

SISTEM PENGATURAN MOTOR AC SERVO DENGAN KONTROLER PID *SELF TUNING* BERBASIS *FUZZY LOGIC*

Tatang Sukendar⁽¹⁾, Berayan Munthe⁽²⁾

⁽¹⁾⁽²⁾ Widyaiswara P4TK BMTI Bandung,⁽¹⁾⁽²⁾

ABSTRACT

PID controller is combination of propotional, integral and derivative controller. Propotional, integral and derivative controller are used to fast the reaction of system, disappear offset and get adding energy when starting the change of load. Performance of Conventional PID controller very high once in a while need set back parameter on changes of the load dynamically. Self Tuning PID controller can set PID parameter and adapt to changes of the load dynamically so that disadvantages on convensional PID controller can be reduce. In this research start on identify plant and finished on experiment PID controller so that can improve exactly performance of PID controller combine with self tuning fuzzy logic The result of research that have done to control plant, self tuning PID controller can adapter with changes of the load then conventional PID controller.

Key word : *Convensional PID controller, self tuning PID controller, fuzzy logic, plant*

PENDAHULUAN

Kontrol otomatis telah memegang peranan yang penting dalam perkembangan ilmu pengetahuan dan teknologi terutama dibidang industri. Kontrol otomatis merupakan bagian terpenting dalam pabrik dan industri modern. Sebagai contoh kontrol yang digunakan untuk mengatur tekanan, temperatur, kecepatan, viskositas dan aliran. Dengan kontrol yang tepat dapat meningkatkan kualitas dan menurunkan biaya produksi serta mempertinggi laju produksi itu sendiri.

Kontroler PID (*Proportional – Integral – Derivative*) merupakan kombinasi dari tiga

jenis kontroler. Jika masing-masing dari ketiga jenis kontroler tersebut berdiri sendiri maka hasil yang dicapai akan kurang baik sebab masing-masing memiliki kelemahan maupun kelebihan sendiri-sendiri.

Apabila dikombinasikan ketiga jenis kontroler tersebut menjadi sistem kontrol tunggal, diharapkan memberikan kontribusi dari kelebihan masing-masing. Kontrol *proportional* adalah suatu penguat linier yang dapat diatur penguatannya. Hubungan antara keluaran $u(t)$ dan sinyal kesalahan $e(t)$ adalah :

$$u(t) = K_P e(t)$$

Dimana : K_P = *gain proportional*
 $u(t)$ = sinyal aksi kontrol
 $e(t)$ = Sinyal error

Kontrol *Proportional-Integral* (PI) merupakan perubahan keluaran kontrol gabungan *proportional* dan *integral* $u(t)$ berubah dengan fungsi yang sebanding

$$u(t) = K_P e(t) + \frac{K_P}{T_i} \int_0^t e(t) dt$$

Dimana : T_i = waktu *integral*

Terapan waktu *integral* T_i mengatur aksi kontrol *integral*, sedangkan K_P memperkuat bagian *proportional* maupun bagian *integral* dari aksi kontrol. Terapan T_i disebut laju reset . Laju reset adalah banyaknya

dengan sinyal error. Hubungan antara keluaran kontroler $u(t)$ dan sinyal eror $e(t)$ adalah :

pengulangan bagian *proportional* dari aksi pengontrolan per detik.

Kontrol *Proportional-Derivative* (PD) didefinisikan sebagai berikut :

$$u(t) = K_P e(t) + K_P T_d \frac{de(t)}{dt}$$

Dimana : T_d = Terapan waktu *derivative*

Kontrol *derivative* sering disebut kontrol laju (*rate control*), karena besar keluaran *controller* sebanding dengan laju perubahan sinyal kesalahan. Terapan waktu turunan T_d adalah selang waktu bertambah majunya respon kontrol *proportional* yang disebabkan oleh aksi aksi laju (*rate action*).

Kontroller *Proportional-Integral-Derivative* (PID), merupakan kombinasi dari kontrol

proportional, kontrol *integral* dan kontrol *derivative*. Gabungan kontrol ini mempunyai keunggulan dalam memperbaiki kesalahan sinyal dibandingkan dengan masing-masing dari tiga jenis kontroller tersebut. Persamaan kontroller PID diberikan dalam persamaan sebagai berikut

$$u(t) = K_P e(t) + K_P T_d \frac{de(t)}{dt} + \frac{K_P}{T_i} \int_0^t e(t) dt$$

Untuk memenuhi system yang diinginkan maka ketiga parameter PID harus ditetapkan secara optimal. Ada beberapa cara untuk menentukan parameter tersebut, diantaranya adalah metode coba-coba (*cut and try method*), metode *Ziegler-Nichols*, metode tanggapan (*step respons*), dan metode analitik

PID kontroler konvensional hanya dapat mengolah eror secara apa adanya, artinya eror diolah secara *deterministic*, misalnya dikalikan secara proporsional, di integralkan, di differensialkan. Disimpan beberapa saat untuk selanjutnya digunakan untuk sinyal kontrol berikutnya. Kontrol PID konvensional hanya tepat pada *plant* yang linier dan tidak terjadi perubahan beban, sedangkan untuk beban

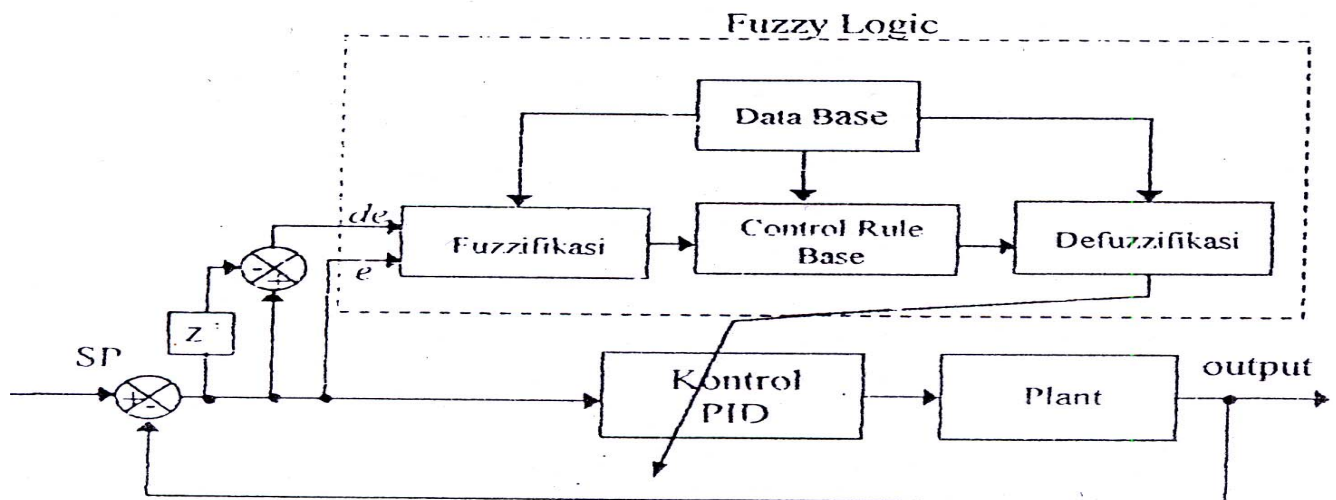
berubah perlu merubah parameter K_p , K_i dan K_d .

Kontrol *PID-Fuzzy Self Tuning* bekerja dengan menerima sinyal dari sensor sebagai output dari plant yang telah terkontrol. Sinyal tersebut kemudian dibandingkan dengan *set point* sehingga menghasilkan error yang baru.

Selanjutnya mencari kwantisasi dari error dan delta berdasarkan tabel kwantisasi yang telah dibuat sebelumnya. Nilai kwantisasi kemudian dijadikan dalam nilai-nilai *Fuzzy* melalui proses *fuzzifikasi* yang akan menghasilkan derajat

keanggotaan dari error dan delta error. Adapun diagram blok kontrol PID – *Fuzzy Self Tuning*

dapat dilihat pada gambar berikut ini.



Gambar 1. Kontrol PID – *Fuzzy Self Tuning*

Proses selanjutnya membandingkan antara derajat keanggotaan dari sinyal estimasi a dan b . Derajat keanggotaan yang terkecil dipilih sebagai derajat keanggotaan untuk sinyal kontrol dalam hal ini adalah nilai K_p (T_i dan T_d constant) dengan menggunakan rumus yang telah ada harga K_p yang masih dalam bentuk *fuzzy* dirubah menjadi harga sebenarnya melalui proses *Defuzzifikasi*. Untuk mendapatkan sinyal kontroler maka harga K_p , K_i dan K_d dimasukkan dalam rumus PID.

METODE PENELITIAN

Dalam penelitian ini menggunakan peralatan *AC Pre-Amplifier* Unit PA 150V, *Servo Amplifier* Unit SA 150 D, *Power Suplpy* Unit PS 150E, Unit Motor Tacho Generator dari AC Servo Unit 150 U, Unit PCL 712, PC, *Driver AC* yang dikontrol dengan tegangan DC, Sinyal conditioning sebagai *feedback*.

Tujuan penelitian adalah menerapkan kontrol *fuzzy* untuk mentuning parameter PID kontrol secara otomatis agar dapat bekerja secara optimal pada beban yang berubah ubah.

Metode penelitian yang digunakan terdiri dari identifikasi plant dan pengujian kontrol PID pada plant tersebut.

Selain itu menggunakan studi kepustakaan untuk mendapatkan metode dan algoritma yang akan digunakan pada program komputer baik dalam kontrol PID maupun logika *fuzzy*.

Studi kepustakaan juga dipakai untuk mendapatkan model matematika dari motor AC servo sehingga dapat ditentukan secara pasti variabel yang dikontrol.

HASIL DAN PEMBAHASAN

A. Pengujian Dan Analisa Data Respon Plant Dengan Kontroler PID

Pada pengujian kontroler PID ini, setelah didapatkan parameter K_p , K_i dan K_d dengan metode *Ziegler-Nichols* dilakukan percobaan dengan *try and error*, maka didapat nilai parameter kontroler PID dengan respon *critical damp*. Kemudian dengan mengatur K_p pada beban nol sampai beban maksimum (K_i dan K_d tetap)

dengan empat sekala, didapat parameter $K_p = 1$ untuk beban nol, $K_p = 1.68$ (beban maksimum), $K_i = 2.5$ dan $K_d = 0.1$ (semua beban) dengan parameter tersebut digunakan acuan pada perancangan PID maupun *Logika Fuzzy*.

1. Beban nol

Pada pengujian beban nol dengan parameter kontrol PID, $K_p = 1$, $T_i = 2,5$ dan $T_d = 0,1$. Setelah respon steady state beban dirubah menjadi beban maksimum.

Berdasarkan data dari hasil pengujian dengan time sampling 0,053 s diperoleh :

$$\begin{aligned} \text{Time constant (t)} &= (1,327 - 1,209) \times \\ &\frac{17 - 16}{17 - 16} \times 0,053 + (16 \times 0,053) \\ &= 1,353 - 1,209 \\ &= 0,89\text{s,} \end{aligned}$$

Rise Time (T_R) $(0,053 \times 26) - 0,496 = 0,7975\text{s}$, *Setling time* (T_{s0}) = 1,378 s; *Time Delay* (T_D) = 0,802 s; waktu yang dibutuhkan untuk steady state pada perubahan beban = 3,34s

2. Beban 1 (25 % dari beban maksimum)

Pada pengujian beban ke 1 ini dengan parameter kontrol PID, $K_p = 1$, $T_i = 2,5$, dan $T_d = 0,1$. Kemudian setelah respon dalam steady state beban diubah menjadi maksimum

Dari data hasil pengujian didapatkan nilai kualitas respon *transient* :

$$\begin{aligned} \text{Time constant (t)} &= (1,327 - 1,22) \times \\ &\frac{17 - 16}{17 - 16} \times 0,053 + (16 \times 0,053) = \\ &1,405 - 1,221 \\ &= 0,89\text{s,} \end{aligned}$$

Rise Time (T_R) $(1,1603 - 0,4491 = 0,7112\text{s}$, *Time Sampling* (T_s) = 1,6764

s; *Time Delay* (T_D) = 0,7497 s; waktu yang dibutuhkan untuk kembali *steady state* saat *Time Delay* (T_D) = 0,7497 s; waktu yang dibutuhkan untuk kembali *steady state* saat beban dirubah = 4.717s.

3. Pengujian Beban ke 2 (50% dari beban maksimum)

Pada pengujian beban ke 2 dengan parameter kontrol PID, $K_p = 1$, $T_i = 2,5$, $T_d = 0.1$. Kemudian setelah respon *steady state* beban dirubah menjadi beban maksimum.

Dari data hasil pengujian didapatkan nilai kualitas respons *transient* *Time Constant* (i) = 1,0735 s, *Rise Time* (T_R) = 0,9482 s, *Setling Time* (T_s) = 2,0773 s; *Time Delay* (T_D) = 0,975s; Waktu yang dibutuhkan untuk *steady state* pada saat beban dirubah maksimum = 2,594 s.

4. Beban ke 3 (75 % dari beban maksimum)

Pada pengujian beban ke 3 dengan parameter kontrol PID, $K_p = 1$, $T_i = 2,5$ dan $T_d = 0,1$, tidak dilakukan perubahan beban.

Dari data hasil pengujian didapatkan nilai kualitas respons *transient* :

$$\begin{aligned} \text{Time Costant (t)} &= 1,0887 \text{ s,} \\ \text{Rise Time (} T_R \text{)} &= 2,339 \text{ s,} \\ \text{Setling time (} T_s \text{)} &= 4,982 \text{ s;} \\ \text{Time Delay (} T_D \text{)} &= 0,989 \text{ s.} \end{aligned}$$

5. Beban Maksimum

Pada pengujian beban ini dengan parameter kontrol PID, $K_p=1$, $T_i=2,5$, $T_d=0.1$, tidak dilakukan perubahan beban.

Dari data hasil pengujian didapatkan nilai khusus respons *transient* :

Time constant (t) = 1,174 s, *Rise Time* (T_R) = 3,284 s, *Setling time* (T_S) = 5,893 s; *Time Delay* (T_D) = 1,0341 s.

6. Beban berubah

Pada pengujian ini beban dari nol dirubah ke beban maksimum kemudian beban dilepas ke beban nol lagi.

Dari data hasil pengujian didapatkan kualitas respons *transient*, dari beban nol kemudian diberi beban maksimum untuk *steady state* membutuhkan waktu = 4,081 s. Sedang jika beban dilepas untuk kembali pada keadaan *steady state* membutuhkan waktu = 16,96 detik.

Tabel 1 Data perbandingan analisa data percobaan

| No. | Kualitas Respons <i>Transient</i> | DATA RESPON PLANT DENGAN KONTROL PID | | | | | KETERANGAN |
|-----|-----------------------------------|--------------------------------------|------------|------------|------------|-------------|----------------|
| | | BEBAN NOL | BEBAN KE 1 | BEBAN KE 2 | BEBAN KE 3 | BEBAN MAKS. | |
| 1. | t | 0,8900 | 0,8910 | 1,0735 | 1,0887 | 1,1740 | Dalam detik |
| 2 | T_S | 1,3780 | 1,6764 | 2,0773 | 4,4982 | 5,8930 | |
| 3 | T_R | 0,7975 | 0,7112 | 0,9482 | 2,3390 | 3,2840 | |
| 4 | T_D | 0,8020 | 0,7497 | 0,9750 | 0,9890 | 1,0431 | |
| 5 | <i>Lag. Time</i> | 0,4089 | 0,477 | 0,5495 | 0,583 | 0,583 | |
| 6 | T_O | 3,3400 | 4,7170 | 2,5940 | - | - | Δ beban |
| 7 | T_b | 16,96 | - | - | - | - | Δ beban |

Keterangan : T_o = Perubahan beban dari beban nol ke beban maksimum
 T_b = Beban dilepas (dari beban maksimum)

Dari tabel 1, nampak dengan jelas bahwa semakin besar beban *plant*, *RiseTime*, *Sampling Time*, *Delay Time* maupun *lagging Time* semakin besar.

B. Pengujian dan Analisa Data Respon Plant Dengan Kontrol PID-Self Tuning

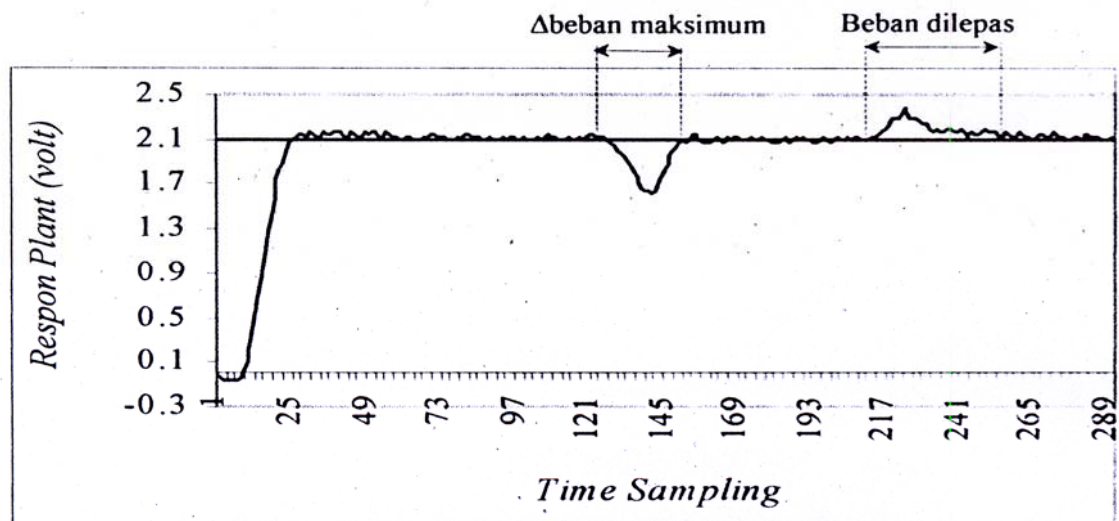
1. Beban Nol

Pada pengujian kontrol PID-*self tuning* dimulai saat *start*, dengan kondisi awal $K_p=1$, setelah diadakan estimasi K_p berubah menjadi 1,008. Pada saat diberi beban maksimum $K_p=1,312$. Dan

pada saat beban dilepas $K_p=1$. Dari data pengujian didapatkan kualitas respons *transient* sebagai berikut : *Time constant* (t) = 0,921 s, *Rise Time* (T_R) = 0,739 s, *Setling time* (T_S) = 1,27 s, *Time Delay* (T_D) = 0,841 s, waktu menuju *steady state* pada perubahan beban

maksimum = 0,475 s, waktu *steady state* ketika beban dilepas = 0,874s,

lagging time = 0,584s.

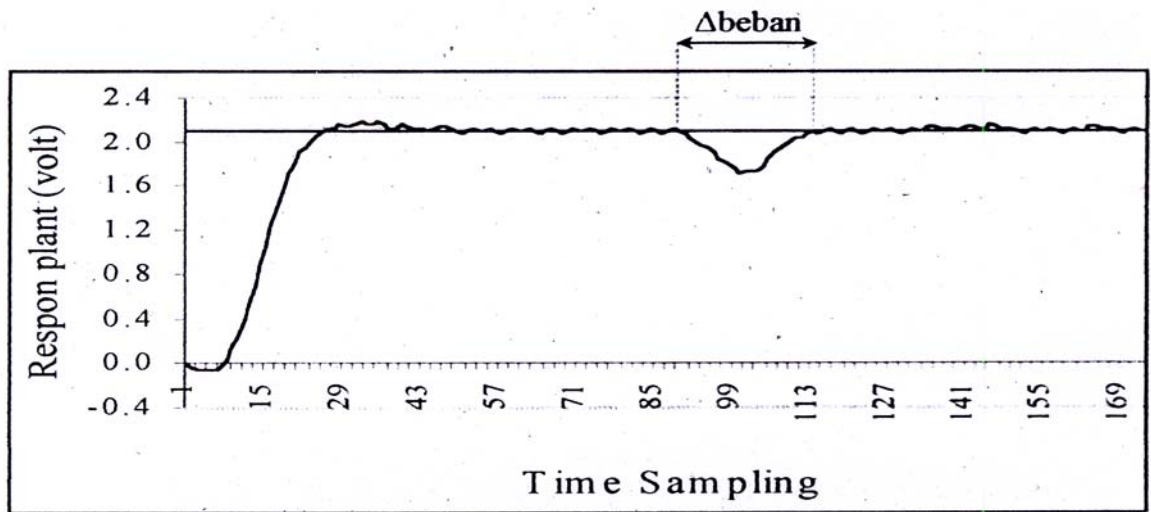


Gambar 2. Grafik respon plant beban nol

2. Beban ke 1 (25 % dari beban maksimum)

Pada pengujian beban ke 1 ini, kontrol PID-*self tuning* pada saat start dengan parameter $K_p = 1$, $T_i = 2,5$ dan $T_d = 0,1$ (T_i dan T_d tetap), setelah dilakukan estimasi K_p berubah menjadi 1,025. Kemudian saat beban dirubah menjadi beban maksimum parameter $K_p = 1,296$. Dari data pengujian didapatkan

nilai kualitas respon *transient* berikut : Time constant (t) = 0,4788 s, *Rise Time* (T_R) = 0,7512 s, *Setling time* (T_S) = 1,281 s, *Time Delay* (T_D) = 0,79 s, *Lagging Time* = 0,4788 s, sedang waktu yang dibutuhkan untuk kembali *steady state* saat beban dirubah menjadi maksimum adalah 0,689 s.



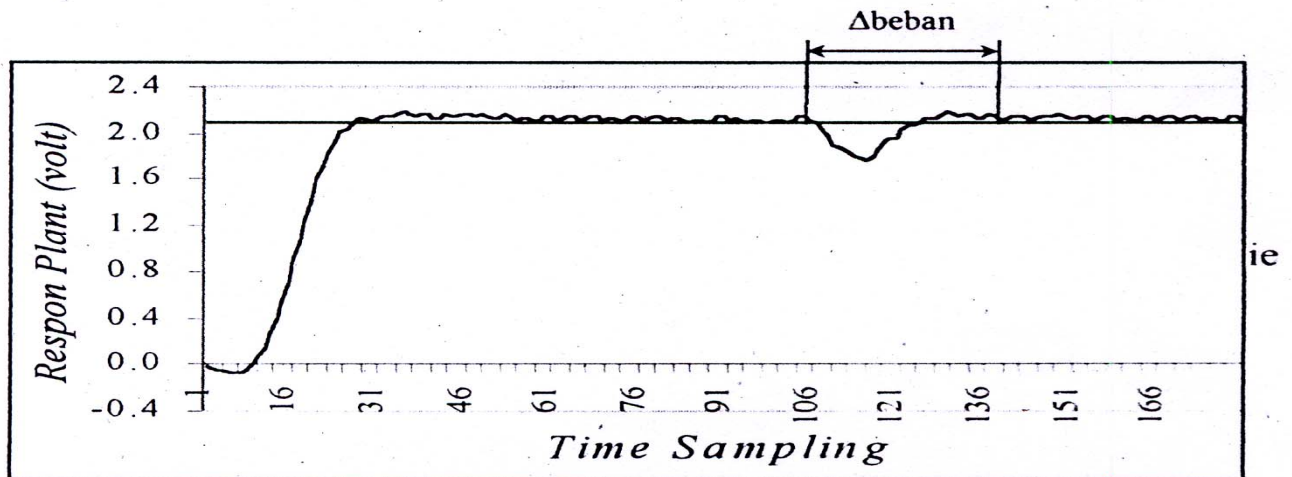
Gambar 3. Grafik respon plant beban ke 1

3. **Beban ke 2 (50 % dari beban maksimum)**

Pada beban kedua saat *start* $K_p = 1$, kemudian terjadi estimasi sehingga nilai $K_p = 1,186$. Pada saat dirubah menjadi beban maksimum K_p menjadi $= 1,64$.

Dari data hasil pengujian didapatkan nilai kualitas respons *transient*, *Time*

constant (t) = 0,979 s, *Rise Time* (T_R) = 1,166 s, *Setling time* (T_S) = 1,366 s, *Time Delay* (T_D) = 0,8956 s, *Lagging Time* = 0,6114 s. Sedangkan waktu yang dibutuhkan untuk kembali *steady state* saat **beban dirubah menjadi maksimum** = 0,689 s.

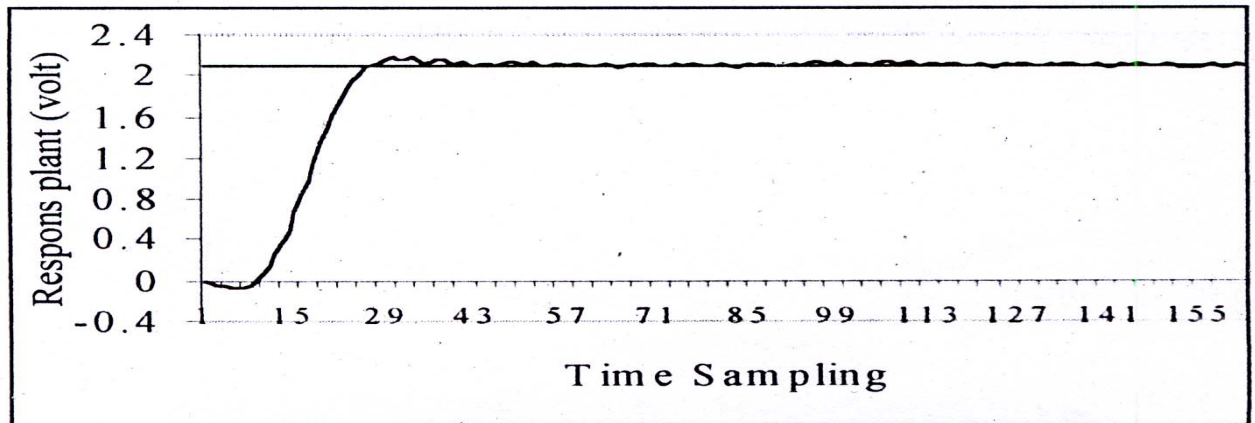


Gambar 4. Grafik respon plant beban ke 2

4. Beban ke 3 (75 % dari beban maksimum)

Dari data hasil pengujian didapatkan nilai kualitas respon *transient*. *Time*

constant (t) = 1,002 s, *Rise Time* (T_R) = 0,756 s, *Setling time* (T_S) = 1,359 s, *Time Delay* (T_D) = 0,922 s, *Lagging Time* = 0,6114 s.



Gambar 5. Grafik respon plant beban ke 3

5. Beban Maksimum

Pada pngujian beban maksimum, saat *start* awal $K_p = 1$, setelah dilakukan estimasi pada *time sampling* ke 16 parameter $K_p = 1,26$.

Dari data hasil pengujian didapatkan nilai kualitas respons *transient* : *Time*

constant (t) = 1,002 s, *Rise Time* (T_R) = 0,700 s, *Setling time* (T_S) = 1,333 s, *Time Delay* (T_D) = 0,936 s, *Lagging Time* = 0,659 s

Tabel 2. Data Pengujian kontrol PID-Self Tuning

| No. | Kualitas Respons <i>Transient</i> | DATA RESPON PLANT DENGAN KONTROL PID-SELF TUNING | | | | | |
|-----|-----------------------------------|--|------------|------------|------------|-------------|----------------|
| | | BEBAN NOL | BEBAN KE 1 | BEBAN KE 2 | BEBAN KE 3 | BEBAN MAKS. | KETERANGAN |
| 1. | t | 0,9210 | 0,4788 | 0,9790 | 1,0020 | 1,0020 | Dalam detik |
| 2. | T_S | 1,2700 | 1,2810 | 1,3660 | 1,3590 | 1,3330 | |
| 3. | T_R | 0,7390 | 0,7512 | 0,7100 | 0,7560 | 0,7000 | |
| 4. | T_D | 0,8410 | 0,7900 | 0,8956 | 0,9220 | 0,9360 | |
| 5. | <i>Lag. Time</i> | 0,5840 | 0,4788 | 0,6114 | 0,6114 | 0,6590 | |
| 6. | T_O | 0,4750 | 0,6890 | 0,3580 | - | - | Δ beban |
| 7. | T_b | 0,8740 | - | - | - | - | Δ beban |

C. Perbandingan Hasil Kontrol PID Konvensional Dan PID-Self Tuning

Dari pengujian kedua sistem kontrol dapat dibuat data perbandingan seperti pada tabel 3

Tabel 3 Perbandingan Kontrol PID Konvensional dan PID-Self Tuning

| No. | Kualitas Respons Transient | DATA PID Konvensional - Data PID-Self Tuning | | | | | KETERANGAN |
|-----|----------------------------|--|------------|------------|------------|-------------|-------------|
| | | BEBAN NOL | BEBAN KE 1 | BEBAN KE 2 | BEBAN KE 3 | BEBAN MAKS. | |
| 1. | T | -0,031 | 0,412 | 0,095 | 0,087 | 0,172 | Dalam detik |
| 2 | T _S | 0,108 | 0,395 | 0,711 | 3,139 | 4,560 | |
| 3 | T _R | 0,059 | -0,040 | 0,238 | 1,583 | 2,584 | |
| 4 | T _D | -0,039 | -0,040 | 0,079 | 0,067 | 0,107 | |
| 5 | Lag. Time | -0,175 | -0,002 | 0,364 | -0,028 | -0,076 | |
| 6 | T _O | 2,865 | 0,6890 | 0,3580 | - | - | Δ beban |
| 7 | T _b | 16,09 | - | - | - | - | Δ beban |

Dari table 3 dapat dilihat bahwa kontrol PID-Self Tuning mempunyai kelebihan sebagai berikut :

1. Kecepatan respons lebih cepat.
2. Waktu mencapai masuk dari respons *steady state* lebih cepat, sangat terlihat pada beban maksimum mencapai selisih 4,56 detik.
3. Waktu yang dibutuhkan untuk masuk pada *steady state* saat diberi perubahan beban lebih cepat.
4. Waktu keterlambatan respons *output* terhadap *input* lebih cepat, hanya pada beban nol agak terlambat dibanding kontrol PID konvensional.

KESIMPULAN

Dari analisa dan hasil pengujian pengaturan kecepatan motor AC-Servo dengan kontroler PID-Self Tuning berbasis Fuzzy dapat diambil beberapa kesimpulan sebagai berikut :

1. Kontrol PID-Self Tuning lebih dapat beradaptasi dengan perubahan beban
2. Kecepatan respon *plant* pada kontrol PID-Self Tuning lebih cepat

dibandingkan kontrol PID konvensional.

3. Waktu yang dibutuhkan respon *plant* untuk masuk *steady state* relative lebih cepat dibandingkan dengan kontrol PID Konvensional.
4. Kontrol PID-Self Tuning lebih cocok diterapkan pada *plant* yang mempunyai respons lambat, terutama pada *plant* yang berbeban besar.

DAFTAR PUSTAKA

- [1] Astrom, Karl J. dan Bjorn Wittenmark, "Computer Controlled System ; Theory and Design", Prentice Hall, Inc Engglewood Cliffts, N.J. Jan. 1984.
- [2] Chee, Teng Fong and H.R. Sirisena "Self-Tuning PID Controller for dead Time Process", IEE Trans. On

- Industrial Electronics, Vol 35, No. 1 Apr., 1988
- [3] Feedback Instruments Ltd Crowborough Sussex England, “***Nodular Servo System MS 150 Book 1 DC, Sysncro & Basic Exsperiments***”.
- [4] Gobal K. Debuy, “***Power Semiconductor Controlled Drives***”, Profesor of Electrical Engineering Indian Institut of Thechnology Kapur, India.
- [5] Jhonson, R, “***System Modelling And Identification***”, Prentice Hall Inc, Englewoodf Cliffs, New Jersey, 1993.
- [6] Katsuhiko Ogata, “***Modern Control Engineering***”, Prentic Hall, Inc Englewood Cliffs, N.J. University of Minnesota..
- [7] M. Maeda and S. Murakami, “***A Self-tuning Fuzzy Controller***”, Fuzzy sets Syst, Vol 51, pp 29-40, Feb, 1992
- [8] Sinha, NK, B Kuzta, “***Modelling And Identification of Dinamis System***” Van Northstrat Reinhpld Company, New York, 1983.
- [9] Zhen-Yu Zhao, Masoyorki Tamizuka and Satoru, “***Fuzzy Gain Scheduling Of PID Controllers***”, IEE, Trans. On System Man and Cyberling Vol. 23, No. 5 Sept/Okt, 1993.