

Penerapan Kendali Logika Fuzzy+Proportional Integral pada Modul Process Control Simulator PCS 327 MK2

Wrastawa Ridwan

Jurusan Teknik Elektro
Universitas Negeri Gorontalo
email : r1space@yahoo.com

Abstrak. Respon output yang diperoleh dengan penerapan kontroler logika fuzzy adaptif pada modul Process Control Simulator PCS 327 MK2 ternyata masih kurang memuaskan yaitu *settling time* yang lambat pada saat pembebanan dan *overshoot* yang besar. Pada penelitian ini akan dirancang suatu kontrol hybrid terdiri dari kontroler logika fuzzy adaptif dan kontroler Proportional Integral (PI) yang digunakan untuk mengendalikan respon output dari modul Process Control Simulator PCS 327 MK2. *Gain* Kontrol digunakan untuk mengubah sinyal kontrol sehingga didapatkan respon output dari sistem sesuai dengan yang diinginkan (setpoint). Penambahan kontroler integral digunakan untuk mengurangi overshoot yang ada. Algoritma pembelajaran yang digunakan pada sistem fuzzy adalah gradient descent untuk mengestimasi parameter-parameter fuzzy. Sistem akan diberi gangguan beban setelah mencapai *steady state* sebesar 0,5 volt dan 1 volt. Dalam penelitian ini diperoleh hasil respon output dengan rata-rata *error* terkecil pada saat $K = 1,2$ dan $K_i=0,2$ dengan rata-rata *error* 0,2245 Volt.

Kata kunci : Process Control Simulator PCS 327 MK2, kontrol hybrid

1. Pendahuluan

Process Control Simulator PCS-327 MK-2 adalah simulator analog dengan fungsi tertentu yang memakai rangkaian terpadu *operational amplifier* disusun dengan aturan sedemikian sehingga prinsip-prinsip metode kontrol proses dapat diajarkan pada level teknisi dan teknologis. Secara umum, PCS 327 MK-2 ini terdiri dari bagian *controller* dan bagian proses. Bagian *controller* terdiri dari aksi kontrol *integral*, *derivative* dan *proportional*. Seiring dengan perkembangan teknologi pengaturan kerja sistem, maka pengontrolan sistem atau yang sering disebut *controller* terasa semakin diperlukan keberadaannya. Khususnya di bidang pengontrolan perangkat keras, baik itu berupa alat-alat elektronik maupun alat-alat mekanik, telah dikenalkan berbagai metode pengontrolan yang cukup baik untuk digunakan, salah satunya adalah pengendali logika *fuzzy*.

Pada penelitian ini akan dianalisis penerapan kontrol *hybrid* pada bagian proses dari *Process Control Simulator PCS-327 MK-2* sedangkan *controller-nya* menggunakan *adaptive fuzzy logic controller* dan pengendali PI (*Proportional Integral*). Keterbatasan pengetahuan operator manusia mengenai *plant* yang akan diatur menyebabkan kesulitan dalam menentukan aturan-aturan dasar dalam perancangan *fuzzy logic controller*, sehingga dikembangkan suatu metode untuk mengidentifikasi sistem dengan menerapkan konsep *inversi kausal*. Sedangkan pengendali PI diharapkan dapat mengendalikan respon sistem untuk mengatasi *overshoot* yang ada pada saat pembebanan.

Permasalahan dalam penelitian ini adalah bagaimana mendapatkan respon *output* dari *plant* berupa modul *Process Control Simulator PCS-327 MK2* menggunakan kontrol *hybrid*, yaitu *adaptive fuzzy logic controller* dan pengendali PI (*Proportional Integral*). Identifikasi dan desain kontrol menerapkan metode *inversi kausal* dan metode *gradient descent training* sebagai algoritma pembelajaran. Identifikasi dilakukan secara *online* sehingga didapatkan parameter-parameter dari model *fuzzy* yang nantinya digunakan sebagai acuan untuk menentukan besarnya sinyal kontrol yang diperlukan agar menghasilkan respon *output* yang diinginkan (sama dengan *setpoint*), terutama apabila sistem diberi gangguan beban.

Penelitian ini dibatasi pada penerapan kontrol *hybrid* (*adaptive fuzzy logic controller* dan pengendali PI) pada bagian proses modul *Process Control Simulator PCS-327 MK2*. Kemudian sistem akan diberi gangguan beban setelah mencapai *steady state* sebesar 0,5 volt dan 1 volt.

2. Tinjauan Pustaka

2.1 Sistem *Fuzzy* Adaptif

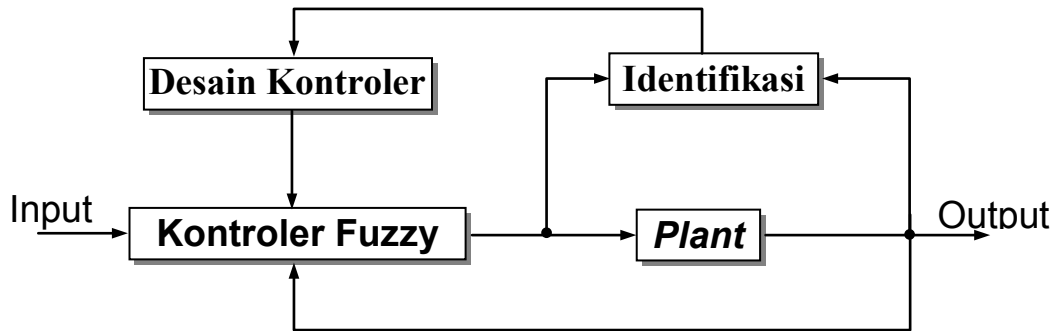
Terdapat dua cara pendekatan untuk *adaptive fuzzy controller*, yaitu *direct adaptive control* dan *indirect adaptive control* [Passino, 1997]. Dalam penelitian ini akan digunakan *indirect adaptive fuzzy controller*.

Pada *indirect adaptive fuzzy controller* dilakukan *on line* identifikasi untuk mengestimasi parameter-parameter *plant* dan bagian “controller designer” akan mengubah parameter dari kontroler. Jika parameter *plant* berubah, identifier akan mengestimasi parameter *plant* dan *controller designer* akan mengubah parameter kontroler. Pendekatan ini dinamakan *indirect adaptive control* karena kita mengubah kontroler secara tidak langsung melainkan terlebih dahulu harus mengestimasi parameter *plant*.

Pada *direct adaptive fuzzy controller* tidak memerlukan perantara model proses atau tahap identifikasi parameter, mendapatkan parameter kontroler secara langsung dari perbandingan aktual performansi loop tertutup dengan beberapa kelakuan yang diinginkan lewat indeks performansi keluaran yang menunjukkan adaptasi kontroler.

Beberapa kelebihan kontroler adaptif secara tak langsung (*indirect adaptive control*) dibandingkan pendekatan secara langsung (*direct adaptive control*):

- Pembangkitan suatu model *plant* membolehkan perubahan parameter yang terdeteksi secara tiba-tiba, sesuai dengan hasil penjejakan dari karakteristik sementara.
- Pemisahan model adaptasi dari disain kontroler, membolehkan model untuk dianalisis secara terpisah dari performansi kontroler dan kestabilan sistem.
- Spesifikasi performansi kontroler dapat diubah untuk mengakomodasi batasan-batasan yang baru.



Gambar 1 Indirect adaptive fuzzy controller

Identifikasi Model Fuzzy

Sistem Logika Fuzzy yang digunakan adalah menggunakan :

- Singleton Fuzzyfier
- Center average defuzzifier
- Aturan penalaran produk (*produk inference engine*)
- Fungsi Keanggotaan Gaussian

Sistem Fuzzy yang akan dirancang mempunyai bentuk :

$$f(x) = \frac{\sum_{l=1}^M \bar{y}^l \left[\prod_{i=1}^N \exp \left(- \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^N \exp \left(- \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}$$

Parameter yang dapat diubah - ubah (*adjustable*) dari sistem logika fuzzy di atas adalah :

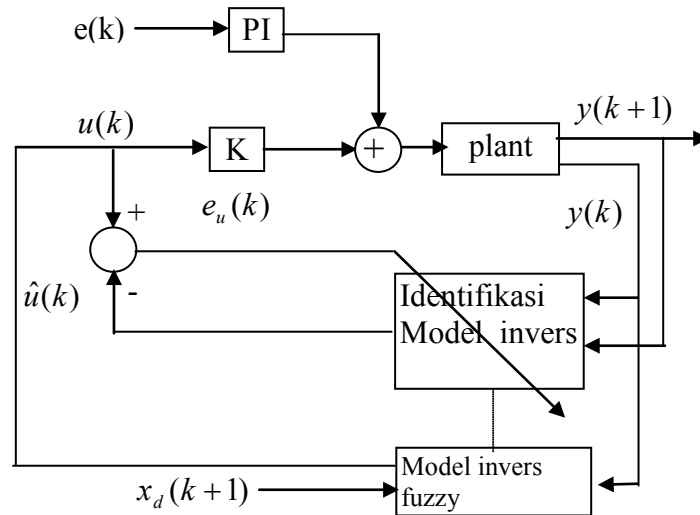
- $\bar{y}^l \in V$
- $\bar{x}_i^l \in U_i$
- σ_i^l

V adalah semesta pembicaraan pada keluaran, sedangkan U_i adalah semesta pembicaraan pada masing-masing masukannya. M adalah banyaknya aturan fuzzy dan N adalah banyaknya masukan. Sedangkan $f(x)$ adalah keluaran sistem fuzzy tersebut. Variabel x_i^l dan σ_i^l ($i = 1, \dots, N; l = 1, \dots, M$) masing - masing adalah titik tengah dan lebar dari fungsi keanggotaan masukan. Masukan dari sistem fuzzy tersebut dinotasikan dengan x_i ($i = 1, \dots, N$), sedangkan titik tengah fungsi keanggotaan di keluarannya adalah \bar{y}^l ($l = 1, \dots, M$).

2.2 Kontrol Hybrid

Kontrol hybrid adalah kontrol yang menggunakan dua atau lebih kontroler dalam satu sistem. Blok diagram dari kontrol hybrid yang terdiri dari *indirect adaptive fuzzy logic*

controller dan kontroler *Proportional Integral* (PI) secara keseluruhan dapat dilihat pada Gambar 2.



Gambar 2 Diagram blok kontrol *hybrid*

Jika prosedur pemodelan fuzzy secara adaptif digunakan untuk menyediakan pemodelan proses dalam desain kontroler dilakukan secara online, maka hasilnya adalah berupa suatu kontroler fuzzy adaptif secara tak langsung (*indirect adaptive fuzzy logic controller*). Sedangkan penambahan kontroler PI adalah untuk mengurangi *overshoot*. Persamaan kontroler PI adalah sebagai berikut.

$$U(k) = K_i * (e(k) + e(k+1))$$

Struktur kontroler ini terdiri atas dua buah umpan balik, yaitu umpan balik untuk kontroler dan umpan balik untuk *update* pemodelan plant. Umpan balik untuk kontroler menggunakan hasil observasi keluaran untuk menentukan sinyal kontrol yang dibutuhkan yang digunakan sehingga dihasilkan respon yang diinginkan, umpan balik kontroler, sedangkan umpan balik untuk *update* pemodelan proses menggunakan observasi masukan/keluaran plant untuk mengadaptasi model invers proses yang dibutuhkan untuk umpan balik kontroler.

3 Metode Penelitian

Metode pengerjaan penelitian ini dibagi dalam beberapa tahap :

1. Mengumpulkan bahan literatur.
2. Mempelajari teori sistem logika *fuzzy* untuk identifikasi dan *controller*.
3. Membuat *software* untuk identifikasi dan *controller*
4. Penyiapan hardware (PCS 327 MK-2, komputer, ADC/DAC).
5. Mengumpulkan data dengan melakukan proses identifikasi sistem dan menerapkan *controller* pada sistem (bagian proses dari modul *Process Control Simulator PCS 327 MK2*) dengan perubahan *gain* kontrol.

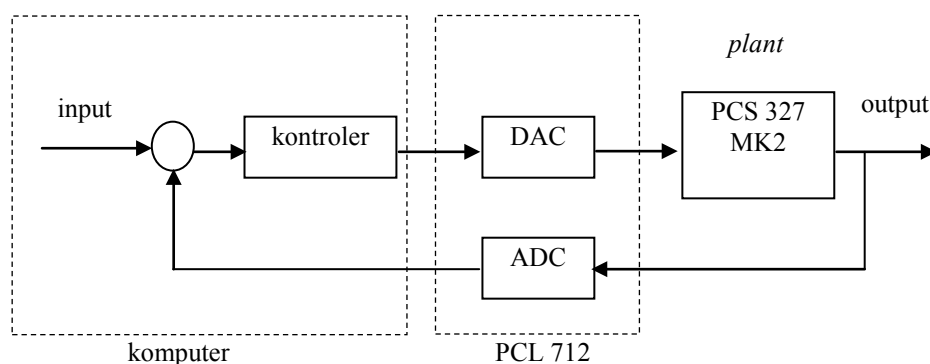
6. Analisis data untuk menentukan spesifikasi respon sistem.

7. Pembuatan laporan penelitian

Penelitian ini akan dilakukan pada Laboratorium Teknik Pengaturan, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, khususnya pada tahap penyiapan hardware, pengumpulan dan analisis data.

Perangkat keras yang digunakan dalam penelitian ini berupa 1 (satu) set komputer untuk mengolah data secara *on-line*, peralatan ADC/DAC berupa multi lab Card PCL-712 dan *plant* yang dikontrol berupa modul PCS 327 MK2. Hubungan antar perangkat keras diperlihatkan dalam Gambar 3.

Perangkat lunak yang digunakan untuk mengimplementasikan algoritma kontrol menggunakan bahasa pemrograman C.



Gambar 3 Blok diagram perancangan sistem

Sebelum diterapkan kontroler logika *fuzzy* adaptif perlu dilakukan proses identifikasi terlebih dahulu. Hasil dari proses identifikasi tersebut berupa parameter-parameter *fuzzy* yang kemudian digunakan untuk kontroler logika *fuzzy* adaptif. Proses pembelajarannya menggunakan algoritma *gradient descent*.

Untuk lebih jelasnya, proses perancangan kontroler logika *fuzzy* adaptif yang diterapkan pada modul *Process Control Simulator PCS 327 MK2*, melalui algoritma sebagai berikut :

a. Algoritma identifikasi *plant*

1. Modul *Process Control Simulator PCS 327 MK2* diberi masukan *step* sebesar 3.8 volt.
2. Menentukan jumlah rule (= 30)
3. Menentukan inisialisasi parameter model *invers fuzzy* yaitu $\bar{x}_1^l(0)$, $\bar{x}_2^l(0)$, $\bar{y}^l(0)$, $\bar{\sigma}_1^l(0)$, $\bar{\sigma}_2^l(0)$ yang nantinya akan diestimasi.
4. Mengestimasi parameter-parameter tersebut dengan algoritma *gradient descent* berdasarkan data masukan/keluaran *plant*.
5. Data parameter-parameter dari langkah 4 digunakan untuk inisialisasi kontroler.

b. Algoritma Kontroler *Hybrid*

1. Hasil identifikasi model invers fuzzy yang berupa parameter-parameter fuzzy digunakan untuk inisialisasi kontroler fuzzy adaptif.
2. Gangguan beban diberikan pada saat respon loop tertutup dari plant mencapai steady state. Beban diberikan sebesar 0.5 volt (13,15%) dan 1 volt (26,31%).
3. Memberikan input kontrol gain (K) yang berubah kemudian menghitung Mean Error (ME).
4. Pada K yang sama diberikan input kontrol gain integral (Ki) yang berbeda.
5. Menganalisis respon sistem untuk mengetahui karakteristik performansi sistem meliputi time settling (ts) pada saat transien dan pembebanan, maksimum overshoot (Mp) dan persen maksimum overshoot (%Mp), output steady state, error *steady state* serta seberapa besar rata-rata error (*Mean Error*).

Software ini dibuat untuk menangani akuisisi data dari modul Process Control Simulator dan mengoleksi data input-output untuk identifikasi secara online, serta untuk implementasi pengaturan dengan kontroler hybrid (logika fuzzy adaptif dan PI).

4 Hasil Penelitian Dan Pembahasan

4.1 Hasil Identifikasi Plant

Identifikasi sistem dilakukan secara *online* dengan memberikan masukan *step* sebesar 3,8 volt pada *plant*. Identifikasi yang dilakukan adalah identifikasi model *invers fuzzy* dari *plant* untuk mendapatkan parameter-parameter sistem *fuzzy* dengan menggunakan algoritma pembelajaran *gradient descent* dengan ketentuan sebagai berikut :

- Jumlah *rule* (M) = 30, Kecepatan pembelajaran (α) = 0,5

Parameter-parameter sistem fuzzy hasil dari identifikasi dapat dilihat pada Tabel 1 [Ridwan, 2007].

Parameter-parameter hasil identifikasi diatas nantinya digunakan untuk inisialisasi parameter pada kontroler logika *fuzzy* adaptif secara *online*. Data parameter-parameter diatas terlihat bahwa nilai \bar{x}_1^l mengalami perubahan cukup besar dari rule ke 1 sampai ke 15, rule ke 16 sampai 29 tidak banyak berubah dan berubah lagi pada rule ke 30. Untuk nilai \bar{x}_2^l tidak mengalami perubahan setelah rule ke 22. Nilai \bar{y}^l tidak mengalami perubahan mulai rule ke 15 sampai 27 dan berubah sedikit pada rule ke 28 samapi rule ke 30. Nilai $\bar{\sigma}_2^l$ cenderung tetap, sedangkan nilai $\bar{\sigma}_1^l$ berubah dari rule ke 1 sampai 29 dan berubah drastis pada rule 30 [Ridwan, 2007].

4.2 Respon Sistem Tanpa Kontroler PI

Respon keluaran setelah diimplementasikan kontroler logika *fuzzy* adaptif jika diberikan masukan *step* sebesar 3,8 volt dengan kontrol *gain* yang bervariasi seperti terlihat pada Tabel 2 dan Tabel 3 [Ridwan, 2007].

Tabel 1 Parameter-Parameter $\bar{x}_1^l, \bar{x}_2^l, \bar{y}^l, \bar{\sigma}_1^l, \bar{\sigma}_2^l$ Sistem Fuzzy Hasil Identifikasi

Rule	\bar{x}_1^l (V)	\bar{x}_2^l (V)	\bar{y}^l (V)	$\bar{\sigma}_1^l$ (V)	$\bar{\sigma}_2^l$ (V)
1	0.6268473761	3.0500610500	3.4168127115	3.6446857694	0.0500000000
2	0.5881927408	3.0500610500	3.1784460826	3.6446857694	0.0500000000
3	0.8579908202	3.0500610500	2.4620093940	3.8796155001	0.0500000000
4	1.0618599637	3.2063492063	1.5919073582	3.8986814149	0.0500000000
5	1.6083521280	3.2063492063	1.0031400384	4.3459853749	0.0500000000
6	1.7946944795	3.3626373626	0.6514397262	4.3001862782	0.0500000000
7	2.4477608092	3.3626373626	0.3047860891	4.5940793200	0.0500000000
8	2.7516889183	3.3626373626	0.0831116144	4.3474977849	0.0500000000
9	3.0981920489	3.3626373626	-0.041106747	4.4427051556	0.0500000000
10	3.1201334242	3.5189255189	-0.186007371	4.4614273753	0.0500000000
11	3.2958200965	3.5189255189	-0.310024055	4.4977758296	0.0500000000
12	3.4360228654	3.5189255189	-0.378948859	4.3231318135	0.0500000000
13	3.5719326341	3.5189255189	-0.411089448	4.3428175513	0.0500000000
14	3.6807137855	3.5189255189	-0.437764554	4.1725253782	0.0500000000
15	3.7478390179	3.5189255189	3.7999999999	4.1790979614	0.0500000000
16	3.7486254407	3.6752136752	3.7999999999	4.2289555727	0.0500000000
17	3.7821402191	3.6752139470	3.7999999999	4.2326570255	0.0500008496
18	3.8216375798	3.6752139470	3.7999999999	4.0868048292	0.0500008496
19	3.8544592207	3.6752139470	3.7999999999	4.0881946683	0.0500008496
20	3.9177562854	3.6752139470	3.7999999999	4.0216559066	0.0500008496
21	3.9392872794	3.6799724815	3.7999999999	4.0221217483	0.0648749014
22	3.9355882148	3.8315018315	3.7999999999	4.1050773056	0.0500000000
23	3.9462880257	3.8315047017	3.7999999999	4.1053234543	0.0500089708
24	3.9700170073	3.8315047017	3.7999999993	4.0513128230	0.0500089708
25	3.9740670116	3.8315047017	3.7999999993	4.0513559643	0.0500089708
26	3.9740670116	3.8315047017	3.7999999993	4.0513559643	0.0500089708
27	3.9775356182	3.8315047017	3.7999888812	4.0513936251	0.0500089708
28	3.9980878646	3.8315047017	3.6053363235	4.0028210263	0.0500089708
29	2.8937728937	3.8315047017	3.6053363235	3.8315018315	0.0500089708
30	3.0500610500	3.8315047017	3.6053363235	0.0500000000	0.0500089708

Tabel 2 Spesifikasi kontrol dengan K bervariasi, tanpa beban

Spesifikasi Gain (K)	t_s (det)	M_p (V)	O_{ss} (V)	E_{ss} (V)	Rata-rata error (V)
1,0	7,4699	0,4688	3,8315	0,0315	0,4364
1,1	7,1899	0,7814	3,8315	0,0315	0,4103
1,2	6,5899	0,9377	3,8315	0,0315	0,3873
1,3	9,9399	1,0940	3,8315	0,0315	0,3707

Tabel 3 Spesifikasi kontrol dengan K bervariasi, berbeban

Spesifikasi <i>Gain</i> (K)	Beban 0,5 V		Beban 1 V	
	t_{s1} (det)	t_{s2} (det)	t_{s3} (det)	t_{s4} (det)
1,0	6,0899	4,8899	6,2599	7,4699
1,1	4,6199	4,2299	5,2799	6,9199
1,2	4,0699	3,3499	4,4499	4,8899
1,3	3,2399	2,8599	3,7899	4,7799

Berdasarkan spesifikasi kontrol diatas terlihat bahwa rata-rata *error* untuk $K = 1,0$ paling besar disebabkan oleh *time settling* (t_s) lambat baik pada saat *transient* maupun pada saat pembebanan. Rata-rata *error* paling kecil pada saat $K = 1,3$ karena mempunyai t_s yang paling cepat pada saat pembebanan. Pada saat pembebanan, diperoleh t_s yang paling cepat pada $K = 1,3$. Namun K yang besar ini mengakibatkan t_s pada saat *transien* dan *maximum overshoot* menjadi besar juga.

Data diatas memperlihatkan bahwa makin besar harga K , harga M_p makin besar dan t_s pada saat pembebanan makin cepat [Ridwan, 2007]. Diperoleh pula bahwa harga M_p dan t_s masih terlalu besar, sehingga spesifikasi ini yang akan diperbaiki dengan penerapan kontrol hybrid, dengan menambahkan kontroler PI pada sistem.

4.3 Respon Sistem dengan Kontroler Hybrid

Kontrol *hybrid* pada penelitian ini adalah kontrol fuzzy adaptif dan kontrol *Proportional Integral* (PI). Penambahan kontroler PI disini untuk mengurangi *maximum overshoot* (M_p) dan mempercepat transien, terutama pada saat pembebanan. Pada percobaan ini diberikan *setpoint* 3,8 volt dengan kontrol gain (K) dan konstanta integrator (K_i) yang bervariasi. Penambahan tegangan beban diberikan ketika sistem telah mencapai *steady state*.

Tabel 4 Spesifikasi kontrol untuk $K_i = 0,2$ dan K bervariasi

Harga K	Parameter							
	t_s (det)	M_p (V)	E_{ss} (V)	Pembebanan				Rerata <i>Error</i> (V)
				t_{s1} (det)	t_{s2} (det)	t_{s3} (det)	t_{s4} (det)	
1	2,69	0,1563	0,0315	6,1	4,51	8,79	7,14	0,3047
1,1	6,22	0,3126	0,0315	4,99	3,95	6,48	6,98	0,2929
1,2	5,43	0,4688	0,0315	3,96	3,14	4,99	4,61	0,2885
1,3	5,44	0,7814	0,0315	3,24	2,69	5,06	3,68	0,3006

Dari data diatas diperoleh, semakin besar harga K , harga t_s pada saat pembebanan makin kecil sehingga harga rata-rata *error* makin kecil. *Error steady state* (E_{ss}) sama untuk semua harga K . Harga M_p sebanding dengan kenaikan harga K . Diperoleh respon terbaik pada $K=1,2$ dengan harga rata-rata *error* paling kecil.

Tabel 5 Spesifikasi kontrol untuk $K_i = 0,25$ dan K bervariasi

Harga K	Parameter							
	t_s (det)	M_p (V)	E_{ss} (V)	Pembebanan				Rerata Error (V)
				t_{s1} (det)	t_{s2} (det)	t_{s3} (det)	t_{s4} (det)	
1	2,2	0,1563	0,0315	6,54	4,34	8,95	7,58	0,2876
1,1	4,23	0,3126	0,0315	5,16	3,68	7,31	6,99	0,2820
1,2	4,99	0,4688	0,0315	4,06	3,14	5,44	4,61	0,2925
1,3	5,5	0,4688	0,0315	3,35	2,85	4,22	3,68	0,2560

Dari data diatas diperoleh, semakin besar harga K , t_s pada saat transien dan M_p makin besar. Pada saat pembebanan, semakin besar harga K , semakin kecil harga t_s . Kecenderungan nilai rata-rata *error* makin kecil dengan kenaikan harga K . Diperoleh respon terbaik pada $K=1,3$ dengan harga rata-rata *error* paling kecil.

Tabel 6 Spesifikasi kontrol untuk $K_i = 0,3$ dan K bervariasi

Harga K	Parameter							
	t_s (det)	M_p (V)	E_{ss} (V)	Pembebanan				Rerata Error (V)
				t_{s1} (det)	t_{s2} (det)	t_{s3} (det)	t_{s4} (det)	
1	2,14	0	0,0315	6,54	4,62	9,72	7,86	0,2524
1,1	3,62	0,1563	0,0315	5,16	3,68	7,69	6,37	0,2532
1,2	3,95	0,3126	0,0315	4,18	3,13	7,10	5,00	0,2245
1,3	9,34	0	0,0315	6,1	3,68	6,87	6,37	0,3641

Dari data diatas diperoleh, semakin besar harga K , t_s pada saat transien makin besar. Harga M_p relatif kecil, bahkan pada $K = 1$ dan $K = 1,3$, tidak terdapat *overshoot*. Pada saat pembebanan, semakin besar harga K , kecenderungan harga t_s semakin kecil. Kecenderungan nilai rata-rata *error* makin kecil dengan kenaikan harga K . Diperoleh respon terbaik pada $K = 1,2$ dengan harga rata-rata *error* paling kecil.

Secara umum diperoleh dari analisis respon output sistem, bahwa kenaikan harga K membuat harga t_s saat *transien* semakin besar, harga M_p semakin besar dan harga t_s saat pembebanan semakin kecil. Kenaikan harga K_i mengakibatkan harga t_s saat transien makin besar dan harga M_p semakin kecil. Nilai t_s saat pembebanan cenderung mengalami kenaikan.

Analisis data secara keseluruhan, diperoleh rata-rata error yang terkecil pada saat $K_i = 0,3$ dan $K = 1,2$.

5 Kesimpulan

1. Kenaikan harga K membuat harga t_s pada saat *transien* semakin besar, harga M_p semakin besar dan harga t_s pada saat pembebanan semakin kecil.
2. Kenaikan harga K_i mengakibatkan harga t_s pada saat transien makin besar dan harga M_p semakin kecil. Nilai t_s saat pembebanan cenderung mengalami kenaikan.
3. Secara keseluruhan diperoleh rata-rata *error* yang terkecil pada saat $K_i = 0,3$ dan $K = 1,2$.

Daftar Pustaka

- 1 Jang, J.S.R., Sun C.T., Mizutani, E., 1997, *Neuro-Fuzzy Fuzzy and Soft Computing : A Computational Approach to Learning and Machine Intelligence*, New Jersey : Prentice Hall International, Inc.
- 2 Landau, I.D., 1990, *System Identification and Control Design Using P.I.M + Software*, New Jersey : Prentice Hall International, Inc.
- 3 Ogata, Katsuhiko, 1997, *Teknik Kontrol Automatik Jilid I*, Jakarta : Erlangga.
- 4 Passino, Kevin M., Yurkovich, S., 1997, *Fuzzy Control*, New York : Addison Wesley Logman.
- 5 Ridwan, Wrastawa, 2007, *Penerapan Adaptive Fuzzy Logic Controller Pada Modul Process Control Simulator PCS 327 MK2*, Gorontalo : Penelitian Dosen Muda.
- 6 Wang, Li-Xin, 1997, *A Course in Fuzzy Systems and Control*, New Jersey : Prentice-Hall International, Inc.
- 7 Yan, J., Ryan, M., Power, J., 1994, *Using Fuzzy Logic Toward Intelleigent Systems*, Cambridge : Prentice-Hall International (UK) Limited.
- 8, 1998, *Model PCL-712 : Multi-Lab (12 bit) A/D+D/A+DIO*, London : Advantech Co. Ltd.
- 9, 1998, *Process Control Simulator PCS327 Book 1*, London : Feedback Instruments Ltd.