

AUTO TUNING PID PADA LABVIEW MENGUNAKAN METODE ALGORITMA GENETIKA

Senanjung Prayoga¹, Rizky Hudhajanto², dan Mohammad Fahreza³
^{1,2,3} *Politeknik Negeri Batam, Jurusan Teknik Elektro, Batam*

E-mail: senanjung@polibatam.ac.id

Received: 13-08-2022

Accepted: 31-08-2022

Published: 31-08-2022

Abstrak

Sistem kontrol PID masih sangat banyak digunakan hingga saat ini karena kehandalan dan kemudahannya. Penentuan parameter nilai Kp, nilai Ki dan nilai Kd secara manual kerap kali membutuhkan waktu yang tidak sebentar untuk mencapai hasil kontrol yang baik dan stabil. Oleh karena itu pada penelitian ini dikembangkan sistem auto tuning PID menggunakan algoritma Genetika yang diprogram memakai *software* labview. Sistem *autotuning* menggunakan algoritma genetika yang dikembangkan ini diuji menggunakan empat fungsi transfer yang merupakan model matematis dari sistem Motor Servo pada pengujian pertama, sistem *Head tank* pada pengujian kedua, sistem Rangkain RLC pada pengujian ketiga dan pada pengujian keempat menggunakan sistem *temperature control*. Hasil kontrol PID yang dituning menggunakan sistem autotuning akan dibandingkan dengan tuning manual menggunakan metode Ziegler Nichols. Pada pengujian ini ada 3 parameter yang akan dibandingkan yaitu *overshoot*, *settling time* dan *rise time*. Hasil dari pengujian ini didapatkan rata-rata nilai pada pengujian algoritma genetika ialah *overshoot* = 6,557%, *settling time* = 1,729 detik, dan *rise time* = 0,260 detik. Sedangkan pada pengujian menggunakan metode Ziegler Nichols mendapatkan nilai *overshoot* = 27,708%, *settling time* = 2,860 detik, dan *rise time* = 0,247 detik.

Kata kunci: *Algoritma Genetika, Autotuning PID, Labview*

Abstract

PID control systems are still very widely used today because of their reliability and convenience. Determining the parameters of Kp value, Ki value and Kd value manually often takes a long time to achieve good and stable control results. Therefore, in this study, a PID auto tuning system was developed using a Genetic Algorithm programmed using Labview software. The autotuning system using the genetic algorithm developed was tested using the transfer function which is a mathematical model of the servo Motor in the first Test, System Head Tank on the second test, RLC Circuit for third Test, and the last one is Temperature control. PID control results tuned using the autotuning system will be compared with manual tuning using the Ziegler Nichols method. In this testing result of the test will be compare with 3 parameters which are overshoot, Settling time, and Rise time. The results of this test showed that the Genetic Algorithm values were overshoot = 6,557%, settling time = 1.729 Sec, rise time = 0.260 Sec. While in testing using the Ziegler Nichols method, the values of overshoot = 27,708%, Settling Time = 2.860 Sec, and rise time = 0.247 Sec.

Keywords: *Genetics Algorithm, Autotuning PID, Labview*

Pendahuluan

Penggunaan kontrol *Proportional-Integral-Derivative* (PID) sangat populer di setiap aspek otomasi industri dan kontrol proses. Karena PID mempunyai struktur yang sederhana stabilitas yang baik dan performa yang kuat dan bisa digunakan di berbagai macam kondisi [1]. Fungsi utama dari kontrol PID adalah untuk menjaga suatu sistem agar selalu pada kondisi / nilai setpoint yang telah ditetapkan serta mengurangi atau menghilangkan *steady state error*. PID memiliki parameter K_p , K_i dan K_d yang harus diatur nilainya agar sistem yang dikontrol menjadi stabil. Proses tersebut dinamakan dengan tuning. Tuning PID bisa dilakukan dengan cara manual dan juga secara otomatis (*autotuning*). Banyak macam metode tuning PID salah satunya adalah Ziegler-Nichols [2]. Hasil nilai tuning K_p , K_i dan K_d yang didapatkan dengan menggunakan metode Ziegler Nichols tidak dapat dijadikan sebagai nilai parameter akhir, melainkan hanya sebagai acuan awal yang harus dilakukan *fine tuning* secara manual agar mendapatkan nilai yang bagus. Tentunya proses ini juga membutuhkan waktu lebih lama lagi dalam mendapatkan parameter yang bagus [3].

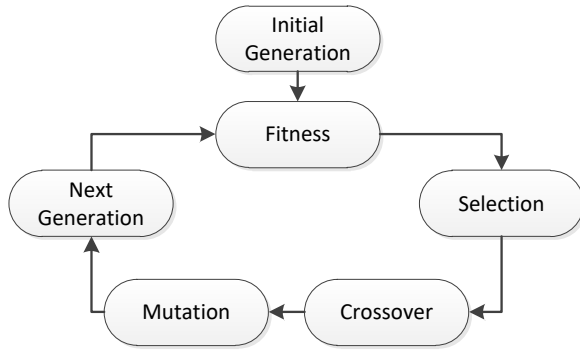
Algoritma genetika efektif untuk menemukan solusi optimal dari berbagai permasalahan. Proses algoritma genetika didasarkan pada mekanisme seleksi alam dan genetika. Algoritma genetika digunakan untuk memecahkan masalah pencarian, optimasi dan *machine learning*. Algoritma genetika dapat memecahkan masalah dengan cepat. Algoritma genetika dapat mengevaluasi banyak kemungkinan solusi dengan cepat dan itu akan meningkatkan kemungkinan untuk mencari solusi yang terbaik. Ini akan berguna khususnya ketika mencoba memecahkan masalah yang rumit [4]. Oleh karena itu pada tulisan ini akan disajikan pengembangan sistem *autotuning* PID menggunakan metode algoritma genetika. Untuk mengetahui performa sistem *autotuning* yang dikembangkan, maka akan dibandingkan dengan tuning PID menggunakan metode Ziegler-Nichols. Pengujian hasil tuning PID akan diterapkan pada empat fungsi transfer yang merupakan model matematis dari sistem Motor Servo, sistem *Head tank*, sistem Rangkaian RLC dan sistem *temperature control*.

Metode Penelitian

Genetic Algorithm atau dalam Bahasa Indonesia disebut algoritma genetika adalah metode pencarian global stokastik yang memanfaatkan fenomena alam. Perkembangan algoritma genetika berawal pada tahun 1960-an, tetapi pertumbuhannya sangat lambat. Lalu Holland mempercepat pertumbuhan di bidang tersebut pada tahun 1975-an [5].

Komputasi algoritma genetika sendiri terinspirasi dari teori evolusi Darwin yang mengatakan bahwa kelangsungan hidup makhluk hidup dipengaruhi aturan yang kuat adalah menang. Darwin juga mengatakan kelangsungan hidup dapat dipertahankan melalui proses reproduksi, kawin silang, dan mutasi. Konsep inilah yang diadaptasi menjadi algoritma komputasi untuk mencari solusi dari suatu permasalahan dengan cara yang alamiah. Solusi tersebut di lihat dalam algoritma genetika sebagai kromosom, sedangkan kumpulan kromosom disebut sebagai gen dan nilainya berupa nilai numerik, tergantung dari permasalahan yang ingin diselesaikan.

Proses kerja algoritma genetika diawali dengan inisiasi nilai random yang disebut Populasi dan setiap individu didalam populasi disebut juga dengan kromosom. Kromosom dapat direpresentasikan dengan nilai lain seperti nilai *string*, *bit*, *floating point*, dan abjad. Setelah itu kromosom-kromosom ini berkembang melalui beberapa iterasi yang disebut juga dengan generasi. Setiap generasi kromosom ini di evaluasi menggunakan fungsi *fitness*, yang merupakan permasalahan sehingga setiap individu dengan solusi terbaik yang akan terpilih. Untuk menghasilkan generasi selanjutnya ($t+1$) sebagai kromosom baru yang disebut dengan *offspring* dibentuk melalui penggabungan dua kromosom generasi saat ini (t) dengan menggunakan operator *crossover* dan memodifikasi sebuah kromosom menggunakan operator mutasi. Satu generasi baru dibentuk melalui proses seleksi sesuai dengan nilai *fitness* kromosom orangtua dan kromosom yang fit yang akan diturunkan. Kromosom dengan *fitness* terbesar memiliki probabilitas tertinggi untuk di pilih [6].



Gambar 1. Langkah-langkah Metode GA

Algoritma genetika memiliki lima proses seperti yang terlihat pada gambar 1. Langkah pertama adalah *initial generation*, yaitu menentukan populasi yang diinginkan, selanjutnya *measure fitness*, yaitu menghitung nilai fitness dari setiap populasi, setelah itu dipilih fitness yang terbaik pada langkah *selection*, lalu dari fitness pilihan tersebut dilakukan kawin silang/*crossover*. Hasil dari kawin silang tadi kemudian dilakukan proses mutasi sehingga menjadi menjadi individu baru. Proses tersebut diulang-ulang sampai mencapai kondisi yang diinginkan.

A. Inisialisasi

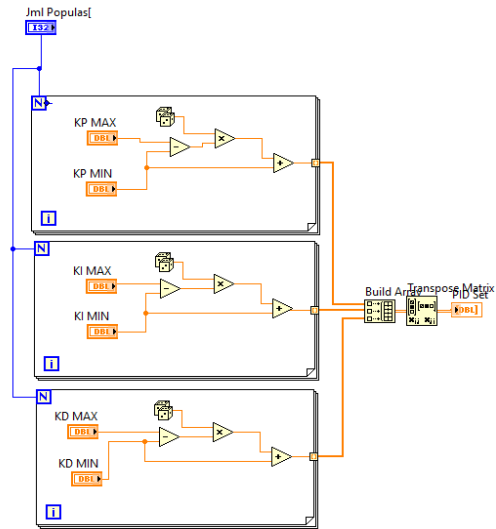
Proses inisialisasi merupakan proses yang proses pembentukan kromosom dengan cara memberikan nilai awal gen secara acak dan sesuai batasan yang telah ditentukan. Pada penelitian ini, proses inisialisasi adalah untuk menentukan nilai dari gen K_p , gen K_i dan gen K_d pada suatu sistem dengan diberi batasan pada nilai tertentu. Untuk jumlah populasi yang digunakan pada penelitian ini adalah 10 untuk memberikan individu-individu yang cukup bervariasi agar proses pemilihan individu memiliki cukup kandidat.

$$K_p = ((K_{p_{MAX}} - K_{p_{MIN}}) \times RandNum) \times K_{p_{MIN}} \quad (1)$$

$$K_i = ((K_{i_{MAX}} - K_{i_{MIN}}) \times RandNum) \times K_{i_{MIN}} \quad (2)$$

$$K_d = ((K_{d_{MAX}} - K_{d_{MIN}}) \times RandNum) \times K_{d_{MIN}} \quad (3)$$

Persamaan 1, 2 dan 3 tersebut adalah rumus perhitungan untuk mendapatkan nilai K_p , K_i dan K_d . Gambar 2 berikut ini adalah program labview dari proses inisialisasi.



Gambar 2. Program Inisialisasi

B. Evaluasi Fungsi Fitness

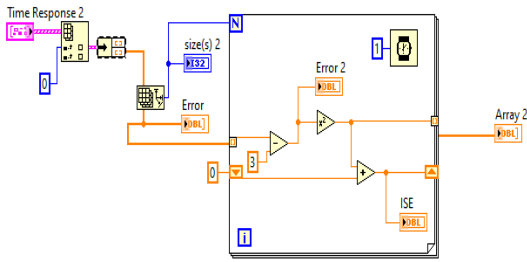
Fungsi *fitness* adalah perhitungan untuk mengevaluasi kesesuaian nilai sebuah kromosom terhadap target yang diinginkan. Berdasarkan prinsip *survival of the fittest* kromosom yang mempunyai nilai *fitness* yang tinggi mempunyai probabilitas yang tinggi untuk dipilih menjadi orangtua yang nantinya akan menghasilkan keturunan pada generasi berikutnya. Fungsi fitness yang dipakai pada penelitian ini adalah evaluasi kinerja kontrol PID yang merupakan penggabungan dari *Integral Absolute Error* (IAE), *Integral Time Absolute Error* (ITAE), dan *Integral Square Error* (ISE). Parameter ini dipilih karena mampu memberikan gambaran kinerja suatu sistem kontrol dan sifat dari ketiga rumus ini yang berlaku umum pada setiap *Plant* yang di uji [7]. Persamaan nomer 4 merupakan persamaan dari ISE, nomer 5 adalah ITAE dan nomer 6 merupakan IAE.

$$ISE = \int_0^{\infty} \{e(t)\}^2 dt \quad (4)$$

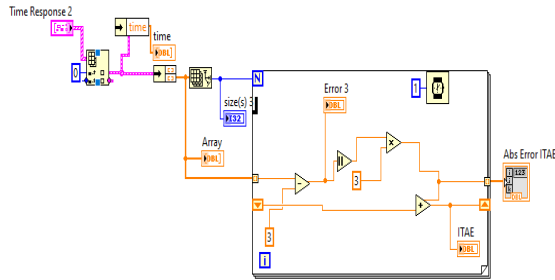
$$ITAE = \int_0^{\infty} | \{e(t)\}^2 dt | \quad (5)$$

$$IAE = \int_0^{\infty} \{e(t)\} dt \quad (6)$$

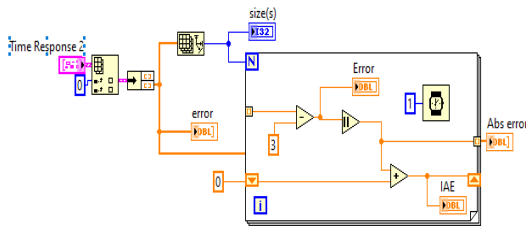
Pada gambar 3 berikut ini merupakan program dari perhitungan *Integral of Square Error*, gambar 4 adalah program perhitungan *Integral Time of Absolute Error*, sedangkan gambar 5 adalah program untuk menghitung nilai *Integral of Absolute Error*.



Gambar 3. Program Integral of Square Error



Gambar 4. Program Integral Time of Absolute Error



Gambar 5. Program Integral Absolute Error

C. Selection

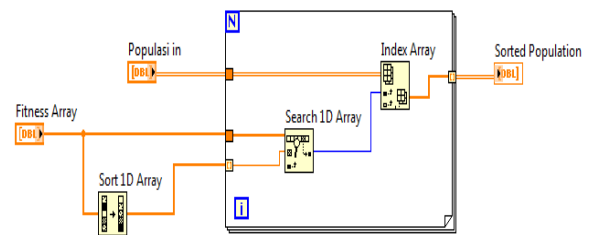
Proses ini dilakukan dengan cara membuat kromosom yang mempunyai nilai fungsi objektif kecil mempunyai kemungkinan terpilihnya besar, atau mempunyai nilai probabilitas yang besar. Fungsi objektif dari penelitian ini bisa dilihat pada persamaan 7. Sedangkan persamaan 8 merupakan fungsi fitness.

$$f_{fungsi\ objektif}[n] = ISE + ITAE + IAE \quad (7)$$

$$f_{fungsi\ fitness}[n] = \frac{1}{f_{fungsi\ objektif}[n]+1} \quad (8)$$

Setelah fungsi objektif dari masing-masing individu dihitung, tahap selanjutnya dipilih individu yang akan menjadi orangtua (*parent*) untuk iterasi selanjutnya. Pemilihan orangtua dapat menggunakan beberapa metode salah satunya adalah *retain*, yaitu dengan cara mengurutkan individu yang mempunyai nilai fitness terbaik (terkecil) dan selanjutnya diambil beberapa

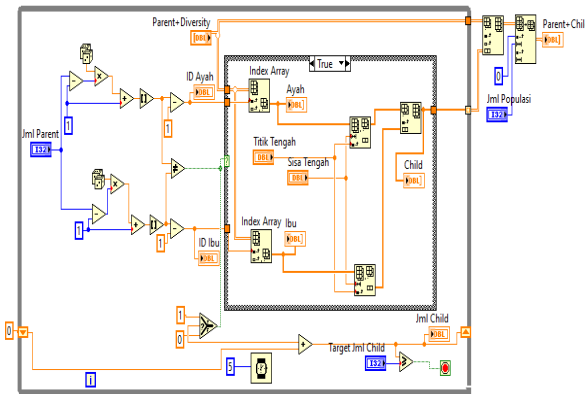
pa individu terbaik. Selanjutnya menambahkan orangtua dengan *diversity*. *Diversity* di alam arti lainnya juga keaneka-ragaman individu, yang dalam kasus ini *diversity* individu diambil dari individu sisa yang tidak terpilih sebagai orangtua. Nilai dari *diversity* yang digunakan adalah 0,2. Untuk menambahkan *diversity* langkahnya adalah membangkitkan nilai acak dengan rentang nilai dari 0 sampai dengan 1 pada setiap individu tersisa dan jika nilainya lebih dari 0,2 maka individu tersebut akan menjadi orangtua. Program selection bisa dilihat pada gambar 6.



Gambar 6. Program Selection

D. Crossover

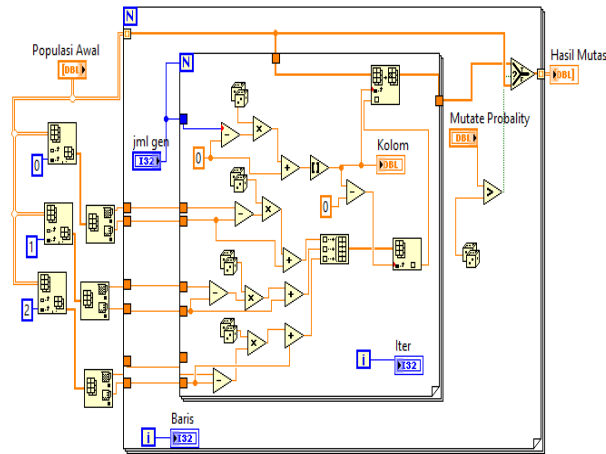
Crossover atau kawin silang adalah operator dari algoritma Genetika yang melibatkan dua induk untuk membentuk kromosom atau individu yang baru (*new population*). Prinsip dari kerja dari kawin silang ini adalah melakukan operasi (operasi aritmatika) pada gen-gen yang disesuaikan untuk mendapatkan individu yang baru (n). Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* yang telah ditentukan. *Crossover* dilaksanakan dengan memilih dua individu orangtua (individu ayah dan ibu) secara acak dan membelahnya menjadi dua bagian. Proses ini akan menghasilkan individu baru yang disebut dengan *child*/ anak, yang akan digunakan untuk iterasi selanjutnya. Kemudian kedua bagian tadi dipasangkan secara bersilang kepada kedua individu tadi. Program untuk melakukan proses *crossover* bisa di lihat pada gambar 7 berikut ini.



Gambar 7. Program Crossover

E. Mutasi

Mutasi adalah salah satu operator dari reproduksi. Mutasi berperan untuk memberikan materi genetik baru ke dalam suatu individu. Mutasi memiliki nilai probabilitas yang kecil. Pada penelitian ini *mutate probability* memiliki nilai 0,1. *random number generator* digunakan untuk menghasilkan nomer acak antara 0 sampai 1 yang akan diberikan kepada setiap individu. Jika suatu individu memiliki nilai nilai lebih kecil dari nilai acak tersebut, maka individu tadi akan dilakukan operasi mutasi. Program mutasi pada Labview bisa dilihat pada Gambar 8 berikut ini.



Gambar 8. Program Mutasi

F. Generasi Baru

Hasil dari mutasi sebelumnya akan dijadikan populasi baru dan akan digunakan untuk iterasi selanjutnya yang disebut sebagai *next generation*. Pada iterasi berikutnya, kelima proses tadi diulangi terus menerus sampai didapatkan nilai *fitness* mendekati nol atau sama dengan nol.

Metode Ziegler Nichols Tipe 2

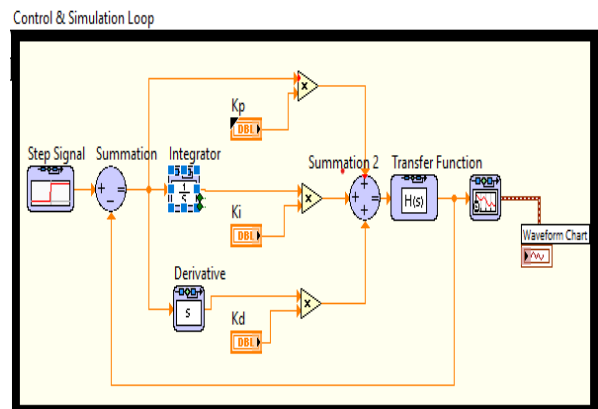
Tuning PID dengan metode Ziegler Nicholls tipe 2 ialah suatu metode tuning yang menggunakan nilai *critical gain* (K_{cr}) dan *critical period* (P_{cr}). Untuk memperoleh parameter K_p , K_i dan K_d , pada metode ini hanya menggunakan pengendali *Proporsional* (K_p) saja, sehingga untuk waktu integral T_i akan diatur hingga tak terhingga dan waktu turunan T_d diatur ke nol. Nilai K_p , dinaikkan dari 0 hingga mencapai nilai kritis K_{cr} , yaitu disaat grafik respon waktunya beresilasi secara terus menerus dengan amplitudo yang sama. Nilai kritis K_p ini disebut sebagai *ultimated gain*.

Kemudian pada saat itu diukur periodenya sehingga didapatkan nilai periode kritikal, P_{cr} . Nilai K_{cr} dan P_{cr} inilah yang digunakan sebagai dasar untuk memperoleh nilai dari parameter K_p , K_i dan K_d dengan cara memasukkan ke dalam rumus yang berada pada tabel 1 berikut ini [8].

Tabel 1. Tuning Ziegler Nichols Tipe 2

Kontrol	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$1/1.2 P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

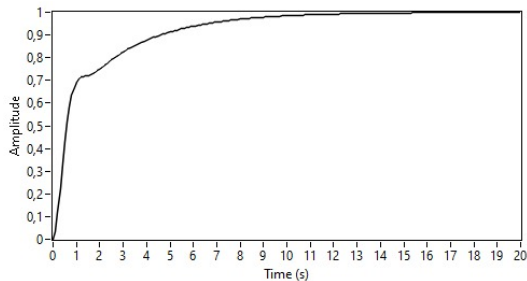
Gambar 9 berikut ini merupakan program kontrol PID dengan input nilai PID secara manual untuk menghasilkan grafik respon waktu sehingga didapatkan parameter K_{cr} dan P_{cr} .



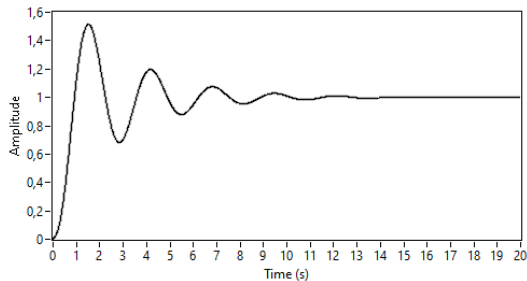
Gambar 9. Block Diagram Ziegler Nichols

Gambar 10 berikut ini adalah grafik respon waktu ketika suatu sistem mulai diberikan kontrol $K_p > 0$. Nilai K_p terus diperbesar sampai grafik mulai beresilasi seperti terlihat pada gambar 11. Pada saat itu nilai K_p belum mencapai nilai kritis. Selanjutnya nilai K_p diperbesar lagi sampai

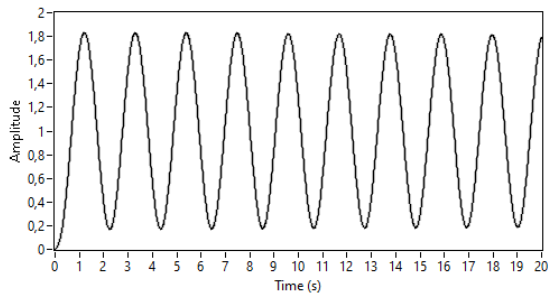
membentuk grafik yang berosilasi secara teratur dengan amplitudo yang tetap dan periode yang tetap juga seperti yang terlihat pada gambar 12. Pada saat ini gain Kp disebut sebagai Kcr dan periodenya disebut sebagai Pcr.



Gambar 10. Grafik Ketika Kp Mulai Diberikan



Gambar 11. Grafik Respon Ketika Kp < Kcr



Gambar 12. Grafik Respon Ketika Kp = Kcr

Hasil dan Diskusi

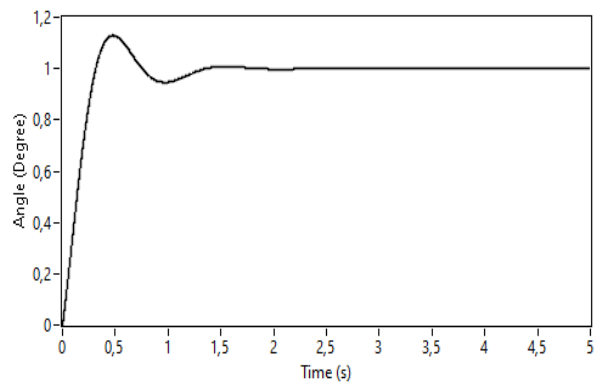
Pengujian dilakukan dengan menerapkan hasil tuning PID untuk mengontrol sistem yang diwakili dengan fungsi transfer. Fungsi transfer tersebut didapatkan dari beberapa sumber paper. Untuk semua pengujian diberikan *input set point* berupa *step response* dengan amplitudo bernilai satu. Pengujian pertama menggunakan Sistem Servo motor, pengujian kedua menggunakan Sistem *level head tank*, pengujian ketiga menggunakan sistem rangkaian RLC dan pengujian keempat menggunakan sistem *temperature control*. Sistem-sistem tersebut akan dikontrol menggunakan PID dari hasil autotuning

dengan algoritma genetika dan akan dibandingkan dengan metode Ziegler Nichols tipe 2. Parameter yang akan dibandingkan ialah *overshoot*, *settling time*, dan *rise time*.

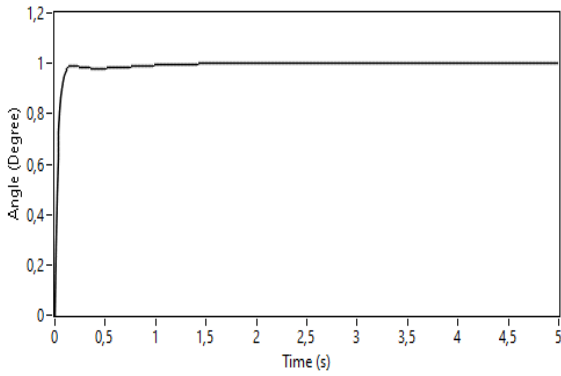
Pengujian 1: Plant Kontrol Posisi Motor Servo

$$\frac{C(s)}{R(s)} = \frac{17,7165}{s^2 + 5,6687s + 17,7165} \quad (9)$$

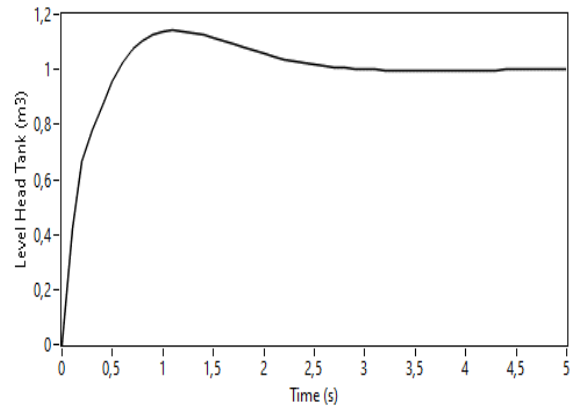
Pada pengujian pertama ini akan dilakukan kontrol posisi pada motor servo. Motor servo tersebut telah dimodelkan secara matematis menggunakan fungsi transfer seperti pada persamaan 7 tersebut [9]. Dari fungsi transfer tersebut kemudian dimasukkan ke sistem autotuning PID menggunakan algoritma genetika dan juga metode Ziegler Nichols. Untuk metode algoritma genetika didapatkan nilai Kp = 11,2498, Ki = 22,2448, Kd = 1,8288. Sedangkan tuning menggunakan metode Ziegler Nichols didapatkan nilai Kp = 12,03, Ki = 5,23, dan Kd = 20,920. Parameter Kp, Ki dan Kd tersebut digunakan untuk mengontrol fungsi transfer pada persamaan 7 tadi. Kemudian didapatkan respon waktu seperti terlihat pada gambar 13 dan 14. Untuk metode algoritma genetika didapatkan *rise time* = 0,060 detik, *overshoot* = 0,014 %, *settling time* = 0,877 detik sedangkan pada metode Ziegler Nichols didapatkan *rise time* = 0,240 detik, *overshoot* = 12,914%, *settling time* = 1,280 detik.



Gambar 13. Grafik Pengujian Metode Zn



Gambar 14. Grafik Pengujian Metode GA

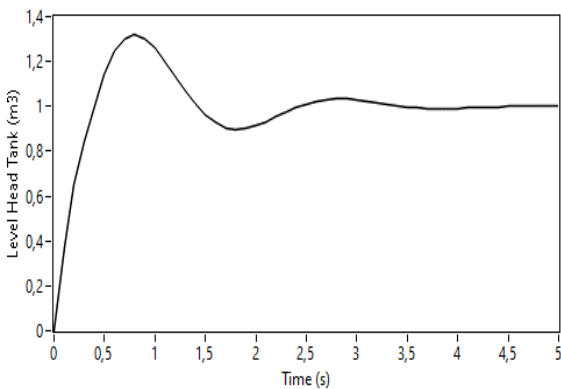


Gambar 16. Grafik Percobaan metode GA

Pengujian 2: Plant Kontrol Level Head Tank

$$\frac{C(s)}{R(s)} = \frac{11,03}{s^2 + 21,79s^2 + 4400s + 31,18} \quad (10)$$

Pada pengujian kedua ini menggunakan fungsi transfer dari suatu sistem *Head Tank* [10]. Hasil nilai parameter PID menggunakan algoritma genetika adalah $K_p = 1599,550$, $K_i = 2057,320$, $K_d = 145,985$. Sedangkan hasil tuning PID menggunakan metode Ziegler Nichols didapatkan nilai $K_p = 1168$, $K_i = 5560$, dan $K_d = 125,1$. Grafik hasil respon waktu pada pengujian menggunakan metode Tuning Ziegler Nichols bisa dilihat pada gambar 15. Sedangkan hasil pengujian menggunakan metode algoritma genetika dapat dilihat pada gambar 16. Pada metode algoritma genetika didapatkan *rise time* = 0,410 detik, *overshoot* = 14,111%, *settling time* = 2,450 detik, sedangkan pada metode Ziegler Nichols di dapatkan *rise time* = 0,300 detik, *overshoot* = 31,727%, *settling time* = 4,000 detik.

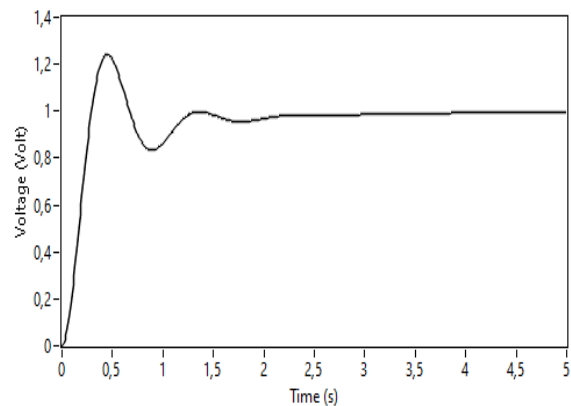


Gambar 15. Grafik Percobaan metode Zn

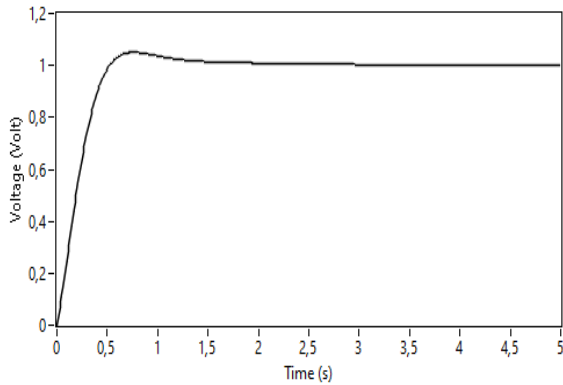
Pengujian 3: Plant Rangkaian RLC

$$\frac{C(s)}{R(s)} = \frac{1}{s^2 + 5s + 8} \quad (11)$$

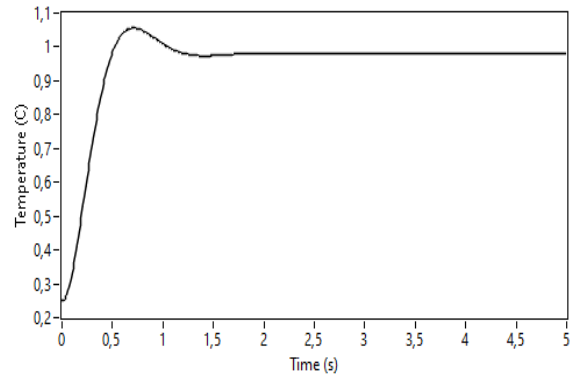
Pada pengujian ketiga ini menggunakan fungsi transfer dari sistem Rangkaian RLC [11]. Hasil nilai parameter PID menggunakan algoritma genetika didapatkan nilai $K_p = 37,1099$, $K_i = 30,9946$, $K_d = 1,9939$, sedangkan pada pengujian menggunakan metode Ziegler Nichols mendapatkan nilai $K_p = 49$, $K_i = 35,5$, dan $K_d = 0,3$. Pada metode algoritma genetika di dapatkan *rise time* = 0,310 detik, *overshoot* = 7,050%, *Settling time* = 2,470 detik sedangkan pada metode Ziegler Nichols didapatkan *rise time* = 0,300 detik, *overshoot* = 24,092%, *settling time* = 3,560 detik. Untuk grafik respon waktu dari kedua pengujian tersebut bisa dilihat pada gambar 17 dan 18 berikut ini.



Gambar 17. Grafik Percobaan metode Zn



Gambar 18. Grafik Percobaan metode GA

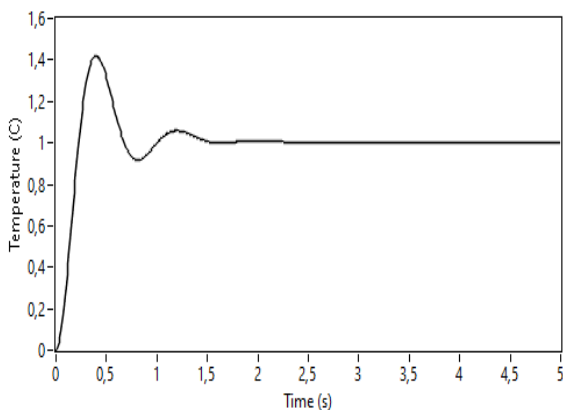


Gambar 12. Grafik Percobaan metode GA

Pengujian 4: Plant Kontrol Temperatur

$$\frac{C(s)}{R(s)} = \frac{s}{0,6s^2 + 4s + 0,4545} \quad (12)$$

Pada pengujian keempat ini menggunakan fungsi transfer dari sistem *temperatur control* [12]. Parameter PID yang dihasilkan dari metode algoritma genetika ialah $K_p = 1,200$, $K_i = 24,200$, $K_d = 0,200$, sedangkan hasil tuning PID menggunakan metode Ziegler Nichols didapatkan nilai $K_p = 4,630$, $K_i = 42,900$, dan $K_d = 0.543$. Pada metode algoritma genetika didapatkan hasil respon waktu *rise time* = 0,250 detik, *overshoot* = 5,056 %, *settling time* = 1,110 detik sedangkan pada metode Ziegler Nichols didapatkan *rise time* = 0,150 detik, *overshoot* = 42,10 %, *settling time* = 2,06 detik. Untuk grafik respon waktu dari kedua pengujian tersebut bisa dilihat pada gambar 19 dan 20 berikut ini.



Gambar 19. Grafik Percobaan metode ZN

Tabel 2. Tabel Hasil Pengujian Metode Algoritma Genetika

Pengujian	Overshoot(%)	Settling Time(sec)	Rise Time(sec)
1	0,014	0,877	0,060
2	14,111	2,450	0,410
3	7,050	2,470	0,320
4	5,056	1,110	0,250
Rata-Rata	6,557	1,729	0,260

Tabel 3. Tabel Hasil Pengujian Metode Ziegler Nichols

Pengujian	Overshoot(%)	Settling Time(sec)	Rise Time(sec)
1	12,914	1,280	0,240
2	31,727	4,000	0,300
3	24,092	3,560	0,300
4	42,100	2,600	0,150
Rata-Rata	27,708	2,860	0,247

Pada Tabel 2 dan 3 tersebut dapat dilihat bahwa dari keempat pengujian antara metode Ziegler Nichols dan algoritma genetika di atas didapatkan rata-rata dari setiap kriteria yang diuji. Rata-rata *overshoot* dari metode Ziegler Nichols sebesar 27,708% sedangkan pada metode algoritma genetika hanya sebesar 6,557%. Lalu untuk parameter *settling time*, rata-rata waktu yang dibutuhkan sistem untuk mencapai *steady state* pada metode Ziegler Nichols sebesar 2,86 detik, sedangkan pada metode algoritma geni-

ka dibutuhkan waktu lebih cepat yaitu 1,797 detik. Untuk parameter *rise time* pada metode Ziegler Nichols didapatkan rata-rata waktu selama 0,247 detik sedangkan pada metode algoritma genetika dibutuhkan waktu lebih lama, yaitu selama 0,260 detik.

Simpulan

Pada tulisan ini telah dijelaskan pengembangan metode autotuning PID menggunakan algoritma genetika. Evaluasi *fitness* yang digunakan pada algoritma genetika ini meliputi nilai IAE, ITAE, dan ISE. Hasil autotuning PID dari metode ini telah diuji untuk mengontrol empat sistem yang berbeda, yaitu sistem Sistem servo motor, sistem *level Head tank*, sistem rangkain RLC dan sistem *temperature control*. Performa kontrolnya dibandingkan dengan metode tuning Ziegler Nichols. Ada tiga parameter yang digunakan untuk perbandingan, yaitu *overshoot*, *rise time* dan *settling time*. Dari pengujian yang dilakukan didapatkan rata-rata *overshoot* dari metode Ziegler Nichols sebesar 27,708% sedangkan pada metode algoritma genetika hanya sebesar 6,557%. Lalu untuk parameter *settling time*, rata-rata waktu yang dibutuhkan sistem untuk mencapai *steady state* pada metode Ziegler Nichols sebesar 2,86 detik, sedangkan pada metode algoritma genetika dibutuhkan waktu lebih cepat yaitu 1,797 detik. Untuk parameter *rise time* pada metode Ziegler Nichols didapatkan rata-rata waktu selama 0,247 detik sedangkan pada metode algoritma genetika dibutuhkan waktu lebih lama, yaitu selama 0,260 detik. Dari penjelasan tersebut dapat disimpulkan bahwa untuk pengujian keempat fungsi transfer yang telah ditentukan, metode algoritma genetika unggul di dua parameter yaitu *overshoot* dan *settling time*.

Daftar Pustaka

- [1] Abdul Jaleel Abdul Nazir, A simplified Genetic Algorithm for online tuning of PID controller in LabView, IEEE Press, Dept of Electronics & Communication College of Engineering, Trivandrum, 2009
- [2] Ying Chen, Yong-jie Ma, Wen-xia Yun, Application of Improved Genetic Algorithm in

PID controller parameter Optimization, TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013

- [3] Gautri Mantri, N.R Kulkarani, Design and Optimization of PID Controller Using Genetic Algorithm, International Journal of research in Engineering and Technology, India, 2013
- [4] Tanasak samakwong, wudhchai Assawinchaicote, PID controller Design for Electrohydraulic Servo Valves System with Genetics Algorithm, King Mongkuts University of Technology Thonburi, Bangkok, 2016
- [5] Milan Choubey, Richa Singh, Adhikari, Dc Motor Control Using Ziegler-Nichols and Genetic Algorithm, International Journal of Electrical, Electronics and Computer Engineering, Jabalpur, 2012
- [6] K. Astrom and T. Hägglund, 1995, PID Controllers : Theory, Design and Tuning 2nd Edition, Instrument Society of America
- [7] A. Jayachitra, R.Vinodha, 2014, Genetic Algoritm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor, Departemen of Electronics and Communication Engineering, V.R.S College of engineering and Technology, Villupuram 607 107, India, Hindawi Publishing Corporation
- [8] Wijaya, Eka Candra. Auto Tuning PID Berbasis Metode Osilasi Ziegler-Nichols Menggunakan Mikrokontroler AT89S52 pada Pengendalian Suhu. Universitas Diponegoro, Semarang. 2005.
- [9] D.C Meena, Ambrish Devanshu Genetic Algorithm Tuned PID Controller For Process Control, IEEE Press, Dept of Electrical Engineering, Delhi Technology Unoversity, 2017
- [10] Nurkarima Ihdina, Priyatman Hendro, Kurniawan Budi, "Pemodelan system level air Head Tank Menggunakan Pengendali PID pada pembangkit Listrik Tenaga Mini Hidro Merasap, Universitas Tanjung Pura, Pontianak, 2014
- [11] Muhammad Ali, "Pembelajaran Perancangan Sistem Kontrol PID Dengan Software Matlab", Jurnal Edukasi Elektro vol 1 no 1, Universitas Negeri Yogyakarta, 2004
- [12] A.R Laware, V.S Bandal, D.B Talange, "Real Time Temperature Control System Using

Auto Tuning PID pada LabView Menggunakan Metode Algoritma Genetika (Senanjung Prayoga)

PID Controller and Supervisory Control and Data Acquisition System (SCADA)", Departemen of Instrumentation and Control Engineering, Ahmednagar, India, 2013