$

dengan n_s = putaran *stator* (rpm)

f = frekuensi (Hz)

p = jumlah kutub

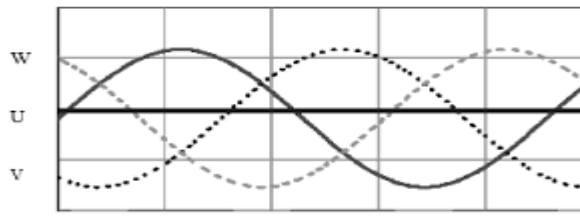
Medan putar magnet *stator* timbul akibat adanya perubahan polaritas pada *stator* U, V, dan W. Perubahan polaritas ini terjadi akibat adanya arus yang mengalir pada *stator* berupa arus bolak – balik yang memiliki polaritas yang berubah - ubah. Perubahan medan magnet putar *stator* dan perputaran *rotor* dapat dilihat dalam Gambar 2.2.



Gambar 2. 2 Medan Magnet Putar *Stator* dan Perputaran *Rotor*

Sumber: Dharmawan, 2009

Ketika *stator* U diberikan tegangan negatif maka akan timbul medan magnet dengan polaritas negatif 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 positif akan berputar menuju medan magnet *stator* U dan V. Selanjutnya ketika V memiliki medan magnet negatif dan U serta W memiliki medan magnet positif, bagian positif 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* akan terjadi secara terus menerus sehingga menciptakan medan putar magnet *stator*. Akibatnya, magnet permanen yang terdapat di dalam *rotor* akan berputar mengikuti medan putar magnet *stator* tersebut. Hal inilah yang menyebabkan *rotor* pada motor BLDC dapat berputar. Perubahan polaritas tegangan *stator* pada motor BLDC dapat dilihat dalam Gambar 2.3.



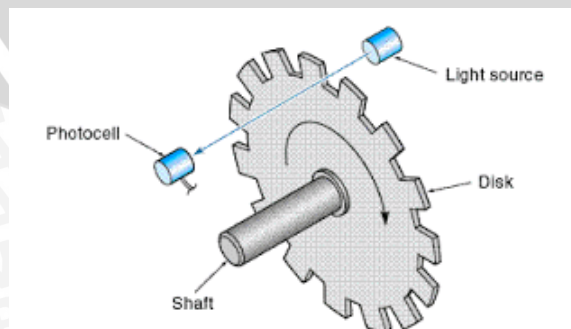
Gambar 2.3 Tegangan *stator* BLDC

Sumber: Dharmawan, 2009

2.2. Rotary Encoder

Rotary encoder adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan sinyal pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah, sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder*.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. *Light Emitting Diode* atau LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu *photo-transistor* diletakkan sehingga *photo-transistor* ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau alat berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai *photo-transistor* melalui lubang-lubang yang ada, maka *photo-transistor* akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Gambar 2.4 menunjukkan desain sederhana dari *rotary encoder*. Banyaknya deretan pulsa yang dihasilkan pada satu putaran akan menentukan akurasi *rotary encoder* tersebut, dengan demikian, semakin banyak deretan pulsa yang dihasilkan pada satu putaran, semakin akurat pula pembacaan *rotary encoder* tersebut.



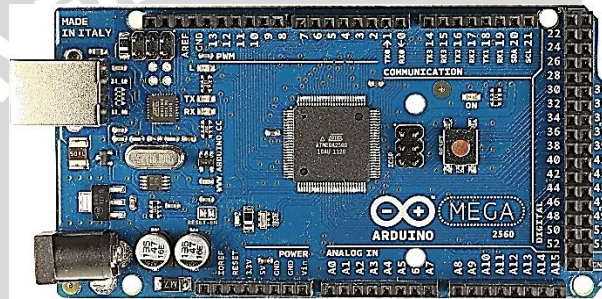
Gambar 2.4 Desain umum *rotary encoder*

Sumber: Lukman, 2012

2.3. Mikrokontroler Arduino Mega 2560

Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai *input* dan *output* serta kontrol dengan program yang bisa ditulis dan dihapus. Cara kerja mikrokontroler adalah dengan membaca, memproses, dan menulis data. Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis.

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega328. *Board* ini memiliki 54 digital *input/output* pin (14 pin dapat digunakan sebagai *output* PWM), 16 *input* analog, 16 MHz osilator kristal, koneksi USB, jack listrik, *header* ICSP, dan tombol *reset*. Contoh perangkat mikrokontroler Arduino dapat dilihat dalam Gambar 2.5.



Gambar 2.5 Mikrokontroler Arduino Mega 2560

Sumber : www.electroschematics.com

Arduino Mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Daya eksternal (non-USB) dapat berasal baik dari adaptor AC ke DC maupun baterai. Adaptor tersebut dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2.1 mm pada konektor power. Ujung kepala dari baterai dapat dimasukkan ke dalam pin *header* Gnd dan V_{in} dari konektor power. Arduino dapat beroperasi dengan catu daya eksternal 6 sampai 20 volt. Namun jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak papan. Kisaran yang disarankan adalah 7 sampai 12 volt.

ATmega 2560 memiliki 256 KB (dengan 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM. Masing-masing dari 54 pin digital di Arduino Mega 2560 dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (), beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara default terputus) dari 20-50 KOhm. Selain itu, beberapa pin memiliki fungsi khusus:

1. *Serial*: 0 (RX) dan 1 (TX); *Serial* 1: 19 (RX) dan 18 (TX); *Serial* 2: 17 (RX) dan 16 (TX); *Serial* 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan

mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.

2. *Eksternal interrupts*: pin 2 (*interrupt 0*), pin 3 (*interrupt 1*), pin 18 (*interrupt 5*), pin 19 (*interrupt 4*), pin 20 (*interrupt 3*), dan pin 21 (*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
3. PWM: 0 - 13. Menyediakan output PWM 8-bit dengan fungsi *analogWrite ()*.
4. *SPI*: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi *SPI* menggunakan *SPI library*.
5. *LED*: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai *high* LED menyala dan ketika pin bernilai *low* LED mati.
6. I^2C : 20 (SDA) dan 21 (SCL). Dukungan I^2C (TWI) komunikasi menggunakan *wire*.

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. Arduino Mega 2560 menyediakan 4 UART TTL (5V) untuk komunikasi serial. Sebuah Arduino Mega 2560 sebagai saluran komunikasi serial melalui USB dan sebagai *port virtual* com untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan driver USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun pada Windows diperlukan sebuah file inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim dari atau ke papan Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

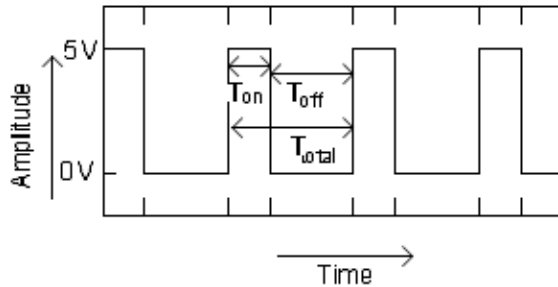
Sebuah *Software Serial Library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board Arduino Mega 2560. Arduino Mega 2560 juga mendukung I^2C (TWI) dan komunikasi *SPI*. Perangkat lunak Arduino termasuk perpustakaan kawat untuk menyederhanakan penggunaan bus I^2C .

2.4. Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) adalah metode yang digunakan untuk mengatur kecepatan dari motor BLDC, di mana kecepatan motor BLDC tergantung pada besarnya *duty cycle* yang diberikan pada motor BLDC tersebut, yang akan mempengaruhi tahanan rata-rata (V_{rms}) motor.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada motor. Persamaan

untuk perhitungan *duty cycle* ditunjukkan pada persamaan (2-3) dengan T_{on} adalah periode logika tinggi, dan T adalah periode keseluruhan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.6.



Gambar 2.6 Sinyal PWM secara umum.

$$\text{Duty Cycle} = \frac{T_{on}}{T} \times 100\% \quad (2-3)$$

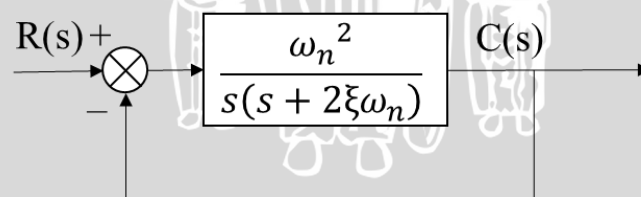
dengan T_{on} = lebar pulsa saat logika tinggi (s)

T = periode pulsa (s)

Timer atau *counter* yang digunakan pada PWM ini yaitu *timer* atau *counter* 3 (16 bit) dengan metode *phase-correct* PWM dan *prescaler factor* (N) yaitu 256.

2.5 Output Sistem Orde Dua

Diagram blok sistem orde dua dengan fungsi alihnya dapat dilihat dalam Gambar 2.7.



Gambar 2.7 Sistem Orde Dua

$$\begin{aligned} \frac{C(s)}{R(s)} &= \frac{\omega_n^2}{s(s + 2\xi\omega_n)} \\ &= \frac{\omega_n^2}{1 + \frac{\omega_n^2}{s(s + 2\xi\omega_n)}} \\ &= \frac{\omega_n^2}{s(s + 2\xi\omega_n) + \omega_n^2} \\ &= \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \end{aligned} \quad (2-4)$$

Dari persamaan (2-4), dapat ditarik akar-akar penyebut fungsi alih atau persamaan karakteristik sebagai berikut:

$$s_{1,2} = \frac{2\xi\omega_n \pm \sqrt{(2\xi\omega_n)^2 - 4\omega_n^2}}{2}$$

$$s_{1,2} = -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1}$$

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{\xi^2 - 1}$$

$$s_{1,2} = -\xi\omega_n \pm j\omega_d \quad (2-5)$$

dengan ξ = rasio peredaman sistem (*damping ratio*)

ω_n = frekuensi natural/alamiah tak teredam

ω_d = frekuensi natural/alamiah teredam

Kelakuan dinamik sistem orde dua dapat dideskripsikan dalam suku dua parameter yaitu ξ dan ω_n . Jika ($0 < \xi < 1$), maka *pole* loop tertutup merupakan konjugat kompleks dan terletak pada sebelah kiri bidang *s*. Dalam hal ini, sistem dikatakan kurang teredam dan *output* peralihan berosilasi. Jika ($\xi = 0$), sistem dikatakan tidak teredam, karena *output* peralihan tidak teredam, dan tidak tercapai keadaan *steady-state*. Jika ($\xi = 1$), maka sistem dikatakan teredam kritis. Jika ($\xi > 1$), sistem dikatakan dalam terlalu teredam. *Output* peralihan sistem teredam kritis dan sistem terlalu teredam tidak berosilasi.

Dalam sistem orde dua, sebuah *input unit-step* pada sistem akan menghasilkan *output* yang berbeda, tergantung dari parameter ξ . Terdapat tiga keadaan yang berbeda yaitu keadaan kurang teredam ($0 < \xi < 1$), teredam kritis ($\xi = 1$), dan terlalu teredam ($\xi > 1$).

2.5.1 Keadaan Kurang Teredam / Underdamped ($0 < \xi < 1$)

Pada keadaan kurang teredam, persamaan (2-4) dapat ditulis sebagai berikut:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s + \xi\omega_n + j\omega_d)(s + 2\xi\omega_n - j\omega_d)} \quad (2-6)$$

Dengan $\omega_d = \omega_n\sqrt{1 - \xi^2}$, dan ω_d adalah frekuensi natural/alamiah teredam. Untuk *input unit-step* atau $R(s) = \frac{1}{s}$, maka:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \quad (2-7)$$

Output sistem tersebut bisa diperoleh dengan menggunakan Transformasi Laplace balik jika $C(s)$ ditulis dalam bentuk berikut:

$$C(s) = \frac{1}{s} - \frac{s + 2\xi\omega_n}{(s^2 + 2\xi\omega_n s + \omega_n^2)}$$

$$C(s) = \frac{1}{s} - \frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2} - \frac{\xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2} \quad (2-8)$$

Mengacu pada tabel transformasi Laplace, diketahui bahwa:

$$\mathcal{L}^{-1}\left[\frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}\right] = e^{-\xi\omega_n t} \cos \omega_d t$$

$$\mathcal{L}^{-1}\left[\frac{\omega_d}{(s + \xi\omega_n)^2 + \omega_d^2}\right] = e^{-\xi\omega_n t} \sin \omega_d t$$

oleh karena itu, transformasi laplace balik dari persamaan (2-8) adalah sebagai berikut:

$$\mathcal{L}^{-1}[C(s)] = c(t)$$

$$c(t) = 1 - e^{-\xi\omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1 - \xi^2}} \sin \omega_d t \right)$$

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1 - \xi^2}} \sin \left(\omega_d t + \tan^{-1} \frac{\sqrt{1 - \xi^2}}{\xi} \right), \text{ untuk } t \geq 0 \quad (2-9)$$

Sinyal kesalahan atau sinyal *error* adalah $e(t) = r(t) - c(t)$, di mana $r(t) = 1$ dan $c(t)$ adalah persamaan (2-9), sehingga:

$$e(t) = 1 - \left(1 - e^{-\xi\omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1 - \xi^2}} \sin \omega_d t \right) \right)$$

$$e(t) = e^{-\xi\omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1 - \xi^2}} \sin \omega_d t \right), \text{ untuk } t \geq 0 \quad (2-10)$$

Dari persamaan (2-10), dapat diketahui bahwa sinyal *error* memiliki bentuk gelombang sinusoidal teredam, dan pada keadaan *steady-state* atau $t = \infty$, tidak ada *error* antara *input* dan *output*.

Jika rasio peredaman sama dengan nol ($\xi = 0$), *output* sistem menjadi tidak teredam, dan osilasi terjadi terus – menerus. *Output* $c(t)$ untuk rasio peredaman nol dapat didapatkan dengan cara mensubstitusikan $\xi = 0$ pada persamaan (2-10), menghasilkan persamaan berikut:

$$c(t) = 1 - \cos \omega_n t, \text{ untuk } t \geq 0 \quad (2-11)$$

Dari persamaan (2-10), dapat diketahui bahwa ω_n adalah frekuensi natural/alamiah tak teredam, yaitu frekuensi di mana sistem akan berosilasi ketika peredaman dikurangi menjadi nol (0).

2.5.2 Teredam Kritis / Critically Damped ($\zeta = 1$)

Dalam hal ini jika dua *pole* dalam persamaan (2-4) sama, maka sistem dapat didekati dengan bentuk teredam kritis.

Untuk *input unit-step* atau $R(s) = \frac{1}{s}$, $C(s)$ dapat ditulis sebagai berikut:

$$C(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad (2-12)$$

Dengan melakukan transformasi Laplace balik pada persamaan (2-12), maka didapatkan $c(t)$, yaitu:

$$c(t) = 1 - e^{-\omega_n t} (1 + \omega_n t) \quad t \geq 0 \quad (2-13)$$

2.5.3 Terlalu Tereدام / Overdamped ($\zeta > 1$)

Dalam hal ini *pole* dalam persamaan (2-4) adalah bilangan nyata / real negatif yang tidak sama.

Untuk *input unit-step* atau $R(s) = \frac{1}{s}$, $C(s)$ dapat ditulis sebagai berikut:

$$C(s) = \frac{\omega_n^2}{(s + \xi\omega_n + \omega_n\sqrt{\xi^2 - 1})(s + \xi\omega_n - \omega_n\sqrt{\xi^2 - 1})s} \quad (2-14)$$

Dengan melakukan transformasi Laplace balik pada persamaan (2-14), maka didapatkan $c(t)$, yaitu:

$$c(t) = 1 + \frac{1}{2\sqrt{\xi^2 - 1}(\xi + \sqrt{\xi^2 - 1})} e^{-(\xi + \sqrt{\xi^2 - 1})\omega_n t} - \frac{1}{2\sqrt{\xi^2 - 1}(\xi - \sqrt{\xi^2 - 1})} e^{-(\xi - \sqrt{\xi^2 - 1})\omega_n t}$$

$$c(t) = 1 + \frac{\omega_n}{2\sqrt{\xi^2 - 1}} \left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2} \right) \quad t \geq 0 \quad (2-15)$$

di mana $s_1 = \xi + \sqrt{\xi^2 - 1}\omega_n$ dan $s_2 = \xi - \sqrt{\xi^2 - 1}\omega_n$. Oleh karena itu, *output* $c(t)$ terdiri dari dua suku eksponensial menurun.

2.6 Output Peralihan

Sistem dengan tenaga tidak dapat memberikan *output* seketika dan akan menunjukkan *output* peralihan walaupun diberi masukan ataupun gangguan. Karakteristik unjuk kerja sistem kontrol yang diinginkan dicirikan oleh suku *output* peralihan terhadap *input unit-step* karena hal itu mudah dilakukan dan cukup drastis. Jika tanggapan terhadap *input unit-step* diketahui, maka secara matematis dapat dihitung *output* untuk *input* yang lain.

Output peralihan sistem kontrol sering menunjukkan osilasi teredam sebelum mencapai keadaan *steady-state*. *Output* peralihan sistem kontrol terhadap *input unit-step* umumnya dikelompokkan sebagai berikut, dan contoh *Output* peralihan dapat dilihat dalam Gambar 2.8

1) Delay Time / Waktu Tunda, t_d

Waktu yang dibutuhkan oleh *output* untuk mencapai setengah harga akhir pada saat lonjakan pertama.

2) Rise Time / Waktu Naik, t_r

Waktu yang dibutuhkan oleh *output* agar bertambah dari 10% menjadi 90% dari nilai akhir.

3) Peak Time / Waktu Puncak, t_p

Waktu yang dibutuhkan oleh *output* untuk mencapai puncak pertama lonjakan (maksimum).

4) Maximum (Percent) Overshoot / (Persen) Lonjakan Maksimum, M_p

Merupakan nilai puncak kurva *output* diukur dari satu (1). Jika nilai akhir pada keadaan *steady-state* tidak sama dengan satu, maka digunakan nilai persen lonjakan maksimum, yaitu:

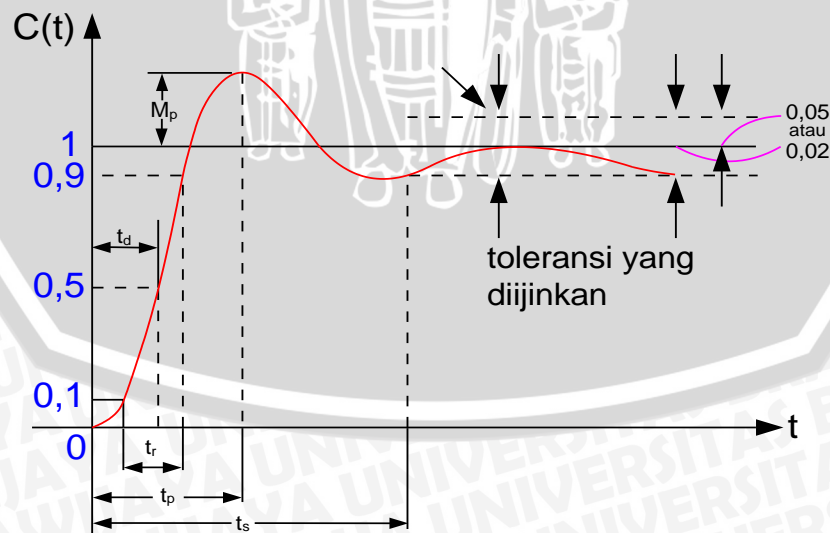
$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2-16)$$

dengan $c(t_p)$ = nilai *output* pada saat lonjakan maksimum.

$c(\infty)$ = nilai *output* pada saat keadaan *steady-state*.

5) Settling Time / Waktu Turun, t_s

Waktu yang dibutuhkan oleh *output* untuk mencapai harga tertentu dan tetap dalam range nilai akhir (biasanya 5% atau 2%).



Gambar 2.8 Output *Unit-step* Sistem Orde Dua

2.7 Kontroler

Kontroler adalah salah satu komponen dalam sebuah sistem kontrol yang berfungsi untuk mengolah sinyal *input* acuan atau *setpoint* dan sinyal *error* menjadi sinyal kontrol. Sistem kontrol dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem kontrol adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika diberi beban (Ogata, 2010). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam menanggapi *input*, dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

Suatu sistem dikatakan stabil jika sebuah sistem diberi beban, maka sistem tersebut akan kembali ke keadaan *steady-state* di mana *output* berada dalam keadaan tetap seperti tidak ada beban. Sistem dikatakan tidak stabil jika *output*-nya berosilasi terus menerus ketika diberi suatu beban. Karena suatu sistem kontrol biasanya melibatkan penyimpanan energy, maka *output* sistem tidak dapat mengikuti *input* secara serentak, tapi menunjukkan *output* peralihan sebelum mencapai keadaan *steady-state*

2.7.1. Kontroler Proporsional (P)

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat respons. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ ditunjukkan pada persamaan berikut:

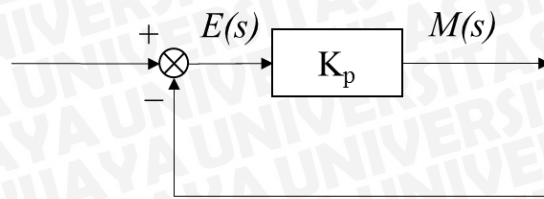
$$m(t) = K_p e(t)$$

atau, dalam besaran transformasi Laplace ditunjukkan pada persamaan berikut:

$$\frac{M(s)}{E(s)} = K_p \quad (2-17)$$

dengan K_p = penguatan proporsional.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata, 2010). Diagram blok kontroler proporsional ditunjukkan pada Gambar 2.9.



Gambar 2.9 Diagram Blok Kontroler Proporsional

Sumber: Ogata, 2010

2.7.2. Kontroler Integral (I)

Kontroler integral memiliki kemampuan untuk mengurangi *offset* yang ditinggalkan oleh kontroler proporsional. Harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$. Persamaan parameter kontroler integral K_i ditunjukkan dalam persamaan berikut:

$$\frac{dm(t)}{dt} = K_i e(t)$$

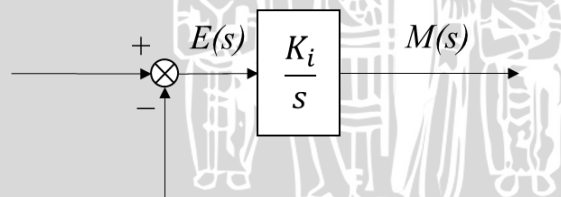
$$m(t) = K_i \int e(t) dt$$

$$\frac{M(s)}{E(s)} = \frac{K_i}{s}$$

(2-18)

dengan K_i = penguatan integral.

Diagram blok kontroler integral ditunjukkan pada Gambar 2.10



Gambar 2.10 Diagram Blok Kontroler Integral

Sumber: Ogata, 2010

2.7.3. Kontroler Proporsional – Integral (PI)

Gabungan aksi kontrol proporsional dan integral mempunyai keunggulan dibandingkan dengan masing-masing dari aksi kontrol tersebut. Masing – masing kontroler baik P maupun I berfungsi untuk mempercepat reaksi sistem, dan menghilangkan *offset*. Oleh karena itu, kontroler PI dapat pula disebut sebagai *lag compensator*. Persamaan kontroler PI ini dapat dinyatakan dalam persamaan berikut:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot \int_0^t e(t) dt$$

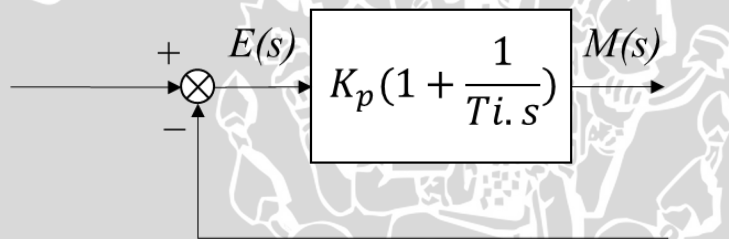
dan dalam transformasi Laplace dinyatakan sebagai berikut:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (2-19)$$

dengan $T_i = \frac{K_p}{K_i} =$ konstanta waktu integral.

Sebuah kontroler PI memiliki *zero* pada $s = -1/T_i$ dan *pole* pada $s = 0$. Maka, karakteristik dari kontroler PI adalah adanya penguatan tak hingga pada saat frekuensi sama dengan 0. Karakteristik tersebut meningkatkan karakteristik *steady-state*. Namun, dengan ditambahkan aksi kontroler PI pada sistem menaikkan tipe bilangan yang dikompensasi oleh sistem sebesar 1, dan menyebabkan sistem yang terkompensasi menjadi kurang stabil, atau bahkan membuat sistem tidak stabil. Maka dari itu, nilai dari K_p dan T_i harus ditentukan secara benar agar mendapatkan respons transien yang baik.

Diagram blok kontroler PI ditunjukkan pada gambar 2.11



Gambar 2.11 Diagram Blok Kontroler PI

Sumber: Ogata, 2010

2.8. Logika Fuzzy

Secara umum, logika *fuzzy* dapat didefinisikan sebagai logika bernilai banyak yang memiliki sifat khusus yang ditujukan untuk memodelkan suatu ketidakjelasan dalam sebuah fenomena, yang di mana nilai kebenarannya dapat dinyatakan dalam suatu bilangan real antara 0 dan 1. Salah satu kegunaan dari logika *fuzzy* adalah penjelasan dari konsep kebenaran parsial, di mana suatu nilai kebenaran dapat berada di antara sama sekali salah dan sama sekali benar (Novák, 1999).

2.8.1. Fungsi Keanggotaan

Suatu himpunan *fuzzy* A dalam semesta pembicaraan U dinyatakan dengan fungsi keanggotaan, μ_A yang harganya berada dalam interval [0,1] (Putra, 2015). Secara matematis hal ini dinyatakan dengan:

$$\mu_A : U \rightarrow [0,1]$$

a) Fungsi keanggotaan berbentuk segitiga didefinisikan sebagai berikut:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u - a}{b - a} & a \leq u \leq b \\ \frac{c - u}{c - b} & b \leq u \leq c \\ 0 & u > c \end{cases}$$

b) Fungsi keanggotaan berbentuk trapesium didefinisikan sebagai berikut

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{u - a}{b - a} & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \frac{d - u}{d - c} & c \leq u \leq d \\ 0 & d \leq u \end{cases}$$

2.8.2. Fuzzifikasi

Fuzzifikasi adalah proses dimana mengolah masukan berupa bilangan *crisp* menjadi himpunan *fuzzy*. Proses tersebut diperlukan karena apa yang kita yakini sebagai bentuk *crisp* dan deterministik sebenarnya tidak deterministik sama sekali.

Sebagai contoh, jika ada proses atau pengukuran kebisingan, kita mungkin ingin untuk menjelaskan hal ini dengan menciptakan himpunan *fuzzy* untuk takaran terukur dari pada dengan asumsi mereka akurat yang diukur.

Ada beberapa strategi yang dapat dilakukan dalam fuzzifikasi, diantaranya:

1. *Fuzzy Singletone*
2. *Fuzzy Number*
3. *Hybrid Fuzzy/ Bilangan Acak*

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi dapat menggunakan cara sebagai berikut:

1. Himpunan *fuzzy* dengan distribusi simetris.

2. Gunakan himpunan *fuzzy* dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Mengatur himpunan *fuzzy* agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium.

2.8.3. Aturan Fuzzy

Aturan *fuzzy* adalah bagian yang menggambarkan dinamika suatu sistem terhadap *input* yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat (Putra, 2015).

Dalam kontrol logika *fuzzy*, aturan kontrol *fuzzy* berbentuk aturan “IF – THEN”. Untuk sebuah sistem *Multi Input Single Output* (MISO) basis aturan pengendalian *fuzzy* berbentuk seperti berikut:

Rule 1 IF X is A_1 AND Y is B_1 THEN Z is C_1

Rule 2 IF X is A_2 AND Y is B_2 THEN Z is C_2

·
·
·

Rule n IF X is A_n AND Y is B_n THEN Z is C_n

dengan X, Y, Z merupakan variabel linguistik, dimana X dan Y merupakan variabel masukan, dan Z merupakan variabel keluaran sistem. A_n , B_n , dan C_n merupakan nilai linguistik dari X, Y, dan Z (Putra, 2015).

2.8.4. Defuzzifikasi

Defuzzifikasi adalah proses kebalikan dari fuzzifikasi, yaitu sebuah proses untuk mengolah sebuah masukan *fuzzy* menjadi keluaran *crisp*. Proses defuzzifikasi diperlukan untuk mendapatkan nilai numerik dari data *fuzzy* yang dihasilkan dari proses inferensi.

Proses defuzzifikasi dinyatakan sebagai berikut :

$$y_0 = \text{defuzzifier}(y)$$

dengan y = aksi kontrol *fuzzy*

y_0 = aksi kontrol *crisp*

defuzzifier = operator defuzzifikasi.

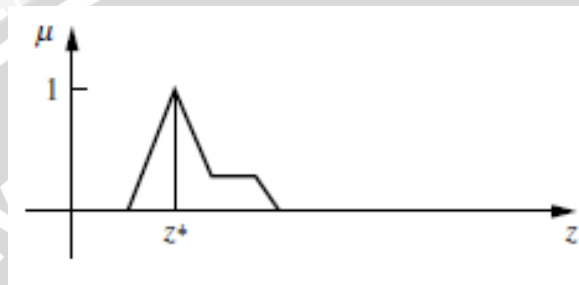
Ada beberapa cara yang bisa dilakukan dalam proses defuzzifikasi antara lain:

1. Prinsip *Max Membership*

Max membership bisa dikatakan juga sebagai metode ketinggian. Dalam metode ini terbatas pada keluaran yang memuncak. *Max membership* dapat dinyatakan dalam persamaan

$$\mu_{C\sim}(z^*) \geq \mu_{C\sim}(z) \text{ for all } z \in Z$$

dengan z^* = nilai defuzzifikasi yang dapat dilihat pada gambar 2.12.



Gambar 2.12 Defuzzifikasi Max Membership

Sumber: Ross, 2004

2. Metode *Centroid*

Metode yang juga bisa disebut juga *centre of gravity* tersebut merupakan metode paling umum dan menarik untuk digunakan. (Sugeno,1985; Lee, 1990). Metode tersebut dapat dijabarkan dalam persamaan sebagai berikut:

$$z^* = \frac{\int \mu_{\underline{C}}(z) \cdot z \, dz}{\int \mu_{\underline{C}}(z) \, dz}$$

dengan \int menyatakan proses integrasi aljabar. Metode tersebut dapat dinyatakan secara grafik pada gambar 2.13.



Gambar 2.13 Defuzzifikasi Centroid

Sumber: Ross, 2004

3. Metode Weighted Average

Metode *weighted average* sangat sering digunakan pada aplikasi *fuzzy* karena cara komputasinya sangat efisien. Sayangnya metode tersebut biasanya terbatas pada penggunaan untuk output membership yang simetris. Metode tersebut dapat dijabarkan dalam persamaan sebagai berikut:

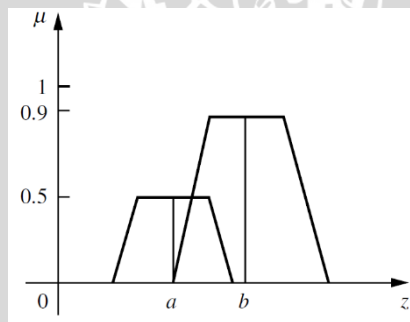
$$z^* = \frac{\sum \mu_{\underline{c}}(z) \cdot z}{\sum \mu_{\underline{c}}(z)}$$

dengan \sum menyatakan jumlah aljabar

z = titik pusat dari masing – masing membership function

Metode *weighted average* didapatkan dengan cara memboboti masing – masing *membership function* output dengan nilai keanggotaannya sendiri – sendiri. Sebagai contoh, 2 fungsi pada gambar 2.14 akan menghasilkan nilai defuzzifikasi sebagai berikut:

$$z^* = \frac{a(0.5) + b(0.9)}{0.5 + 0.9}$$



Gambar 2.14 Defuzzifikasi Weighted Average

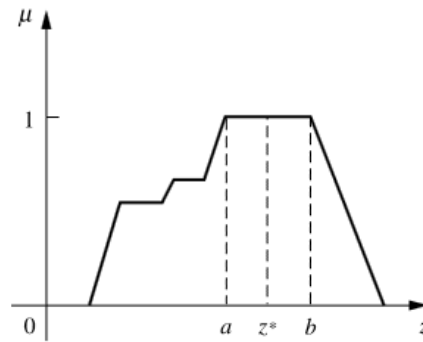
Sumber: Ross, 2004

4. Mean Max Membership

Dikenal juga sebagai *middle of maxima*, metode hampir serupa dengan Max Membership, kecuali lokasi dari metode Max Membership bisa juga non-unik. Diberikan persamaan:

$$z^* = \frac{a + b}{2}$$

dimana nilai a dan b dapat dijabarkan secara grafik pada gambar 2.15:



Gambar 2.15 Defuzzifikasi Mean Max Membership

Sumber: Ross, 2004

2.9 Diskritisasi

Banyak cara yang dapat digunakan untuk proses diskritisasi (mengubah bentuk analog menjadi diskrit), tiga diantaranya yang banyak digunakan dalam bidang kontrol adalah *backward difference*, *forward difference* (metode Euler) dan *bilinear transformation*. Semua metode yang digunakan hanya merupakan pendekatan (*approximations*), sehingga hasilnya tidak akan persis sama dengan bentuk analog. Hal ini dikarenakan dalam proses diskritisasi, selalu ada sebagian informasi yang hilang, yaitu informasi yang terjadi antara satu cuplikan dengan cuplikan lainnya, walaupun frekuensi cuplik yang digunakan tinggi. Hilangnya informasi juga terjadi karena formula matematika yang digunakan dalam proses diskritisasi juga diturunkan berdasarkan pendekatan agar lebih mudah digunakan.

Berikut tahapan diskritisasi yang dapat digunakan:

1. Tulis algoritma analog dalam bentuk transformasi laplace.
2. Lakukan diskritisasi menjadi bentuk transformasi Z dengan mengganti operator s dengan menggunakan salah satu dari tiga metode diskritisasi, yaitu:

a. *Backward difference* :

$$S = \frac{1-z^{-1}}{T_s} \tag{2-20}$$

b. *Forward difference* :

$$S = \frac{1-z^{-1}}{T_s z^{-1}} \tag{2-21}$$

c. *Bilinear transform* :

$$S = \frac{2(1-z^{-1})}{T_s(1+z^{-1})} \tag{2-22}$$

dengan T_s = periode cuplik

Hingga tahap 2, algoritma sudah didapat dalam bentuk diskrit yang dinyatakan dalam transformasi Z . Pada implementasinya, bentuk transformasi Z tersebut perlu diubah menjadi time domain (persamaan beda), yaitu dengan mengubah operator z^n menjadi n kali waktu delay (z^{-1} berarti 1 kali waktu delay, z^{-2} berarti 2 kali waktu delay dan seterusnya).

2.10 Kontroler *Self-Tuning* Parameter PI Metode Logika *Fuzzy*

Teknologi telah berkembang, dimana tidak lagi memakai cara konvensional untuk mendapatkan suatu hasil yang diinginkan dengan memakai persamaan matematika, tetapi bisa menerapkan suatu pengalaman manusia untuk mengontrol sesuatu, yaitu dengan aturan – aturan. *If – then Rules* adalah suatu kontrol yang akan mengikuti pendekatan secara bahasa, sistem ini disebut dengan sistem kontrol logika *fuzzy*. Dimana sistem kontrol logika *fuzzy* ini tidak memiliki ketergantungan pada variabel – variabel proses kontrol. (Lin & Lee, 1996). Sistem ini banyak dikembangkan dalam bidang teknik kontrol, terutama untuk sistem nonlinear dan dinamis. Pada aplikasi industri yang membutuhkan suatu sistem kontrol dengan kecepatan tinggi dan data *output* yang akurat maka pemakaian kontroler PI masih dianggap kurang memuaskan, karena jika menggunakan kontroler PI, dan kontroler ditala terlalu sensitif, maka *overshoot/undershoot* yang dihasilkan akan sangat peka, sehingga osilasi yang ditimbulkan akan lebih tinggi, sedangkan bila kontroler ditala kurang sensitif maka terjadinya *overshoot/undershoot* dapat diperkecil, tetapi waktu yang dibutuhkan akan semakin lama, dan ini akan menjadi suatu masalah dalam suatu proses industri (Kristiyanto, 2015).

Sistem kontrol *self-tuning* parameter PI metode logika *fuzzy* dalam penelitian ini merupakan kontrol logika *fuzzy* yang digunakan untuk menala parameter kontroler PI. Sistem kontrol tersebut kemudian dicoba untuk diimplementasikan pada sistem untuk mengontrol *plant* yang berupa kecepatan motor BLDC dengan pembebanan tertentu.

Sistem utama dalam kontrol *self-tuning* parameter PI adalah kontroler PI, sedangkan logika *fuzzy* disini berfungsi untuk memperbaiki performansi sistem dengan cara menala nilai parameter PI yaitu K_p dan K_i secara otomatis terhadap pembebanan yang berubah.