

Simulasi Kontroler PID *Tuning* Menggunakan Logika Fuzzy dan Algoritma Genetika Sebagai Pengendali Kecepatan Motor DC

Iqlimah Khadari¹, Subiyanto¹, Dhidik Prastiyanto¹

¹Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Semarang, Jawa Tengah.

Informasi Artikel

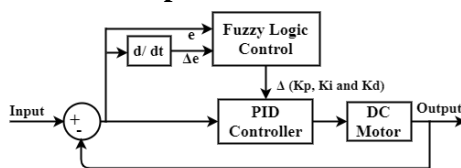
Naskah Diterima : 15 Oktober 2019

Direvisi : 22 Oktober 2019

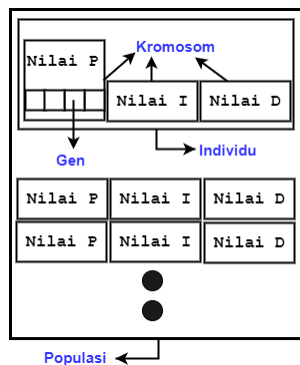
Disetujui : 20 Desember 2019

*Korespondensi Penulis:
darikhadari@gmail.com

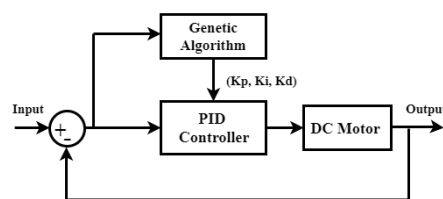
Graphical abstract



Gambar 2.1 Struktur Self - tuning fuzzy PID controller



Gambar 2.2 Ilustrasi kromosom dalam GA based PID controller



Gambar 2.3 Struktur GA based PID

Abstract

The majority of industrial control systems still use PID controller. Tuning of PID controller is important process to applied PID controller. Optimization of PID controller (K_p , K_i and K_d) parameters is the key goal in industrial control systems. However, tuned the PID controller parameters by empirical method not optimum. This paper presents a fuzzy logic controller (self-tuning fuzzy PID) and genetic algorithm (GA based PID). Self-tuning fuzzy PID and GA based PID applied to simulation of a DC motor speed control. Several tests are carried out to investigate the performances of both controllers. Simulation results have demonstrated that the use of self-tuning fuzzy PID controller results in under Case D and Case E testing gives better performance with less overshoot. On the other hand, GA based PID controller has better performance under Case B and C testing.

Keywords: simulation, PID, fuzzy logic, genetic algorithm, speed of DC motor.

Abstrak

Sistem kendali PID masih banyak digunakan dalam industri sistem kendali. Tuning kontroler PID merupakan salah satu proses penting dalam implementasi kontroler PID. Tuning kontroler PID secara empirik seringkali mendapatkan hasil yang tidak optimal. Oleh karena itu, penelitian ini menyajikan kontroler PID yang dituning menggunakan logika fuzzy (self-tuning fuzzy PID) dan algoritma genetika (GA based PID). Desain kontroler PID diimplementasikan dalam bentuk simulasi sebagai pengendali kecepatan motor DC. Beberapa pengujian pada simulasi dilakukan untuk melihat performa kinerja dari kedua sistem kontrol. Hasil simulasi menunjukkan bahwa self-tuning fuzzy PID menunjukkan kinerja sistem yang baik dengan sedikitnya nilai overshoot saat pengujian sistem pada Case D dan E. Sedangkan GA based PID controller mampu bertahan di kisaran nilai set point saat diuji pada Case B dan Case C.

Kata kunci: simulasi, Kontroler PID, logika fuzzy, algoritma genetika, kecepatan, motor DC.

© 2019 Penerbit Jurusan Teknik Elektro UNTIRTA Press. All rights reserved

1. PENDAHULUAN

Penggunaan motor listrik, terutama motor DC dalam bidang industri cukup dominan. Mulai dari peralatan rumah tangga, air conditioner, lift, robot[1] hingga alat transportasi saat ini ada yang menggunakan motor DC[2].

Teknologi yang digunakan dalam mengatur kecepatan motor DC bermacam – macam. Metode kontrol secara konvensional seperti; mengatur tegangan jangkar[3-4], dan menggunakan kontroler PID. *Tuning* kontroler PID merupakan salah satu proses penting dalam implementasi kontroler PID. Mendapatkan parameter kontroler PID yang sesuai dengan karakteristik sistem bukan hal yang mudah. Ziegler – Nichols adalah metode *tuning* kontroler PID yang paling umum digunakan[5-6]. Meskipun bagus, *tuning* PID menggunakan Ziegler – Nichols tidak selalu

menampilkan hasil respon yang optimal. Respon transien yang dihasilkan bisa saja lebih buruk apabila objek (*plant*) mengalami perubahan dinamis.

Logika fuzzy atau fuzzy logic controller (FLC) merupakan salah satu metode yang dapat digunakan untuk *tuning* kontroler PID. Kombinasi antara FLC dan PID disebut **self – tuning fuzzy PID controller**. Penelitian terkait performa kinerja antara kontroler PID konvensional dengan *self – tuning* Fuzzy PID pun dilakukan oleh beberapa peneliti[7-9]. *Self - tuning* fuzzy PID menampilkan hasil simulasi lebih baik dengan kecepatan konstan saat terjadi pembebanan apabila dibandingkan dengan kontroler PID konvensional. Salah satunya pembebanan dalam motor DC adalah beban dengan variabel torsi yang bervariasi dengan kecepatan operasi konstan (contoh; kipas angin)[10].

Selain itu, report penelitian lain melibatkan teori evolusi khususnya algoritma genetik (*genetic algorithm*) sebagai *tuning* kontroler PID. *GA based* PID nama lain dari kontroler PID yang *dituning* menggunakan algoritma genetik. Peran algoritma genetik (GA) dalam kontroler PID adalah untuk mendapatkan nilai parameter PID (K_p , K_i , dan K_d) yang optimal[11-14]. Bahkan, Krohling dan Rey[15] membuktikan bahwa penggunaan *GA based* PID mampu membuat kecepatan motor DC tetap stabil saat menghadapi pembebanan.

Pada penelitian ini, penulis menggunakan *self-tuning* Fuzzy PID dan *GA based* PID pada simulasi pengendali kecepatan motor DC. Analisis kinerja kedua metode dinilai berdasarkan respon transien pada beberapa kasus yang diujikan pada simulasi. Parameter yang digunakan untuk analisis sistem kendali adalah error keadaan tunak (steady state error, E_{ss}), waktu tunda (delay time, t_d), waktu naik (rise time, t_r), waktu puncak (peak time, t_p), lewatan maksimum (maximum overshoot, M_p) dan waktu tunak (settling time, t_s).

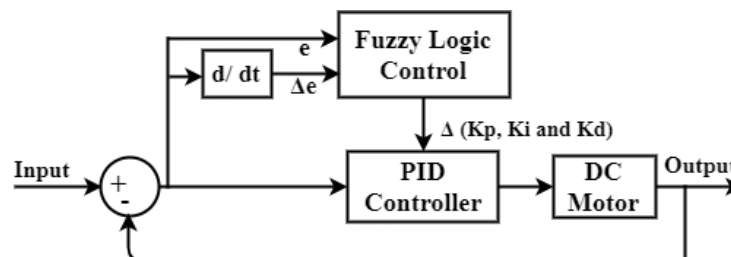
2. METODE PENELITIAN

Metode penelitian yang digunakan adalah metode penelitian eksperimen. Jaedun[15] menyatakan bahwa penelitian eksperimen adalah penelitian kausal (sebab akibat) yang pembuktiannya diperoleh melalui komparasi atau perbandingan. Dua kelompok yang akan dibandingkan tersebut harus komparabel, dalam kondisi yang sama. Pengumpulan data sekunder didapat melalui *study literature* (skripsi, tesis, buku teks, jurnal) serta informasi pendukung lain dari internet. Data sekunder yang didapat digunakan sebagai acuan mengisi nilai parameter dari tiap komponen yang terdapat pada simulasi. Adapun langkah – langkah pengambilan data yang dilakukan saat eksperimen melalui simulasi adalah;

1. Membuat *FIS* (*Fuzzy Inference System*) berdasarkan jurnal acuan.
2. *Tuning* kontroler PID menggunakan algoritma genetika dengan jumlah generasi 50 generasi.
3. Mengimplementasikan *self-tuning* Fuzzy PID dan *GA based* PID pada simulasi sistem pengendalian kecepatan motor DC.
4. Melakukan pengujian pada simulasi dengan menjalankan simulasi pada beberapa kondisi (Case A – Case E).
5. Data respon transien *self-tuning* Fuzzy PID dan *GA based* PID pada setiap kondisi pengujian diamati, dan dianalisis.

2.1 Self-tuning Fuzzy PID

Kontribusi utama *fuzzy logic controller* (FLC) pada proses *self-tuning* fuzzy PID ada pada penyesuaian FLC yang bersifat *adaptive* sebagai *input*[17]. Gambar 2.1, mendeskripsikan FLC digunakan sebagai *tuning* parameter kontroler PID dengan *error* (e) dan *delta error* (Δe) sebagai masukan (*input*), dan parameter delta K_p , K_i , dan K_d sebagai keluaran FLC. Tahapan dalam *self-tuning* fuzzy PID terdiri dari fuzzifikasi, system inferensi fuzzy dan defuzzifikasi.



Gambar 2.1 Struktur *Self - tuning fuzzy* PID controller

Dalam penerapan *self-tuning* Fuzzy PID, menimbulkan persamaan (2.1) karena parameter PID diupdate secara otomatis oleh *fuzzy logic control*.

$$U^{PID} = Kp_2 e + Ki_2 \int e + Kd_2 \frac{de}{dt} \tag{2.1}$$

Keterangan Kp_2 , Ki_2 , dan Kd_2 adalah nilai koefisien baru *self-tuning* fuzzy PID[7],[18-19].

2.1.1 Fuzzifikasi (Fuzzification)

Fuzzifikasi adalah proses transformasi variabel numerik (*crisp variable*) ke dalam variabel linguistik. Nilai *error* (e) dan *delta error* (Δe) dinormalisasi dari variabel numerik (-1,1) ke dalam variabel linguistik yang diberi label sebagai **NB** (*Negative Big*), **NM** (*Negative Medium*), **NS** (*Negative Small*), **ZE** (*Zero*), **PS** (*Positive Small*), **PM** (*Positive Medium*) dan **PB** (*Positive Big*). Variabel linguistik Kp , Ki , dan Kd yang berperan sebagai keluaran diberi label sebagai **ZE** (*Zero*), **MS** (*Medium Small*), **S** (*Small*), **M** (*Medium*), **B** (*Big*), **MB** (*Medium Big*), dan **VB** (*Very Big*). Variabel numerik untuk Kp dan Ki dinormalisasi dari (0, 20). Variable numerik untuk Kd dinormalisasi dari nilai (0, 0.2).

2.1.2 Sistem inferensi Fuzzy (Fuzzy Inference System)

Tabel 2.1 – 2.3 menunjukkan matriks keputusan yang digunakan (sistem inferensi fuzzy) pada *self-tuning* fuzzy PID[7],[18-19].

Tabel 2.1 Aturan fuzzy untuk Kp

$\Delta e/e$	NB	NS	ZE	PS	PB
NB	VB	VB	VB	VB	VB
NS	B	B	B	MB	VB
ZE	ZE	ZE	MS	S	S
PS	B	B	B	MB	VB
PB	VB	VB	VB	VB	VB

Tabel 2.2 Aturan fuzzy untuk Ki

$\Delta e/e$	NB	NS	ZE	PS	PB
NB	M	M	M	M	M
NS	S	S	S	S	S
ZE	MS	MS	ZE	MS	MS
PS	S	S	S	S	S
PB	M	M	M	M	MS

Tabel 2.3 Aturan fuzzy untuk Kd

$\Delta e/e$	NB	NS	ZE	PS	PB
NB	ZE	S	M	MB	VB
NS	S	B	MB	VB	VB
ZE	M	MB	MB	VB	VB
PS	B	VB	VB	VB	VB
PB	VB	VB	VB	VB	VB

2.1.3 Defuzzifikasi (Defuzzification)

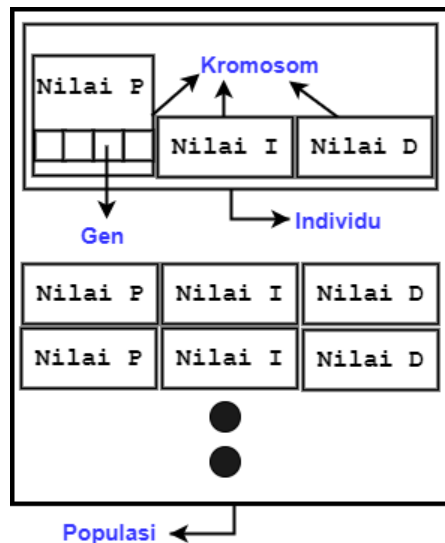
Model defuzzifikasi yang digunakan adalah *center of gravity* (COG). Persamaan (2.2) adalah persamaan yang digunakan untuk mendapatkan defuzzifikasi[7]. Keterangan (i=1,2,3...) dan u (xi) adalah nilai keanggotaan dari elemen xi, u (COG) adalah *output* dari kontrol logika Fuzzy.



$$u(COG) = \frac{\sum_{i=1}^n u(x_i) \cdot x_i}{\sum_{i=1}^n u(x_i)} \tag{2.2}$$

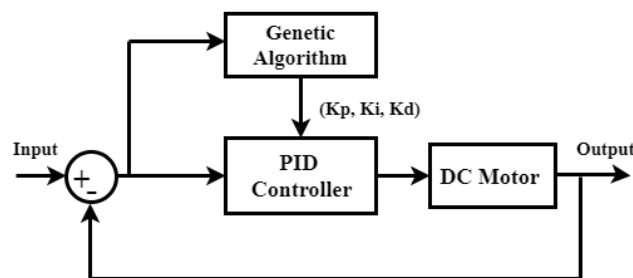
2.2 GA based PID

Proses dalam GA based PID diawali dengan **inisialisasi populasi**, yaitu menciptakan individu secara acak yang memiliki susunan gen (*chromosome*) tertentu. Gen berisi nilai bilangan *real* dari nilai kontroler P, nilai kontroler I, dan nilai kontroler D yang dibuat secara *random*. Nilai P dirandom dari nilai 0 hingga 20, nilai I dirandom dari 0 hingga 20, dan nilai D akan dirandom dari nilai 0 hingga 5. Ilustrasi perumusan kromosom dalam GA based PID ditunjukkan Gambar 2.2.



Gambar 2.2 Ilustrasi kromosom dalam GA based PID controller

Individu dari populasi yang bertahan hidup pada generasi berikutnya dipilih melalui **proses seleksi**. Kemudian bereproduksi melalui proses **crossover dan mutasi** [20] dalam [21]. Gambar 2.3 menampilkan struktur dari GA based PID controller. Tabel 2.4 Memuat nilai parameter algoritma genetika. Nilai parameter algoritma genetika sebagai input dalam proses algoritma genetika, sedangkan keluaran dari algoritma genetika adalah nilai Kp, Ki dan Kd yang optimal. Nilai Kp, Ki dan Kd kembali diinputkan pada kontrol PID dalam simulasi pengendali kecepatan motor DC.



Gambar 2.3 Struktur GA based PID

Tabel 2.4 Parameter Algoritma Genetika yang digunakan

No	Parameter	Value
1.	Population	100
2.	Generations	50
4.	Crossover fraction	0.65
5.	Lower bound [Kp Ki Kd]	[0 0 0]
6.	Upper bound [Kp Ki Kd]	[20 20 5]

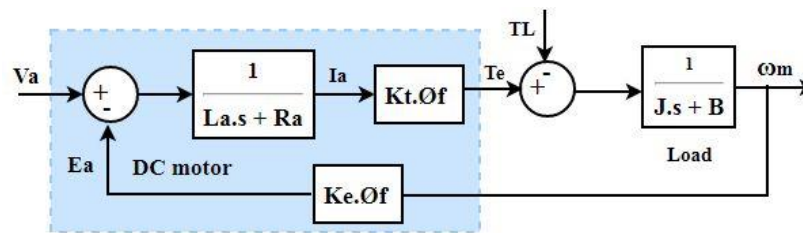


2.3 Implementasi Kontroler PID sebagai Pengendali Kecepatan Motor DC

Simulasi pengendalian kecepatan motor DC dilakukan pada software MATLAB. Gambar 2.4 adalah diagram blok sistem loop tertutup sebuah motor DC yang merepresentasikan persamaan (2.3). Rangkaian simulasi terdiri dari beberapa blok yang saling terhubung dengan fungsi yang berbeda. Blok utama adalah blok motor DC, kemudian bloklain yang saling mendukung seperti; blok pengendali kecepatan, pengendali arus, pensaklaran dan sebagainya.

Tabel 2.5 Nilai parameter motor DC yang digunakan

Spesifikasi motor DC : 5 HP, 240 Volts	
Parameters	Value
Armature inductance (Henry)	$L_a = 0.01 \text{ H}$
Armature resistance (ohm)	$R_a = 0.5 \Omega$
Rotor Inertia (kg.m ²)	$J = 0.05 \text{ kg.m}^2$
Armature voltage (Volt)	$V_a = 280 \text{ V}$
Frictional constant of motor (Nm(rad/sec))	$B = 0.02 \text{ Nm/A}$
Back emf constant (V (rad/sec))	$K_b = 1.23 \text{ V (rad/sec)}$



Gambar 2.4 Diagram blok sistem loop tertutup motor DC[22].

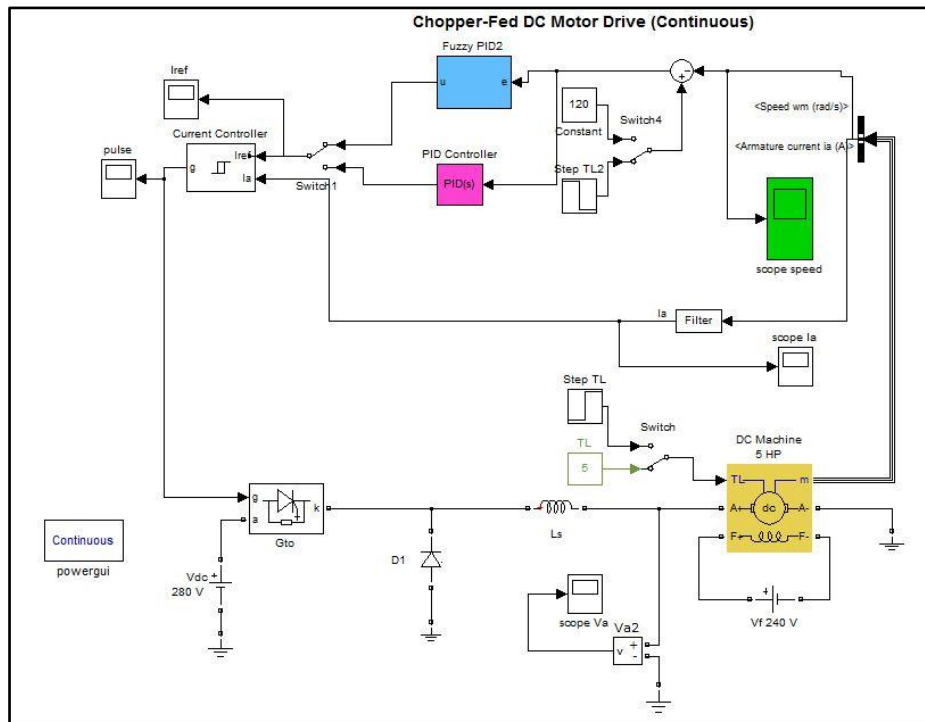
Dengan memasukkan nilai parameter yang ada pada Tabel 2.5 ke dalam persamaan (2.3), maka didapatkan fungsi alih motor DC yang digunakan seperti pada persamaan (2.4) :

$$\frac{\omega_m}{V_a} = \frac{k}{(JL_a)s^2 + (RaJ + BL_a)(s) + (k^2 + Ra)} \quad (2.3)$$

$$G(s) = \frac{1.23}{0.005s^2 + 0.0252s + 1.523} \quad (2.4)$$

Pada Gambar 2.5 tampak sebuah motor arus searah disuplai tegangan arus searah melalui rangkaian *chopper* arus searah. Rangkaian *chopper* terdiri dari thyristor GTO dan sebuah diode yang diseri dengan induktor. Pengendali arus histerisis menghasilkan sinyal penyalan thyristor GTO yang memaksa arus jangkar bernilai sama dengan arus referensi. Sedangkan rangkaian pengendali kecepatan menggunakan *self - tuning Fuzzy PID* dan *GA based PID*. Simulasi kecepatan motor ditampilkan melalui grafik pada *scope speed*. Simulasi diujikan pada *Case A* sampai dengan *Case E* ;

1. Case A, kondisi dengan kecepatan tetap (120 rad/s) dan beban torsi tetap (5 Nm).
2. Case B, kondisi dengan kecepatan tetap (120 rad/s) dan beban torsi meningkat (dari beban torsi 5 Nm menjadi 20 Nm).
3. Case C, kondisi dengan kecepatan tetap (120 rad/s) dan beban torsi menurun (dari beban torsi 20 Nm menjadi 5 Nm).
4. Case D, kondisi dengan kecepatan meningkat (dari kecepatan 120 rad/s hingga 170 rad/s, interval peningkatan 10 rad/s setiap 0.5 sekon) dan beban torsi tetap (5 Nm).
5. Case E, kondisi dengan kecepatan menurun (dari kecepatan 170 rad/s hingga 120 rad/s, interval peningkatan 10 rad/s setiap 0.5 sekon) dan beban torsi tetap (5 Nm).

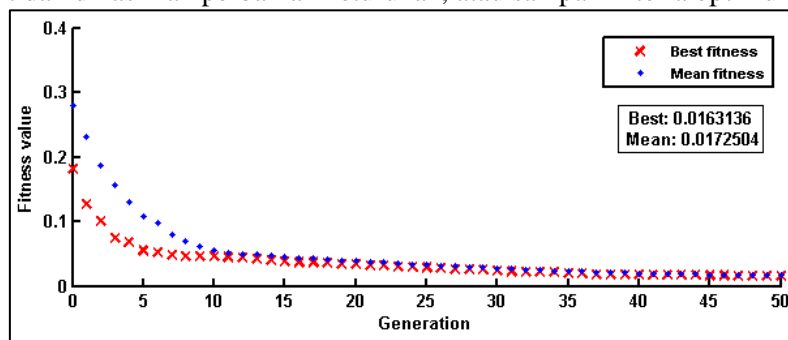


Gambar 2.5 Rangkaian simulasi kendali kecepatan motor DC

3. HASIL DAN PEMBAHASAN

3.1 Hasil GA based PID

Hasil *tuning* kontroler PID menggunakan algoritma genetika didapatkan nilai $K_p=19.856$; $K_i=19.61$ dan $K_d=0.243$. Nilai tersebut didapat melalui proses algoritma genetika dengan jumlah generasi sebanyak 50 generasi. Komputasi algoritma genetika mengalami kondisi konvergen(memusat) bahkan berhenti pada sekitar generasi ke-45 hingga generasi ke-50, (lihat gambar 3.1). Hasil penelitian ini sejalan dengan kutipan teori yang menyatakan bahwa siklus proses evaluasi nilai *fitness* sampai dengan proses reproduksi akan berlangsung berulang kali (pada sekian generasi) sampai tidak dihasilkan perbaikan keturunan, atau sampai kriteria optimum ditemukan[23].



Gambar 3.1 Kondisi konvergen pada generasi ke 50 pada proses GA *tuning* PID

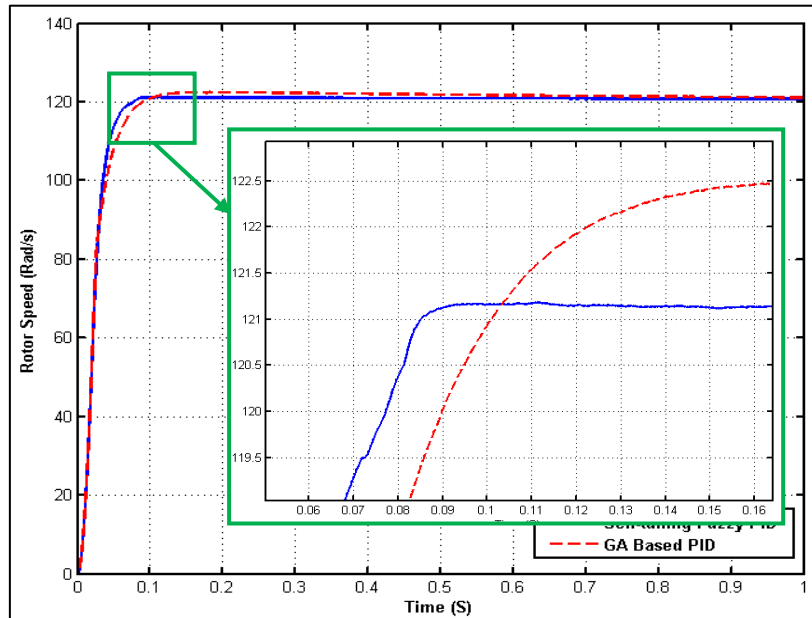
3.2 Respon Transien Sistem Pengendali Kecepatan Motor DC

3.2.1 Respon pada Case A

Hasil simulasi berupa data respon transien dari kedua kontroler ditampilkan pada tabel 3.1. Hasil simulasi berupa grafik pada Gambar 3.2. *Self-tuning* fuzzy PID memiliki nilai respon lebih cepat dan lebih dekat dengan nilai *set point* dibanding dengan GA based PID. *Self-tuning* fuzzy PID hanya membutuhkan waktu sekitar 0.052 detik untuk mencapai kecepatan *set point* (*rise time*). Nilai *overshoot*nya hanya 1%, 0.32 detik untuk mencapai *settling time* dan *steady state error* hanya 0.66%.

Tabel 3.1 Data respon transien *Self-tuningfuzzyPID* dan *GA based PID* pada *Case A*

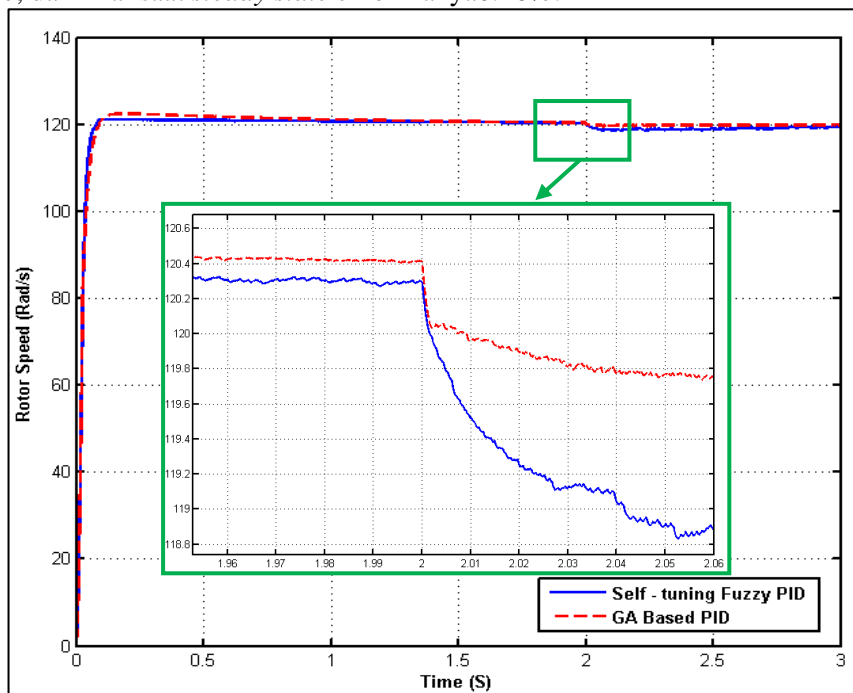
Kontroler	Overshoot	Rise time	Settling time	Steady state error
<i>Self-tuning Fuzzy PID</i>	1 %	0.052 s	0.32 s	0.66 %
<i>GA based PID</i>	2.08 %	0.074 s	0.42 s	1.08 %



Gambar 3.2 Grafik respon *Self-tuningfuzzy PID* dan *GA basedPID* pada *Case A*

3.2.2 Respon pada *Case B*

Data respon transien ditampilkan pada tabel 3.2. Saat terjadi peningkatan nilai beban torsi, nilai respon *GA based PID controller* tidak berbeda jauh dengan nilai *set point*. Nilai *undershoot* hanya mencapai 0.3%, dan nilai saat *steady state error* hanya 0.16%.



Gambar 3.3 Grafik respon *self-tuningfuzzyPID* dan *GA based PID* pada *Case B*

Tabel 3.2 Data respon transien *self-tuning fuzzy PID* dan *GA based PID* pada *Case B*

Kontroler	Undershoot	Steady state error
<i>Self-Tuning Fuzzy PID</i>	1 %	0.41 %
<i>GA based PID</i>	0.3 %	0.16

3.2.3 Respon pada *Case C*

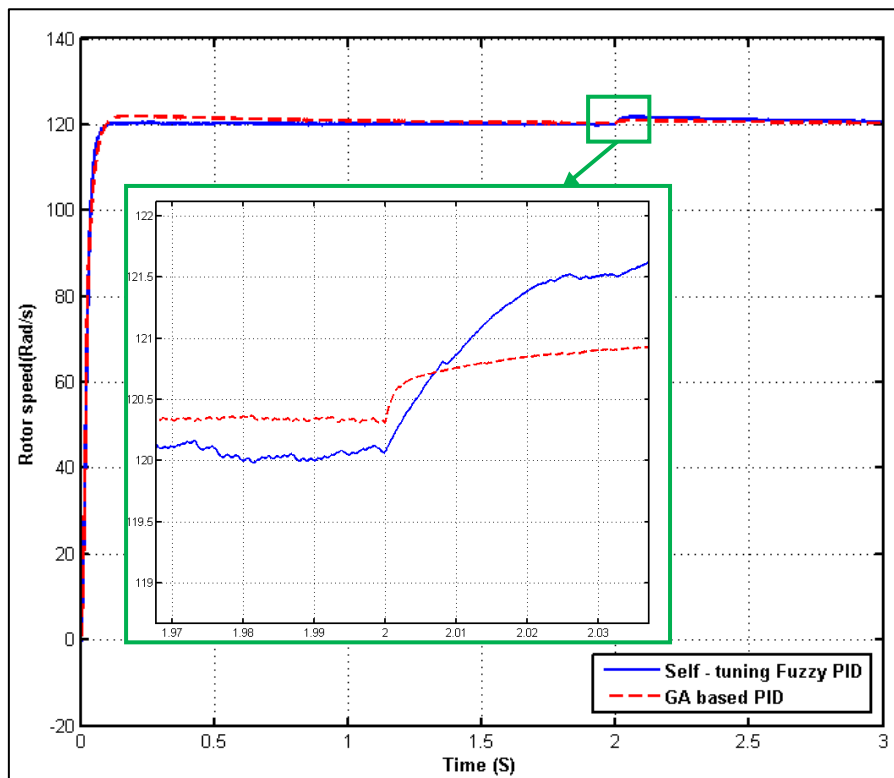
Kondisi *Case C* diaplikasikan pada simulasi selama 3 detik. Nilai respon transien ditabulasikan pada tabel 3.3, hasil simulasi berupa grafik ditampilkan pada gambar 3.4.

Undershoot yang dimiliki *GA based PID* pada *Case C* hanya 0.83% dan nilai *steady state error* 0.33%. Sedangkan nilai *undershoot self-tuning fuzzy PID* sebesar 1.41% dan nilai presentase saat *steady state error* 0.58%.

Tabel 3.3 Data respon transien *self-tuning fuzzy PID* dan *GA based PID* pada *Case C*

Kontroler	Undershoot	Steady state error
<i>Self-tuning Fuzzy PID</i>	1.41 %	0.58 %
<i>GA based PID</i>	0.83 %	0.33 %

Berdasarkan hasil simulasi berupa grafik ataupun data respon transien pada kedua *Case (Case B dan Case C)* menunjukkan bahwa respon transien *GA based PID* relatif lebih dekat dengan nilai *set point* dibanding *self-tuning fuzzy PID*. *GA based PID* mampu menghadapi perubahan beban torsi saat motor beroperasi.



Gambar 3.4 Grafik respon *self-tuning fuzzy PID* dan *GA based PID* pada *Case C*

3.2.4 Respon pada *Case D*

Pada *Case D* dan *Case E*, kecepatan diatur meningkat dan menurun. Gambar 3.5 menampilkan grafik respon simulasi pada *Case D*. Sedangkan data dalam tabel 3.4 menunjukkan kondisi respon transien pada *Case D* pada saat 0.5 detik pertama.

Berdasarkan data respon transien pada *Case D*, *self-tuning fuzzy PID* menunjukkan nilai respon yang lebih dekat dengan nilai *set point*. Nilai *overshoot* yang dimiliki *self-tuning fuzzy PID* hanya sebesar 1.08 % dan nilai *steady state error* sebesar 0.77 %.

Tabel 3.4 Data respon transien *Self-tuning* fuzzyPID dan GA basedPID pada Case D

Kontroler	Rise time	Overshoot	Steady state error
<i>Self-tuning</i> Fuzzy PID	0.51 s	1.08 %	0.77 %
GA based PID	0.53 s	1.41 %	1.07 %

3.2.5 Respon pada Case E

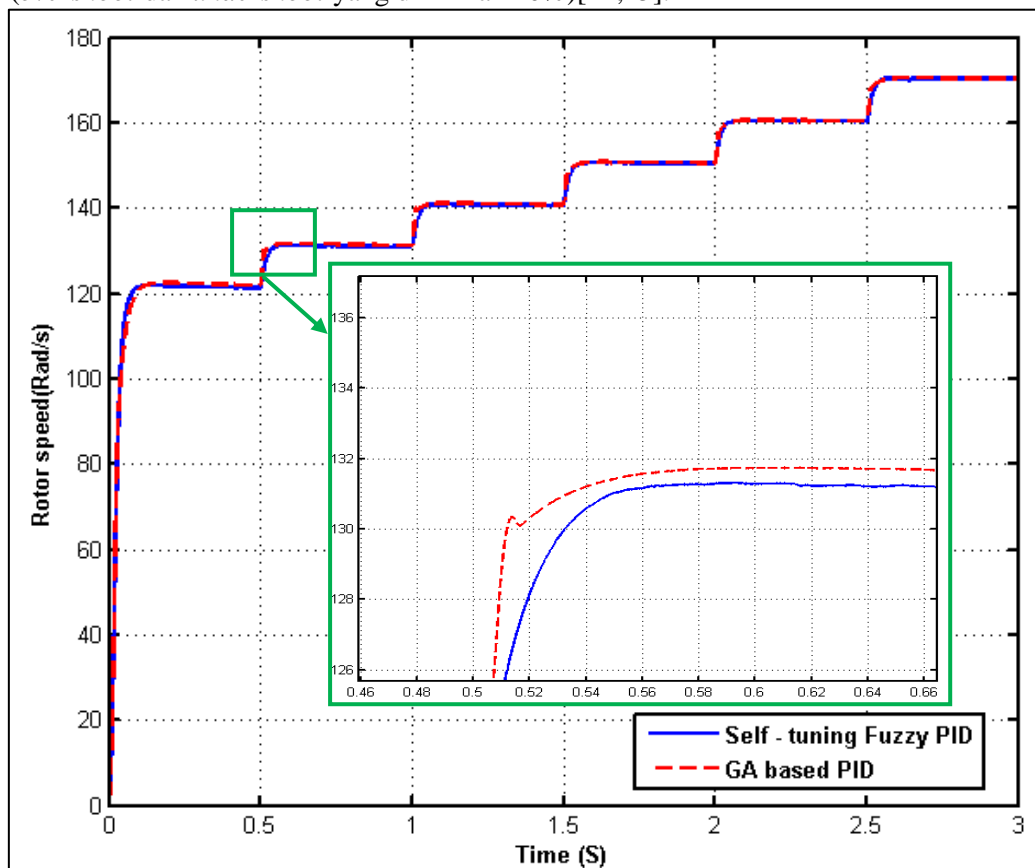
Gambar 3.6 menampilkan grafik hasil simulasi pada Case E. Data yang ditabulasikan pada tabel 3.5 adalah respon transien Case E pada waktu 2.5 detik.

Tabel 3.5 Data respon transien *Self-tuning* fuzzyPID dan GA based PID pada Case E

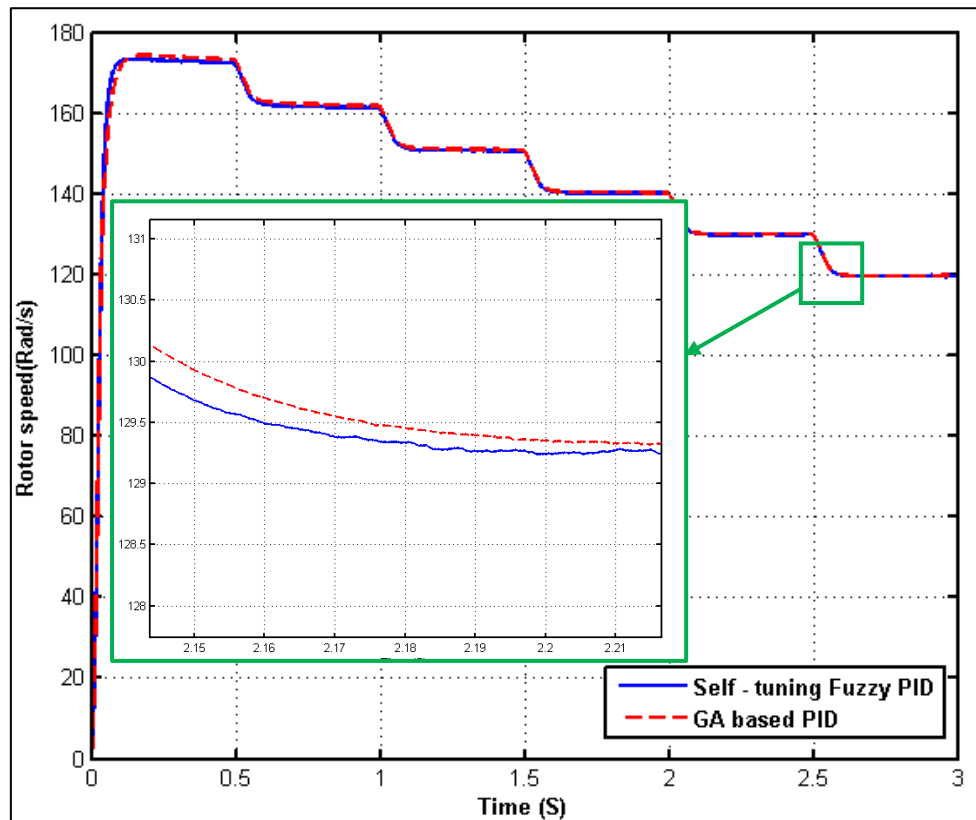
Kontroler	Rise time	Undershoot	Steady state error
<i>Self-tuning</i> Fuzzy PID	2.587 s	0.29 %	0.20 %
GA based PID	2.594 s	0.25 %	0.22 %

Hasil tabulasi menunjukkan bahwa meskipun *self-tuning* fuzzy PID memiliki nilai *undershoot* yang tinggi, waktu yang dibutuhkan untuk mencapai nilai kecepatan *set point*nya lebih cepat dan nilai *steady state error*nya lebih rendah dibanding GA based PID. *Self-tuning* fuzzy PID mampu lebih baik dalam menghadapi perubahan *setting* kecepatan meningkat atau pun menurun saat motor sedang beroperasi.

Berdasarkan Gambar 3.2 sampai dengan 3.6 terlihat bahwa presentase nilai *overshoot* maupun *undershoot* berkisar 2% sampai 5%. Data ini menunjukkan bahwa respon transien *self-tuning* fuzzy PID dan GA based PID sebagai pengendali kecepatan motor DC masih berada dalam batas ketentuan umum (*overshoot* dan *undershoot* yang diizinkan 10%)[24,25].



Gambar 3.5 Grafik respon *Self-tuning* fuzzyPID dan GA basedPID pada Case D



Gambar 3.6 Grafik respon *self-tuning* fuzzy PID dan *GA based* PID pada *Case E*

4. KESIMPULAN

a. Kesimpulan

Berdasarkan hasil pengujian dan pembahasan, implementasi logika fuzzy dan algoritma genetika untuk *tuning* parameter kontroler PID sebagai pengendali kecepatan motor DC mampu mengarahkan sistem menuju output yang diinginkan (*set point*). Selain itu, berdasarkan hasil tabulasi respon transien, presentase nilai *overshoot* maupun *undershoot* berkisar 2% sampai 5%, data ini menunjukkan bahwa respon transien kontroler PID sebagai pengendali kecepatan motor arus searah masih berada dalam batas ketentuan umum (*overshoot* dan *undershoot* yang diizinkan 10%).

b. Saran

Penelitian tentang pengembangan pengendali kontroler PID *tuning* menggunakan logika fuzzy dan algoritma genetika sebagai pengendali kecepatan motor arus searah, masih perlu ditindak lanjuti agar diperoleh efektifitas dan hasil yang lebih sempurna. Peneliti memberikan beberapa saran untuk penelitian selanjutnya sebagai berikut:

- a) Perancangan *self-tuning* fuzzy PID controller dan GA based PID controller dapat diimplementasikan ke dalam penelitian eksperimen yang melibatkan hardware motor DC.
- b) Kecepatan motor DC juga dapat dikendalikan menggunakan kontroler PID yang dituning menggunakan metode variasi lain, seperti: Adaptive Neuro Fuzzy Inference System (ANFIS), Artificial Neural Network (ANN), PSO, Ant colony algorithm dan sebagainya.

REFERENSI

- [1] R. Sharma, K.P.S. Rana, and V. Kumar,. (2014) ‘Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator’, Expert System with Applications, No.41.
- [2] Z. Bitara, S. Al Jabia and I. Khamis. (2011). ‘Modeling and Simulation of Series DC Motors in Electric Car’, The International Conference on Technologies and Materials for Renewable Energy, Environment and Sustainability, TMREES14: Energy Procedia: No. 50, Pp. 460 – 470.

- [3] A. Faizy, and Kumar. (2011). 'DC Motor Control using Chopper'. Bachelor Thesis. National Institute of Technology Rourkela: India.
- [4] A.K. Sinha, and B. K. Sathy. (2013). 'Speed Control of DC Motor Using Chopper'. Bachelor Thesis. National Institute of Technology Rourkela: India.
- [5] J.C. Babilio and S.R. Matos. (2002). 'Design of PI and PID controllers with transient performance specification', IEEE Transaction on Education, Vol.45. No.4.
- [6] K.H. Ang, G.C.Y. Chong and Y. Li. (2005) 'PID control system analysis, design, and technology', IEEE Transactions on Control Systems Technology, Vol.13. No.4.
- [7] M.M.F. Algreer, and Y.R.M. Kuraz. (2008) 'Design fuzzy self-tuning of PID controller for chopper-fed DC motor drive', Al-Rafidain Engineering, vol.16, no.2.
- [8] R. Arulmozhiyal, R. Kandiban. (2012) 'An intelligent speed controller for brushless dc motor', 7th IEEE Conference on Industrial Electronics and Applications.
- [9] R.A. Hasanjani, S.Javadi, and R.S. Nadooshan. (2014) 'DC motor speed control by self-tuning fuzzy PID algorithm', Transaction of the Institute of Measurement and Control.
- [10] J. J. Keljik. (2013). 'Electricity4: AC/DC Motor, Control and Maintenance (10th Edition)', Delmar 5 Maxwell Drive, Clifton Park, NY 12065-2919, USA.
- [11] V. Chopra, S.K. Singla, and L. Dewan. (2014) 'Comparative analysis of tuning a PID controller using intelligent methods', Acta Polytechnica Hungarica, vol.11, no.8.
- [12] M. Chebre, A. Meroufel, and Y. Bendaha. (2011) 'Speed Control of Induction Motor Using Genetic Algorithm-based PI Controller', Acta Polytechnica Hungarica. Vol. 8, No. 6.
- [13] A. T. El-Deen, A. A. H. Mahmoud, and A. R. El-Sawi. (2015) 'Optimal PID Tuning for DC Motor Speed Controller Based on Genetic Algorithm' International Review of Automatic Control (I.R.E.A.CO.). Vol. 8, N. 1.
- [14] N. P. Adhikari, M. choubey, R. Singh. . (2012) 'DC Motor Control Using Ziegler Nichols and Genetic Alogorithm Technicque', proc.International Journal of Electrical, Electronics and computer Engineering, vol.no.1, pp33-36.
- [15] R. A. Krohling and J. P. Rey. (2001) 'Design of optimal disturbance rejection PID controllers using genetic algorithms,' IEEE Transactions on Evolutionary Computation, vol. 5, no. 1, pp. 78-82.
- [16] A. Jaedun. (2011). 'Metode Penelitian Eksperimen', Pelatihan Penulisan Artikel Ilmiah, oleh LPMP Provinsi Daerah Istimewa Yogyakarta, Tanggal 20 – 23 Juni 2011.
- [17] G. Chen and T.T, Pham. (2000) 'Introduction to Fuzzy sets, Fuzzy Logic, and Fuzzy Control Systems', CRC Press: USA.
- [18] A.A. El-samahy, and M.A. Shamseldin. (2016) 'Brushless DC motor tracking control using self-tuning fuzzy PID control and model reference adaptive control', Ain Shams Engineering Journal, pp. 1 – 12.
- [19] M. Shamseldin. Speed Control of High performance Brushless DC Motor. Thesis. 2016. Helwan University:Egypt.
- [20] Gen dan cen
- [21] W. F. Mahmudy. (2015). 'Dasar – dasar Algoritma Evolusi', Modul Kuliah: Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.
- [22] S. H. Kim (2017) 'Electric Motor Control', Elsevier Science; 1 edition (May 26, 2017). chapter 2.
- [23] A. Jayachitra, and R. Vinodha. (2014) 'Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor', Advances in Artificial Intelligenc.pp. Volume 2014, Article ID 791230, 8 pages.
- [24] D. Xue, Y. Chen and D. P. Atherton. (2007) 'Linear Feedback control, Society of Industrial and Applied Mathematics, Chapter 3 and 6.
- [25] K. Ogata. (1996). 'Teknik Kontrol Automatik', PT Penerbit Erlangga, Jakarta, Indonesia.