

# Pengaturan Kecepatan Motor Induksi Tiga Fasa dengan Metode PID Self Tuning Berdasarkan Fuzzy pada Rancangan Mobil Hybrid

Septian Ekavandy<sup>#1</sup>, Dedid Cahya Happyanto<sup>#2</sup>

<sup>#</sup>Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya  
Kampus PENS-ITS Sukolilo, Surabaya  
ivan@student.eepis-its.edu  
dedid@lecturer.eepis-its.edu

**Abstrak**— Pengontrol Fuzzy PID Self-Tuning adalah kombinasi dari pengontrol Proporsional – Integral – Derivative konvensional dengan pengetahuan berdasarkan Fuzzy Logic. Untuk mendapatkan nilai inisial dari parameter PID harus dilakukan autotuning, dalam hal ini menggunakan metode Ziegler-Nichols dengan metode frekuensi. Pengontrol PID konvensional mengatur proses closed-loop, sedangkan Fuzzy Logic mengatur parameter-parameter dan penguatan PID untuk keluaran sistem. Pengontrol Fuzzy PID Self-Tuning sangat cocok diterapkan pada mobil hybrid karena mempunyai respon yang baik pada perubahan set point dan perubahan beban. Dalam hal ini menggunakan beban berupa motor induksi tiga fasa dimana frekuensi penyulutannya dikendalikan dengan inverter yang dikontrol dengan PWM.

**Kata kunci**— Motor AC Induksi Tiga Fasa, Fuzzy Logic Controller, PID Self Tuning Controller, Ziegler-Nichols Frequency Response Method, Autotuning Method

## I. PENDAHULUAN

PID merupakan metode pengendalian yang dengan mudah diimplementasikan pada berbagai plant di kalangan industri, hal ini dikarenakan kemurahan dan kemudahannya dalam pengendalian serta implementasi dalam sistem, khususnya pada sistem dengan loop tertutup. Selain itu kontroler PID juga menghasilkan respon yang baik terhadap sebagian besar loop di industri. Seiring meningkatnya teknologi mikrokontroler yang semakin mengalami peningkatan kinerja secara komputasional, para pakar kontrol pun semakin berusaha meningkatkan kemampuan kinerja kontroler PID. Hal ini diusahakan dengan cara menempelkan sistem kontrol yang lain ke dalam kontroler PID. Kontroler logika fuzzy banyak dipilih untuk meningkatkan kinerja kontroler PID. Selain ketangguhan respon sistem terhadap gangguan beban, fuzzy juga mempermudah sistem komputasional pada PID dengan menerapkan kontrol berbasis linguistik yang lebih mudah dirancang sesuai dengan intuisi operator ahli.

## II. TEORI PENGONTROL FUZZY PID SELF TUNING

Pada bab ini akan dibahas sekilas tentang segala hal yang berkaitan dengan pengontrol fuzzy PID Self Tuning.

### A. Kontroler PID Self Tuning

Kemudahan dan ketangguhan PID berbasis fuzzy banyak digunakan diberbagai kalangan industri sebagai teknik pengendalian pada sebagian besar loop-nya. Seiring penelitian yang terus berkembang dalam ilmu kontrol modern, PID menawarkan berbagai macam kemudahan dalam perancangannya. Untuk menentukan ketiga parameter kontrol awalnya, tidak lagi diperlukan pengetahuan tentang model sistem atau pemodelannya itu sendiri. Hal ini dapat dilakukan dengan mengaplikasikan teknik autotuning. Ada berbagai macam teknik yang ditawarkan antara lain metode respon step Ziegler-Nichols, relay autotuner metode *good-tuning*, namun dalam hal ini dipilih metode *good tuning*. Karena menurut berbagai literatur, memang metode Z-N yang paling banyak digunakan, namun demikian metode ini kadang tidak cocok dengan plant yang akan dikontrol. Setelah dilakukan percobaan memang terbukti bahwa metode *good tuning* dalam hal ini lebih baik dari pada metode Z-N. Untuk mendapatkan ketiga parameter awal PID ( $K_p$ ,  $T_i$ , dan  $T_d$ ) dilakukan dengan cara mengatur sisi parameter integral dan derivatif menjadi nol, dan berangsur-angsur meningkatkan nilai  $K_p$  sehingga dicapai respon sistem yang baik dalam batas masih dapat ditoleransi. Pada saat kondisi kritis tersebut didapatkan nilai *Ultimate Gain* ( $K_u$ ) dan *Ultimate Period* ( $T_u$ ). Dimana  $K_u$  adalah nilai  $K_p$  saat itu dan nilai  $T_u$  adalah jarak antara puncak *overshoot* dan puncak *undershoot* yang pertama kali. Untuk mendapatkan nilai ketiga parameter PID, nilai  $K_u$  dan  $T_u$  dimasukkan ke dalam tabel metode *Good Gain Tuning* seperti pada Tabel 1 dibawah ini:

TABEL I  
METODE GOOD GAIN TUNING

Jenis kontroler	$K_p$	$T_i$	$T_d$
PID	$0.8K_p$	$1.5T_u$	$0.25T_i$

Setelah ketiga nilai parameter awal PID didapatkan, selanjutnya dimasukkan ke dalam persamaan umum PID sebagai berikut:

$$u(k) = Kc \left[ e(t) + \frac{1}{Ti} \int_0^t e(\tau) d\tau + Td \frac{de(t)}{dt} \right]$$

Karena proses yang akan diimplementasikan dalam domain diskrit, maka persamaan PID yang digunakan adalah sebagai berikut:

$$u(k) = Kc \left[ e(k) + \frac{Ts}{Ti} \sum_{i=1}^k e(k) + \frac{Td}{Ts} \Delta e(k) \right]$$

Dimana:

$u(k)$  = aksi kontrol,

$Kc$  = proportional gain,

$Ts$  = waktu sampling,

$Ti$  = integral time constant,

$e(k)$  = error antara set point dikurangi keluaran ke-k,

$e(i)$  = jumlah error ke-k,

$\Delta e(k)$  = beda error ke-(k-1) dengan error ke-(k).

Permasalahan yang selalu ada pada persamaan PID konvensional adalah jika diinginkan respon set point yang bagus akan menghasilkan respon yang kurang baik terhadap perubahan beban. Begitu pula sebaliknya, jika ingin menghasilkan respon yang baik terhadap perubahan beban akan menghasilkan respon set point yang terlalu overshoot. Idealnya, kontroler yang baik adalah jika respon sistem bagus terhadap set point maupun perubahan beban. Untuk mengatasi kelemahan yang ada, menurut pustaka [ ] dilakukan dengan cara menambahkan faktor pemberat set point (*weighting factor*)  $\beta$  pada sisi proporsional. Menurut pustaka [ ] jika  $\beta$  diatur dengan nilai kurang dari satu, overshoot akan bisa diredam dan tak akan mempengaruhi respon terhadap perubahan beban. Nilai  $\beta$  akan dimodulasi oleh kontroler sejalan dengan waktu dan perubahan kondisi. Maka persamaan PID di atas menjadi:

$$u(k) = Kc \left[ (\beta yr - y(k)) + \frac{Ts}{Ti} \sum_{i=1}^k e(i) + \frac{Td}{Ts} \Delta e(k) \right]$$

Dimana:

$yr$  = set point sistem,

$y(k)$  = keluaran sistem ke-(k) sesuai pembacaan sensor.

Merujuk pustaka [ ], selain  $\beta$ , parameter lain yang berpengaruh terhadap perubahan parameter PID adalah  $\alpha$ . Parameter ini berkontribusi pada perubahan domain integral dan derivatif, seperti persamaan berikut:

$$K_{pGG} = 0.8K_u, \quad T_i = \frac{3}{1 + \alpha} T_{ou}, \quad T_d = 0.25T_i$$

Apabila nilai  $\alpha$  pada persamaan di atas adalah 1, maka tabel metode Good Gain Tuning akan terwakili. Dari persamaan PID dan persamaan di atas dapat diambil kesimpulan bahwa saat nilai  $\alpha$  sama dengan 1 dan nilai  $\beta$  sama dengan 1, keluaran PID akan kembali ke nilai inisial. Nilai  $\alpha$  dan  $\beta$  akan berubah-ubah, sehingga parameter PID akan berubah sesuai dengan kondisi tertentu pada sampling saat itu. Perubahan nilai  $\alpha$  dan  $\beta$  dapat digambarkan dengan persamaan berikut ini:

$$\alpha(k) = 1 + h_\alpha(k) * sfa$$

$$\beta(k) = 1 + h_\beta(k) * sf\beta$$

dimana:

$h_\alpha(k)$  = keluaran fuzzy waktu k untuk perubahan  $\alpha$ ,

$h_\beta(k)$  = keluaran fuzzy waktu k untuk perubahan  $\beta$ ,

$sfa$  = faktor penyekala untuk  $\alpha$ ,

$sf\beta$  = faktor penyekala untuk  $\beta$ .

### B. Kontroler Logika Fuzzy

Logika fuzzy dipilih untuk mendapatkan variasi nilai parameter PID. Logika fuzzy adalah kontroler yang mewakili pemikiran manusia, karena berbasis linguistik. Jika operator tersebut sudah lama menangani *plant* tertentu, maka intuisinya akan menghasilkan kontrol yang bagus jika dituangkan pada aturan kontrol fuzzy. Berikut ini adalah langkah-langkah kontrol fuzzy:

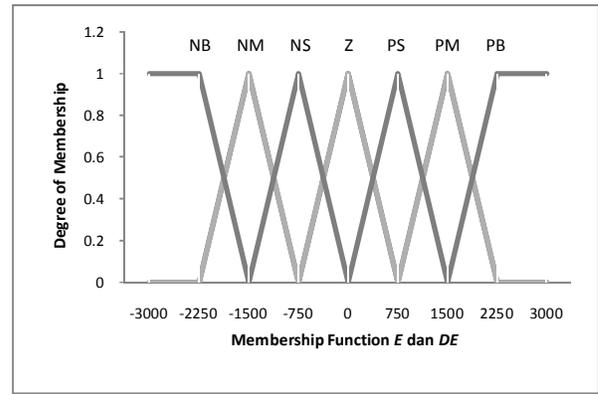
1) *Fuzzifikasi*: Fuzzifikasi adalah hal pertama yang dilakukan dengan mentransformasi *range* dari nilai variabel-variabel input kedalam semesta pembicaraan, kemudian nilai linguistik yang cocok. Dengan kata lain, tujuan dari *fuzzifier* adalah untuk mentransformasi ruang input  $crisp U \subset R^n$  kedalam sebuah *fuzzy set* yang didefinisikan dalam  $U$  dan dikarakteristikan oleh sebuah fungsi keanggotaan  $\mu_A : U \rightarrow [0,1]$  yang dikemas oleh istilah linguistik seperti 'kecil', 'besar', dan lain sebagainya. Ada sejumlah *fuzzifier* yang diketahui, tetapi yang paling umum adalah *fuzzifier* singleton, yang menginterpretasikan sebuah input  $y_0$  sebagai sebuah *fuzzy set A* dengan fungsi keanggotaan  $\mu_A(y)$  sama dengan nol kecuali  $y_0$  dengan  $\mu_A(y_0)$  yang bernilai satu.

2) *Aturan Dasar Kontrol Fuzzy*: Aturan dasar kontrol fuzzy mewakili informasi linguistik dan basis pengetahuan yang di dapat dari pengalaman teknisi. Berikut ini adalah sekumpulan aturan kontrol fuzzy yang diekspresikan sebagai pernyataan kondisional fuzzy {IF (*antecedent*) THEN (*consequent*)}. *Antecedent* adalah sebuah variabel fuzzy yang mewakili sebuah kondisi dalam domain aplikasinya dan *consequent* adalah variabel fuzzy yang lain yaitu adalah aksi kontrol. Secara umum, aturan-aturan umum kontrol fuzzy mempunyai bentuk sebagai berikut:

$$R^j : \text{if } x \text{ is } A_j, \text{ and } y \text{ is } B_j \text{ then } z \text{ is } C_j, \quad j = 1, 2, 3, \dots$$

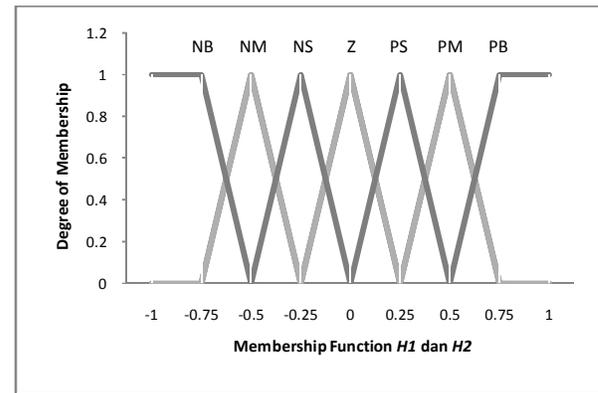
Dimana  $x$  dan  $y$  adalah variabel-variabel input untuk sistem fuzzy,  $z$  adalah variabel outputnya, dan  $A_j, B_j, C_j$  adalah variabel fuzzy dengan fungsi keanggotaannya secara berturut-turut  $\mu_{A_j}(x), \mu_{B_j}(y),$  dan  $\mu_{C_j}(z)$ . Masing-masing aturan kontrol fuzzy dapat dilihat sebagai penggambaran sebuah relasi antara variabel-variabel fuzzy. Ini berarti bahwa sebuah pernyataan kondisional fuzzy dapat diimplementasikan dengan sebuah implikasi fuzzy (*fuzzy relation*) sebagai  $A_j \times B_j \rightarrow C_j$  yang didefinisikan sebagai sebuah *fuzzy set* dalam *product space*  $U \times R_j$ . Banyak aturan-aturan implikasi fuzzy yang telah diajukan dalam proyek akhir ini, tetapi yang paling sering digunakan dan yang digunakan dalam proyek akhir ini adalah aturan operasi minimum yang didefinisikan dalam persamaan (2.15) berikut:

3) *Inferensi Fuzzy*: atau yang biasa disebut penyimpulan fuzzy adalah sebuah pemetaan dari *fuzzy sets* dalam  $U$  ke *fuzzy sets* dalam  $R$ , berdasarkan aturan-aturan kontrol fuzzy dan kesimpulan dari aturan komposisional. Suatu misal adalah sembarang *fuzzy set* dalam  $U$ , kemudian masing-masing menentukan sebuah *fuzzy set* dalam  $R$  berdasarkan pada kesimpulan dari aturan komposisional seperti persamaan (2.16) berikut ini:



Gambar 1 Fungsi keanggotaan kedua input fuzzy

Aturan-aturan kontrol dihubungkan dengan koneksi “else” yang merepresentasikan sebagai kondisi “or”. Operator yang paling umum digunakan untuk *fuzzy set* operasi fuzzy “or” adalah operasi *max*. Maka dari itu, keseluruhan hasil dari semua aturan kontrol fuzzy dihitung dengan mengkombinasikan hasil individual dari masing-masing aturan menggunakan operator *max*.



Gambar 2 Fungsi keanggotaan kedua keluaran fuzzy

4) *Defuzzifikasi*: Defuzzifikasi memuat pengonversian dari kesimpulan aksi kontrol fuzzy kedalam sebuah nilai *crisp* kemudian menyekalakan aksi kontrol *crisp* tersebut ke dalam *range* yang cocok untuk elemen kontrol akhir. Metode yang paling banyak digunakan adalah *Center of Area (COA)*. Pada metode ini, pusat luasan atas fungsi keanggotaan yang dihasilkan dari proses penyimpulan yang dihitung. Apabila semesta pembicaraan adalah diskrit, maka COA dapat ditunjukkan seperti pada persamaan (2.17) berikut ini:

Setelah fungsi keanggotaan input dan output didapatkan, langkah selanjutnya adalah menentukan aturan-aturan dasar kontrol fuzzy. Langkah ini dibuat sebagai interkoneksi antara input dan output dalam berbagai kondisi. Tabel II dan III adalah aturan dasar yang menentukan output  $\alpha$  dan  $\beta$  secara berturut-turut.

TABEL II  
ATURAN DASAR FUZZY UNTUK MENENTUKAN NILAI H1

		DE							
		H1	NB	NM	NS	Z	PS	PM	PB
E	NB	NB	NB	NM	PB	PM	PS	Z	
	NM	NB	NM	NS	PM	PS	Z	NS	
	NS	NB	NM	NS	PS	Z	NS	NM	
	Z	NB	NM	NS	Z	NS	NM	NB	
	PS	NM	NS	Z	PS	NS	NM	NB	
	PM	NS	Z	PS	PM	NS	NM	NB	
	PB	Z	PS	PM	PB	NM	NB	NB	

TABEL III  
ATURAN DASAR FUZZY UNTUK MENENTUKAN NILAI H2

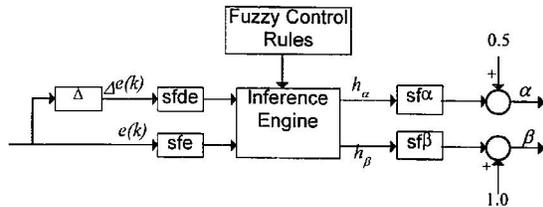
DE

Dimana  $n$  adalah jumlah level kuantisasi dari output, dan  $\alpha$  adalah derajat keanggotaan, sedangkan  $\beta$  sendiri adalah data saat itu.

$e(k)$  dan  $\Delta e(k)$  akan menjadi input kontrol fuzzy. Sedangkan output fuzzy adalah  $\alpha$  dan  $\beta$  untuk sisi  $\alpha$ , dan  $\beta$  untuk sisi  $\beta$ . Himpunan fuzzy pada kedua input dan output tersebut dibuat tujuh keanggotaan mulai dari *Positive Big (PB)* hingga *Negative Big (NB)*. Hal ini ditujukan agar keluaran yang dihasilkan lebih presisi. Pembagian himpunan keanggotaan fuzzy berdasarkan pada segala kemungkinan yang mungkin akan terjadi dalam sistem, baik pada segi input maupun output. Gambar 1 dan Gambar 2 berikut ini adalah fungsi keanggotaan input dan output fuzzy.

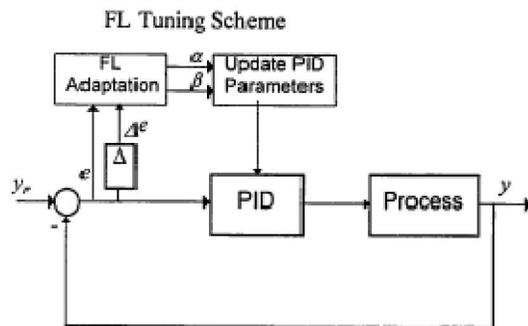
<b>H1</b>	<b>NB</b>	<b>NM</b>	<b>NS</b>	<b>Z</b>	<b>PS</b>	<b>PM</b>	<b>PB</b>
<b>NB</b>	NB	NB	NM	PB	PM	PS	Z
<b>NM</b>	NB	NM	NS	PM	PS	Z	NS
<b>NS</b>	NB	NM	NS	PS	Z	NS	NM
<b>Z</b>	NB	NM	NS	Z	NS	NM	NB
<b>PS</b>	NM	NS	Z	PS	NS	NM	NB
<b>PM</b>	NS	Z	PS	PM	NS	NM	NB
<b>PB</b>	Z	PS	PM	PB	NM	NB	NB

Blok diagram pengontrol logika fuzzy dapat digambarkan seperti Gambar 3 berikut ini.



Gambar 3 Blok diagram pengontrol logika fuzzy

Sedangkan blok diagram pengontrol fuzzy PID self tuning dapat digambarkan pada Gambar 4 berikut ini.



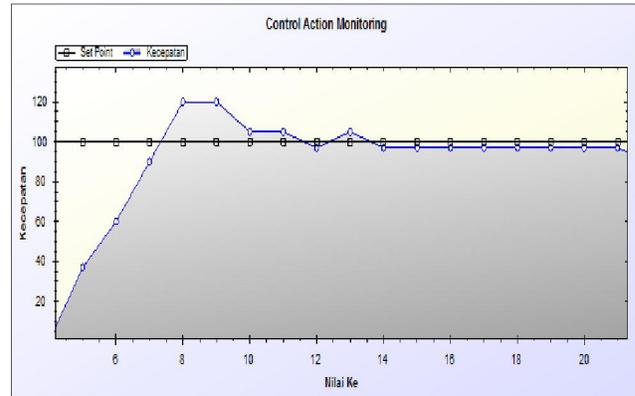
Gambar 4 Blok diagram pengontrol Fuzzy PID self Tuning

### III. PENGUJIAN, PENGAMBILAN DATA, DAN ANALISA

Proses pengerjaan yang dilakukan berturut-turut adalah menentukan set point, kemudian menentukan parameter awal PID untuk set point tersebut, kemudian menjalankan plant. Dari proses tersebut terbaca kecepatan motor pada sensor yang menjadi umpan balik pengontrol, kemudian terjadi perhitungan dan terjadi aksi kontrol. Sebelum itu semua, yang dilakukan terlebih dahulu adalah melihat data kecepatan motor pada frekuensi-frekuensi tertentu. Hal ini dilakukan dengan cara melakukan penambahan mulai dari 1 hingga 255 register OCR2 pada mikrokontroler Atmega32 merujuk pustaka [ ]. Ini dikarenakan OCR2 adalah register 8 bit yang mempunyai nilai maksimal 255, jika nilainya 0, maka tegangan yang keluar untuk men-drive inverter adalah 0V, jika nilainya 255, maka tegangan yang keluar untuk men-drive inverter adalah 5V, ini berarti motor dijalankan pada frekuensi maksimum. Inverter tiga fasa yang digunakan adalah OMRON SYSDRIVE 3G3MV, inverter ini dapat dikendalikan secara *local* maupun *remote*, tergantung pengaturannya. Jika inverter ingin dikontrol dari luar, bisa dilakukan dengan mengatur modeny.

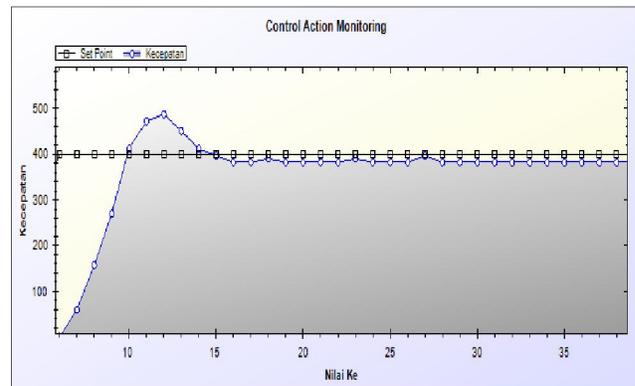
Pengaturan secara *remote* dapat dilakukan dengan memberikan arus sebesar 0mA – 20mA, atau 4mA – 20mA. Bisa juga di kontrol secara *remote* menggunakan tegangan antara 0V - 5V. Untuk lebih mudahnya, dilakukan pengaturan *remote* dengan input tegangan, ini dikarenakan keluaran maksimal dari mikrokontroler adalah 5V. Jika output ini dimodulasi melalui PWM, maka output yang dihasilkan akan bervariasi mulai dari 0V sampai 5V.

Gambar 5 sampai dengan Gambar 8 berikut ini adalah data-data yang didapatkan pada pengujian alat. Gambar 5 diambil saat set point diatur pada kecepatan 360. Respon kontrolnya adalah sebagai berikut.



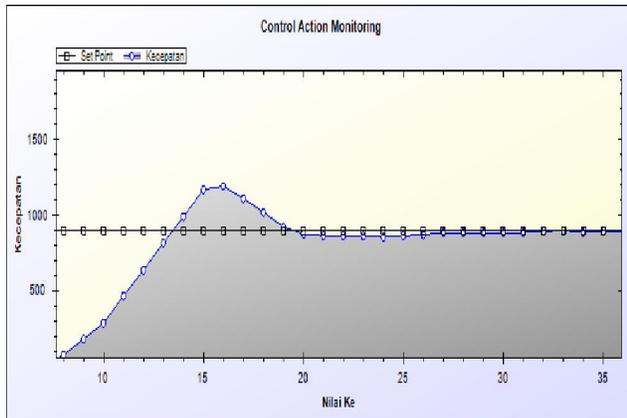
Gambar 5 Respon kecepatan saat set point di atur 100

Gambar 6 diambil saat set point diatur pada kecepatan 600. Respon kontrolnya adalah sebagai berikut.



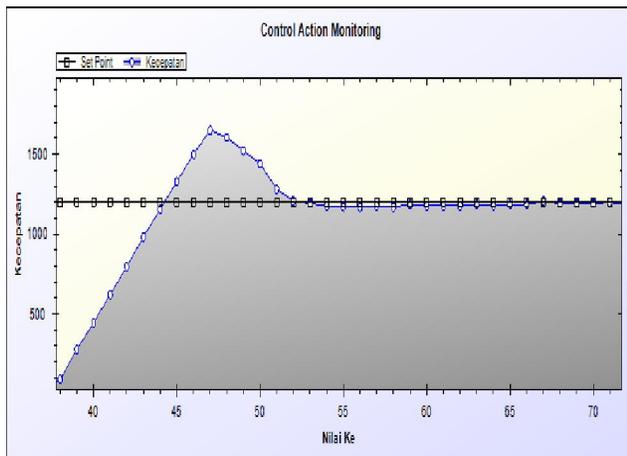
Gambar 6 Respon kecepatan saat set point di atur 400

Gambar 7 diambil saat set point diatur pada kecepatan 900. Respon kontrolnya adalah sebagai berikut.



Gambar 7 Respon kecepatan saat set point diatur 900

Gambar 8 diambil saat set point diatur pada kecepatan 900. Respon kontrolnya adalah sebagai berikut.



Gambar 8 Respon kecepatan saat set point diatur 1200

Dari data-data yang didapatkan, terlihat dari keempat percobaan penggantian set point, respon yang keluar hasilnya

cukup memuaskan. Hal ini dikarenakan pemilihan *scaling factor* yang tepat, sehingga perubahan error terjadi sangat proporsional, yang berdampak pada persen overshoot yang relatif rendah. Hal ini membuktikan bahwa kontroler PID tangguh terhadap perubahan set point, baik pada pengoperasian frekuensi rendah maupun pengoperasian pada frekuensi tinggi. Perlu diketahui saat pemberian frekuensi maksimal, motor kurang lebih berputar dengan kecepatan 1770 rpm.

#### IV KESIMPULAN

Setelah dilakukan proses perencanaan, pembuatan dan pengujian alat serta dengan membandingkan dengan teori-teori penunjang, dan dari data yang didapat maka dapat disimpulkan beberapa hal.

1. Metode pengontrolan fuzzy PID self tuning sangat bagus digunakan pada berbagai macam perubahan set point, hal ini dikarenakan kemampuan adaptasinya sangat bagus.
2. Pada pengoperasian frekuensi tinggi, persen overshoot yang dihasilkan semakin baik.
3. Steady state error yang dihasilkan relatif kecil pada berbagai pengujian set point.

#### DAFTAR PUSTAKA

- [1] W Eddy ; "PID for Line Follower", Chicago Area Robotics Group ; 2007
- [2] Maeda, Y. Kuswadi, Son. M, Nuh, Sulistyio MB. *Kontrol Automatik*. Politeknik Elektronika Surabaya; 1993.
- [3] Datasheet Atmega 32
- [4] Astrom, K. J., T. Hagglund, C. Hang, and W. Ho,"Automatic Tuning and adaptation for PID controllers – A Survey", IFAC Adaptive systems in control and signal processing, Grenoble, France, 19992
- [5] Donoghue, J. F., "Review of control design approaches for transport delay process", ISA Transaction, vol.16, no.2, 1977
- [6] Astrom, K. J., "Intelligent tuning", IFAC Adaptive systems in control and signal processing, Grenoble, France, 1992
- [7] Astrom, K. J., and B. Wittermark, "Adaptive Control", Addison-WessleyPublication Co.,1989.