

Penalaan Parameter Pengendali PID untuk Pengendalian Kecepatan Motor Arus Searah Menggunakan Metode Algoritma Genetika dan Jaringan Syaraf Tiruan

Tuning of PID Controller Parameters for Direct Current Motor Speed Control Using Genetic Algorithms and Artificial Neural Networks Methods

Bhakti Yudho Suprpto^{1*}, Afnizar Azmi², Febby Nora³, Suci Dwijayanti⁴

^{1,2,3,4}Jurusan Teknik Elektro, Fakultas Teknik

Universitas Sriwijaya

Jl. Raya Palembang-Prabumulih, Km. 32, Inderalaya, Sumatera Selatan, 30662, Indonesia

email: ¹bhakti@ft.unsri.ac.id, ²bom.azmy@gmail.com, ³eternal_fedec@yahoo.com,

⁴sucidwijayanti@ft.unsri.ac.id

ABSTRAK

DOI;
10.30595/jrst.v4i1.5050

Histori Artikel:

Diajukan:
04/08/2019

Direvisi:
08/02/2020

Diterima:
10/03/2020

Dalam proses kendali diperlukan penalaan parameter yang baik sehingga tercapai tujuan pengendalian. Metode yang paling dipakai yaitu metode Ziegler Nichols. Namun terkadang kesulitan metode ini adalah memerlukan pengetahuan tentang model terutama pada sistem orde tinggi khususnya dalam penentuan osilasi pada sistem tersebut. Pada penelitian ini dipergunakan Jaringan Syaraf Tiruan untuk menentukan parameter peluang pindah silang (Pc) dan peluang mutasi (Pm) yang terdapat pada Algoritma Genetika untuk menentukan parameter pengendali Proportional Integral Derivative (PID). Penelitian ini mengambil objek motor arus searah. Dari penelitian ini didapatkan hasil terbaik pada populasi 100 dengan parameter PID yaitu Kp bernilai 1.0309, Ki bernilai 25.9346 dan Kd bernilai 0.0186, dimana nilai fitness terbaik, yaitu 0.22443 pada generasi ke 64, dengan nilai fitness rata-rata 11.6918. Respon sistem yang dihasilkan juga tidak memiliki overshoot, tidak memiliki *peak time*, *settling time* 0.345 detik, dan *rise time* 10-90% sebesar 0.10977 detik. Sehingga dapat dikatakan bahwa penggunaan Jaringan Syaraf Tiruan yang dikombinasikan dengan Algoritma Genetika dalam menentukan parameter pengendali PID cukup berhasil.

Kata Kunci: Algoritma Genetika, Jaringan Syarat Tiruan, Motor Arus Searah, Parameter Pengendali PID

ABSTRACT

In the process of control required good parameter tuning in order to achieve control objectives. The most used method is the Ziegler Nichols method. But sometimes the difficulty of this method is that it requires knowledge of the model, especially in high-order systems, especially in determining oscillations in the system. In this research is utilized by a Neural Network to determine crossover probability (Pc) parameter and mutation probability (Pm) of Genetic Algorithm to determine parameters the Proportional Integral Derivative (PID) controller. This Research used the direct-current motor as object. From this research is got best result at population : 100 with the PID controller parameter that is Kp : 1.0309, Ki : 25.9346 and Kd : 0.0186, where best value fitness, that is 0.22443 at 64 generation, with the value of fitness mean 11.6918. System Response is yielded no overshoot, no peak time, settling time in 0.345 second, and rise time 10-90% equal to 0.10977 second. So that can be said that a Neural Network use combined with the Genetic Algorithm in determining parameter of PID controller enough succeed.

Keywords: Genetic Algorithm, Neural Network, Direct Current Motor, Parameters of PID Controller.

1. PENDAHULUAN

Dalam kehidupan ini pasti akan menghadapi persoalan yang harus diselesaikan. Begitu pula dengan ilmu pengetahuan, juga memerlukan pemecahan terhadap persoalan yang dihadapi. Pemecahan masalah tersebut biasanya dilakukan dengan melakukan pendekatan agar dapat menyelesaikan masalah secara optimal dan efisien. Didalam permasalahan. Pendekatan tersebut sebagian besar menggunakan prinsip probabilitas. Salah satu metode yang sering digunakan oleh para ilmuwan untuk menyelesaikan berbagai permasalahan optimasi yaitu dengan Algoritma Genetika (AG). Algoritma Genetika merupakan bagian penting saat ini dalam bidang kecerdasan buatan dan riset operasi meskipun pada awalnya bukan digunakan untuk penyelesaian masalah, tetapi sebagai instrumen untuk mensimulasikan evolusi biologis dari sistem alami adaptif namun seiring perkembangan teknologi algoritma ini telah mampu menjadi solusi pemecahan masalah (García-Martínez, Rodríguez, & Lozano, 2018).

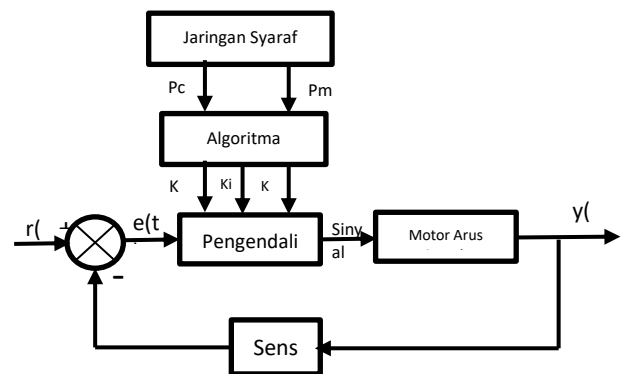
Selain algoritma genetika, metode lain pada kecerdasan buatan yang banyak digunakan yaitu jaringan syaraf tiruan. Penggunaan jaringan syaraf tiruan ini yaitu sebagai klasifikasi (Kalash et al., 2018), pemodelan (Goldberg, 2016), kendali (Heryanto, Suprijono, Suprpto, & Kusumoputro, 2017) dan lain sebagainya. Adapun kelebihan neural network ini adalah kemampuannya bekerja pada sistem yang nonlinier, meniru fungsi syaraf dan otak manusia (Suprpto, Heryanto, Suprijono, & Kusumoputro, 2017). Sehingga dengan kombinasi kedua metode ini diharapkan mampu untuk mengatasi persoalan.

Pada penelitian ini digunakan obyek yaitu motor arus searah yang akan dikendalikan kecepatannya. Pengendalian kecepatan ini biasanya menggunakan pengendali *Proportional Integral Derivative* (PID). Pengendali PID ini memiliki keunggulan yaitu sederhana, kokoh dan dapat diandalkan. Namun permasalahan yang timbul dengan penggunaan PID ini yaitu sulitnya menentukan parameter pada pengendali PID ini. Meskipun ada metode yang cukup handal dan sering dipergunakan saat penalaan parameter ini seperti metode Ziegler Nichols. Metode ini juga masih menggunakan pendekatan pada sistem yang diterapkan. Untuk mengatasi hal tersebut maka pada paper ini dibahas penggunaan algoritma genetika dan jaringan syaraf tiruan untuk penalaan parameter PID pada motor arus searah.

Tujuan dari makalah ini adalah memaparkan metode selain Ziegler Nichols untuk penalaan parameter pengendali PID untuk pengendalian kecepatan pada motor arus searah. Metode tersebut menggunakan Algoritma Genetika yang digabungkan dengan Jaringan Syaraf Tiruan. Kedua metode ini memiliki kelebihan dalam melakukan optimasi. Metode ini juga telah diterapkan untuk motor arus searah namun kombinasi dengan metode logika samar (Suprpto & Sariman, 2012).

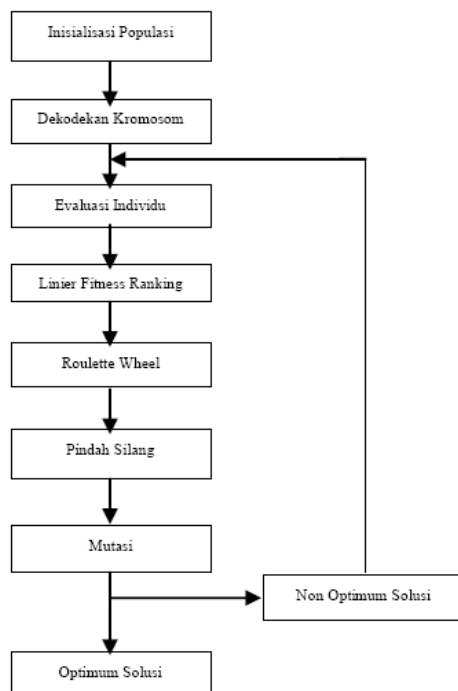
2. METODE PENELITIAN

Pada penelitian ini berdasarkan pada blok diagram seperti yang ditunjukkan pada Gambar 1 berikut ini:



Gambar 1. Blok diagram penelitian

Pada diagram blok seperti pada Gambar 1 tersebut, terlihat bahwa Algoritma Genetik ini akan dipergunakan untuk menentukan parameter K_p , K_i dan K_d pada pengendali PID dimana parameter pada Algoritma Genetika seperti peluang, pindah silang, dan peluang mutasi ditentukan menggunakan Jaringan Syaraf Tiruan (*Neural Network*). Tahapan proses yang akan dilakukan dengan menggunakan algoritma genetika dapat dilihat pada diagram alir Gambar 2 berikut ini :



Gambar 2. Tahapan proses pada Algoritma Genetika

Jika dilihat pada Gambar 2, maka langkah yang pertama dalam algoritma genetika adalah inisialisasi populasi yang bertujuan untuk membangkitkan sebuah populasi yang terdiri dari beberapa kromosom yang terdiri atas beberapa gen. Pada tahap ini, ditentukan ukuran populasi yang diinginkan (*population size*), merupakan sebuah matrik dua dimensi berukuran banyaknya kromosom dalam populasi \times banyaknya gen dalam suatu kromosom. Pada paper ini akan digunakan beberapa ukuran populasi yang berbeda, yaitu 20, 50, 70, 90, dan 100. Setelah menentukan inisialisasi populasi awal, maka masing-masing kromosom yang ada dalam populasi tersebut akan dikodekan menurut banyaknya variabel yang digunakan. Dalam hal ini terdapat tiga buah variabel, yaitu $x(1)$, $x(2)$ dan $x(3)$ yang menggantikan parameter pada PID. Kemudian dilakukan teknik pengkodean terdiri dari proses encoding dan decoding, meliputi penyandian gen dari kromosom, dimana satu gen biasanya mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk string bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator logika. Setelah itu, ditentukan nilai fitness dan fungsi aktivasi. Fungsi fitness ini terdiri dari parameter-parameter optimasi, sebelum dapat dihitung maka setiap parameter harus dihitung dalam nilai aktualnya. Nilai fitness tidak

diperbolehkan bernilai negatif (Sait & Youssef, 2000)

$$F(.) = F(P1, P2, \dots, Pn) \quad (1)$$

Sedangkan fungsi evaluasi ditentukan apa yang hendak dicapai, pada umumnya fungsi evaluasi berguna untuk memaksimalkan atau meminimalkan suatu objektif permasalahan. Persamaan fungsi evaluasi dapat ditulis dalam persamaan 2 berikut :

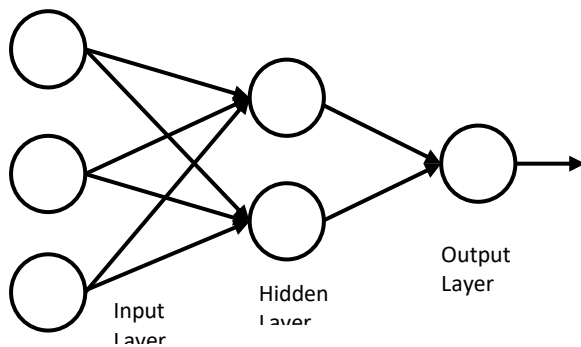
$$F(.) = \min \text{ or } \max [F(P1, P2, \dots, Pn)] \quad (2)$$

Selanjutnya sampai pada tahap seleksi. Pada tahapan ini digunakan model Seleksi *Stochastic Uniform*. Di dalam metode *Stochastic Uniform* ini, individu yang ada digambarkan pada segmen garis yang susunannya harus berurutan. Pada segmen garis tersebut akan ada sejumlah pointer dimana jumlahnya sama dengan jumlah individu yang ingin dilakukan proses seleksi pada segmen tersebut. Jika diketahui N merupakan jumlah individu yang akan dipilih sesuai dengan standar tertentu, maka jarak antar pointer pada segmen tersebut adalah $1/N$ sedangkan proses pengacakan dilakukan pada posisi *pointer* yang pertama dengan nilai *range* $[0, 1/N]$.

Tahap berikutnya adalah pindah silang (*crossover*) yang merupakan operator utama dalam menjalankan algoritma genetika dimana dalam operasinya, operator ini bekerja pada dua kromosom dalam suatu waktu. Kombinasi kedua kromosom ini akan membentuk *offspring*. Kromosom *offspring* atau sering disebut dengan kromosom anak terbentuk dan akan mewarisi sebagian sifat kromosom induknya.

Sebagai tahap akhir, proses mutasi merupakan tahapan penting karena peranannya dalam menggantikan gen yang hilang disebabkan karena proses seleksi. Proses seleksi ini akan memungkinkan gen yang tidak muncul pada saat inisialisasi populasi. Proses mutasi ini dapat dilakukan pada semua gen yang ada dengan probabilitas tertentu. Hasil akhirnya dari proses mutasi ini adalah adanya solusi yang optimum sesuai dengan standar yang telah ditetapkan. Nilai optimum yang dihasilkan oleh metode algoritma genetika ini merupakan parameter K_p , K_i dan K_d dari pengendali PID. Sedangkan masukan dari metode Algoritma Genetika ini hasil dari pelatihan yang dilakukan oleh jaringan syaraf tiruan. Jaringan Syaraf Tiruan ini merupakan salah satu metode cerdas yang saat ini paling banyak digunakan karena salah satu kelebihanannya yaitu mampu bekerja pada sistem yang memiliki sifat nonlinier tinggi serta mampu untuk meniru perilaku dari sistem otak manusia yang selalu mencoba untuk

mensimulasikan proses pembelajaran pada otak manusia tersebut. Pada paper ini, algoritma pembelajaran yang dipergunakan yaitu *Backpropagation Neural Network* (BPNN). Algoritma BPNN ini populer karena kemampuannya dalam mempelajari pemetaan multidimensional yang rumit pada sistem nonlinier dan sering disebut dengan “*beyond regression*” (Hecht-Nielsen, 1988; Rojas & Rojas, 2011). Algoritma ini menggunakan error output untuk mengubah nilai bobotnya dalam arah mundur. Untuk mendapatkan error ini, tahap perambatan maju harus dikerjakan lebih dahulu. Fungsi aktivasi yang sering digunakan adalah fungsi aktivasi sigmoid. Adapun arsitektur dari BPNN yang digunakan pada paper ini dapat dilihat pada Gambar 3 berikut :



Gambar 3. Arsitektur Algoritma BPNN

Algoritma BPNN ini memiliki proses pelatihan yang sederhana. Apabila keluaran dari sistem memberikan hasil yang salah, maka bobot yang ada pada *hidden layer* dan *input layer* akan dikoreksi berdasarkan nilai kesalahan tersebut agar galat (*error*) yang didapat semakin kecil. Secara garis besar, struktur algoritma BPNN tersebut terdiri atas lapis masukan (*input layer*) x_n , lapis tersembunyi (*hidden layer*) z_m , dan lapis keluaran (*output layer*) y_l seperti yang dapat dilihat pada Gambar 3. Sehingga persamaan pada algoritma BPNN ini dapat dilihat pada Gambar 3 sampai 6 sebagai berikut

$$z_{in_m} = v_{0m} + \sum_{n=1}^N x_n v_{nm} \quad (3)$$

$$z_m = f_1(z_{in_m}) \quad (4)$$

$$y_{in_l} = w_{0l} + \sum_{m=1}^M z_m w_{ml} \quad (5)$$

$$y_l = f_2(y_{in_l}) \quad (6)$$

Dimana z_{inm} merupakan nilai yang masuk pada hidden layer, v_{nm} adalah bobot antara input dan hidden layer, v_{0m} yaitu nilai *threshold*, z_m adalah

keluaran dari hidden layer, f_1 dan f_2 adalah fungsi aktivasi y_{in1} merupakan masukan pada *output layer*, y_1 adalah keluaran dari *output layer*. Adapun nilai galat yang dari sistem ini seperti pada persamaan 7 berikut ini :

$$\frac{1}{2} \sum_{l=1}^L (y_{tl} - y_l)^2 \quad (7)$$

Masukan dari Jaringan syaraf tiruan ini adalah besarnya populasi yang ada pada satu generasi, dan besarnya generasi maksimum yang telah ditentukan. Pada pola pelatihan (*learning*), digunakan 10 buah data yang akan digunakan sebagai *training set*. Semakin banyak data yang digunakan, maka pola pelatihan akan semakin baik. Adapun data training seperti terlihat pada Tabel 1 berikut ini

Tabel 1. Data Training

Pop	Gen	Pm	Pc
15	10	0.6	0.54
20	40	0.45	0.725
31	70	0.25	0.85
38	15	0.55	0.275
52	52	0.215	0.75
65	80	0.01	0.875
70	25	0.22	0.35
85	58	0.15	0.55
86	78	0.08	0.78
25	45	0.5	0.71

Populasi yang dijadikan data training set berkisar antara 10 sampai 60 populasi. Sedangkan maksimum generasi berkisar antara 60 sampai dengan 100 generasi. Untuk probabilitas crossover (Pc) dan probabilitas mutasi (Pm), nilai dari probabilitas mutasi bernilai lebih kecil daripada probabilitas crossover. Hal itu dikarenakan jika terlalu banyak individu yang dimutasi, akan mengakibatkan hilangnya kemiripan dengan induk sebelumnya.

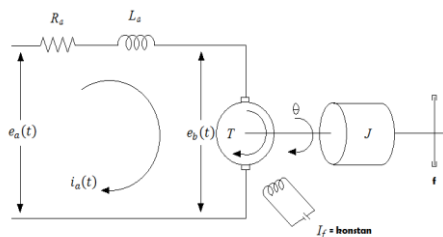
Parameter yang digunakan pada penelitian ini yaitu neuron pada input layer sebanyak 2 buah (pop, gen), Neuron pada hidden layer 3 buah, Neuron pada output layer 2 neuron (pm, pc). Nilai konstanta laju pembelajaran sebesar 0,2. Unjuk kerja dari pelatihan pada jaringan syaraf tiruan ini dilihat berdasarkan nilai Mean Square Error (MSE). Pelatihan optimumnya didapat pada epoch ke 11 dengan nilai MSE sebesar 4.79×10^{-07} . Pada epoch tersebut didapatkan nilai peluang crossover dan peluang mutasi memiliki range antara 0,3 sampai dengan 0.6. Masing-masing memiliki variasi nilai yang tidak jauh berbeda.

Keluaran dari algoritma genetika ini merupakan nilai parameter dari pengendali

PID. Pengendali PID merupakan gabungan dari pengendali proporsional (P), integral (I), dan diferensial (D). Keluaran pengendali PID akan mengubah respon mengikuti perubahan yang ada pada hasil pengukuran sensor dan *set point* yang ditentukan. Pengendali PID tersebut dapat dilihat pada persamaan 8 berikut ini

$$u(t) = Kp \cdot e(t) + Ki \int e(t)dt + Kd \frac{de(t)}{dt} \quad (8)$$

Sedangkan obyek dari penelitian ini yaitu motor arus searah yang akan dikendalikan kecepatannya. Motor arus searah memerlukan sumber tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor arus searah disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan yang berubah-ubah arah pada setiap setengah putaran. Motor Arus Searah yang digunakan pada penulisan ini adalah jenis motor DC pengontrolan jangkar Adapun prinsip kerja motor arus searah ini seperti terlihat pada Gambar 4 berikut ini



Gambar 4. Diagram skematik Motor Arus Searah

Pada motor Arus searah ini seperti pada Gambar 4, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik maupun sebaliknya berlangsung melalui medan magnet. Agar proses perubahan energi mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan. Dengan memberi arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

Torsi yang dihasilkan motor adalah berbanding lurus dengan hasil kali dari arus jangkar $i_a(t)$ dan fluks celah udara ϕ , yang berbanding lurus dengan arus medan (K. Ogata, 2002),

$$\phi = K_f i_f \quad (9)$$

dimana K_f adalah konstanta. Dengan demikian, torsi T dapat ditulis seperti pada persamaan 10

$$T = K_f i_f K_1 i_a(t) \quad (10)$$

Pada motor DC dengan pengontrolan jangkar, arus medan dijaga konstan. Untuk arus medan yang konstan, fluks menjadi konstan, dan torsi menjadi berbanding lurus dengan arus jangkar, sehingga (K. Ogata, 2002) :

$$T = K i_a(t) \quad (11)$$

dimana K adalah konstanta torsi motor. Pada saat jangkar berputar, suatu tegangan yang berbanding lurus dengan hasil kali fluks dengan kecepatan sudut, terinduksi pada jangkar. Untuk fluks yang konstan, tegangan terinduksi $e_b(t)$ berbanding lurus dengan kecepatan sudut $d\theta(t)/dt$. Dimana $\omega(t) = \frac{d\theta(t)}{dt}$, maka (K. Ogata, 2002) :

$$e_b(t) = K_b \frac{d\theta(t)}{dt} = K_b \omega(t) \quad (12)$$

dimana K_b adalah konstanta gaya gerak listrik balik. Kecepatan motor DC dengan pengontrolan jangkar diatur oleh tegangan jangkar $e_a(t)$ yang dicatu oleh suatu penguat (atau oleh generator yang dicatu oleh suatu penguat). Persamaannya :

$$L_a \frac{di_a(t)}{dt} + R_a i_a(t) + e_b(t) = e_a(t) \quad (13)$$

Arus jangkar menghasilkan torsi yang dikenakan pada inersia dan gesekan, sehingga :

$$J \frac{d\omega(t)}{dt} + f \omega(t) = T = K i_a(t) \quad (13)$$

Dengan menganggap semua syarat awal nol, dan mencari transformasi Laplace dari persamaan (11), (12), dan (13), diperoleh persamaan :

$$K_b \omega(s) = E_b(s) \quad (14)$$

$$(L_a s + R_a) I_a(s) + E_b(s) = E_a(s) \quad (15)$$

$$(Js + f) \omega(s) = T(s) = K I_a(s) \quad (16)$$

Dengan menganggap $E_a(s)$ sebagai input dan $\omega(s)$ sebagai output, diperoleh fungsi alih sistem

$$\frac{\omega(s)}{E_a(s)} = \frac{K}{(Js + f)(L_a s + R_a) + K K_b} \quad (17)$$

3. HASIL DAN PEMBAHASAN

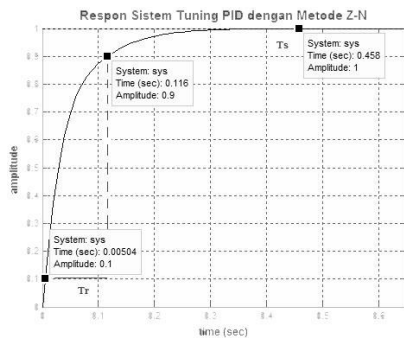
Penelitian ini dilakukan melakukan simulasi terhadap plant yang diambil yakni sebuah Motor Arus Searah terkendali Jangkar Merk U.S. Electrical Motors Type Dripproof dengan spesifikasi seperti pada Tabel 2 berikut ini (Thomas & Poongodi, 2009).

Tabel 2. Data Spesifikasi Motor Arus Searah

No	Parameter	Satuan	Nilai
1	Tegangan jangkar nominal	Volt	230
2	Kecepatan Nominal Arus jangkar	Rad/s	1500
3	nominal	Ampere	8,5

4	Daya mekanik	Hp	2
5	Resistansi jangkar	ohm	2,45
6	Induktansi jangkar	Hendry	0.035
7	Momen inersia	kgm^2/rad	0.022
8	Konstanta motor	Nm/Amp	1,2
9	Konstanta Back-up	Nm/rad	1,2
10	Koefisien gesek	$Nm/rad/s$	0.0005

Dengan data tersebut maka dilakukan pengujian untuk mencari parameter pengendali PID dengan menggunakan metode Ziegler Nichols dan didapatkan nilai Kp sebesar 1.15476, Ki sebesar 25.213, Kd sebesar 0.01322. Kemudian dilakukan pengujian dengan mengimplementasikan nilai parameter yang didapat tersebut, hasilnya dapat dilihat pada Gambar 5 berikut



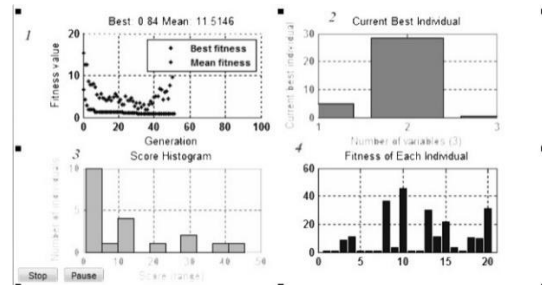
Gambar 5. Respon sistem dengan metode Ziegler Nichols

Dari Gambar 5, dapat kita lihat bahwa dengan pengendali PID yang diberikan oleh metode Ziegler-Nichols (Z-N) sistem tidak memiliki *overshot*, *settling time* sebesar 0.458 detik, dan *rise time* 0-90% sebesar 0.11096 detik. Selanjutnya dilakukan penalaan parameter dengan menggunakan algoritma genetika namun sebelumnya dengan memanfaatkan nilai dari peluang crossover dan peluang mutasi yang memiliki range antara 0,3 sampai dengan 0.6. Kemudian algoritma genetika yang dilakukan menggunakan parameter populasi sebesar 20, 70 dan 100 sedangkan dengan generasi 100 serta metode seleksi menggunakan *Stochastic Uniform*, mutasi metodenya adalah *uniform* dan *crossover* berbasis *Scattered*.

3.1 Pengujian 1

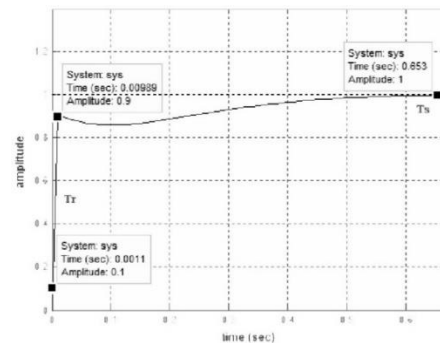
Pada pengujian ini, digunakan populasi yang berukuran 20. Hasil dari proses iterasi dapat dilihat pada Gambar 6. terdapat pada subplot 2, dimana $x_{(1)}$ yang mewakili Kp bernilai 4.7777, $x_{(2)}$ yang mewakili Ki bernilai 28.2395

dan $x_{(3)}$ yang mewakili Kd bernilai 0.4390. Sedangkan subplot 1 menunjukkan nilai fitness terbaik, yaitu 0.84 pada generasi ke 51, dengan nilai fitness rata-rata 11.5146. Artinya, indeks performansi sistem pada pengujian ini menunjukkan hasil yang tidak baik.



Gambar 6. Grafik hasil AG pada Pengujian 1

Pada subplot 3 grafik kawasan nilai fitness untuk masing-masing individu pada generasi terakhir. Terlihat bahwa nilai fitness terbanyak terdapat pada kawasan 1 sampai 5, yaitu mencapai 10 individu. Kawasan terbanyak kedua adalah 10 sampai 14 sebanyak 4 individu, sisanya masing-masing tidak lebih dari 2 individu. Selain itu, nilai fitness masing-masing individu pada generasi ke 51 ditunjukkan oleh subplot 4. Dengan menggunakan hasil Kp, Ki dan Kd diatas, menghasilkan respon seperti terlihat pada Gambar 7 berikut

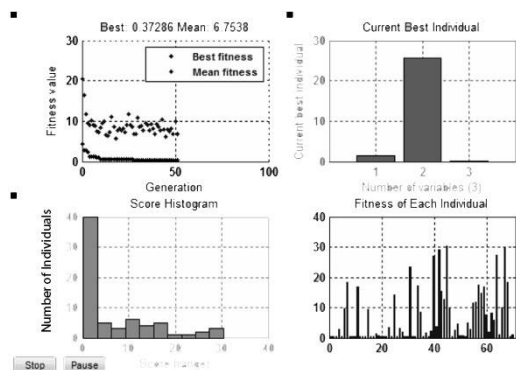


Gambar 7. Respon sistem tuning PID dengan AG pada Pengujian 1

Dari Gambar 7 diatas, terlihat bahwa dengan pengendali PID yang diberikan oleh metode Algoritma Genetika pada Pengujian 1, sistem tidak memiliki *overshot*, *settling time* sebesar 0.653 detik dan *rise time* antara 10-90% sebesar 0.00879 detik. Hasil ini tidak lebih baik daripada hasil yang diberikan oleh metode Ziegler-Nichols, dimana sistem mencapai kondisi stabil pada waktu 0.458 detik.

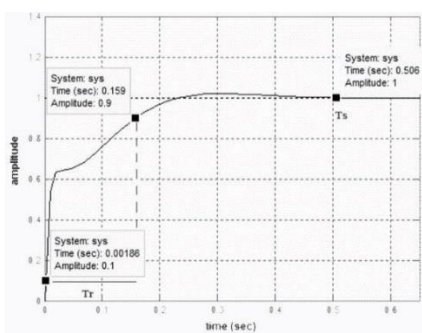
3.2 Pengujian 2

Pada pengujian 2 digunakan populasi yang berukuran 70. Hasil dari proses iterasi dapat dilihat pada Gambar 8. Masing-masing parameter PID terdapat pada subplot 2, dimana $x_{(1)}$ yang mewakili Kp bernilai 1.5587, $x_{(2)}$ yang mewakili Ki bernilai 25.7211 dan $x_{(3)}$ yang mewakili Kd bernilai 0.0706. Sedangkan subplot 1 menunjukkan nilai fitness terbaik, yaitu 0.372 pada generasi ke 51, dengan nilai fitness rata-rata 6.75. Dapat dilihat bahwa indeks performansi sistem pada pengujian ini lebih baik dari pengujian sebelumnya.



Gambar 8. Grafik hasil AG pada Pengujian 2

Pada subplot 3 grafik kawasan nilai fitness untuk masing-masing individu pada generasi terakhir. Terlihat bahwa nilai fitness terbanyak terdapat pada kawasan dibawah 3, yaitu 40 individu. Selain itu, nilai fitness masing-masing individu pada generasi ke 51 ditunjukkan oleh subplot 4. Dengan menggunakan hasil Kp, Ki dan Kd diatas, menghasilkan respon seperti pada Gambar 9 berikut



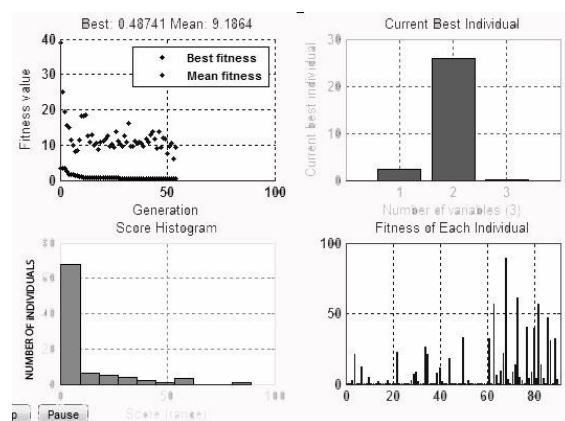
Gambar 9. Respon sistem tuning PID dengan AG pada Pengujian 2

Pada pengujian 2 ini, jumlah populasi adalah 70. Sistem juga tidak memiliki overshoot, settling time sebesar 0.506 detik, dan rise time 10-90% sebesar 0.15714 detik. Respon ini menunjukkan bahwa pengujian ini lebih baik daripada kedua pengujian sebelumnya, namun tidak metode Ziegler-Nichols yang stabil pada waktu 0.458 detik. Dari kedua pengujian ini,

belum ditemukan variasi Kp, Ki, dan Kd yang lebih baik dari hasil dengan metode Ziegler-Nichols.

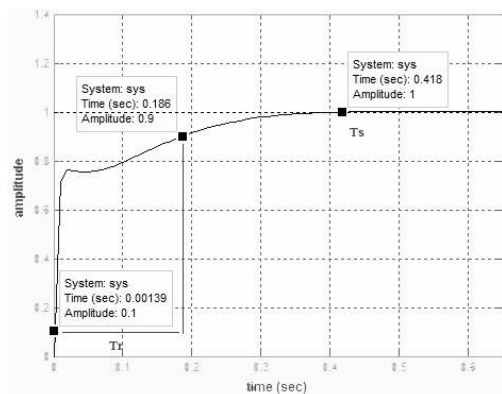
3.3 Pengujian 3

Pada pengujian ini, digunakan populasi yang berukuran 90. Hasil dari proses iterasi dapat dilihat pada Gambar 10. Masing-masing parameter PID terdapat pada subplot 2, dimana Kp bernilai 2.3178, Ki bernilai 25.9846 dan Kd bernilai 0.1253. Sedangkan subplot 1 menunjukkan nilai fitness terbaik, yaitu 0.48741 pada generasi ke 54, dengan nilai fitness rata-rata 9.1864. Pada subplot 3 grafik kawasan nilai fitness untuk masing-masing individu pada generasi terakhir. Terlihat bahwa nilai fitness terbanyak terdapat pada kawasan dibawah 10, yaitu lebih dari 60 individu, dan biasanya dibawah 3 individu. Selain itu, nilai fitness masing-masing individu pada generasi ke 54 ditunjukkan oleh subplot 4.



Gambar 10. Grafik hasil AG pada Pengujian 3

Dengan menggunakan hasil Kp, Ki dan Kd diatas, responnya terlihat seperti pada Gambar 11 berikut

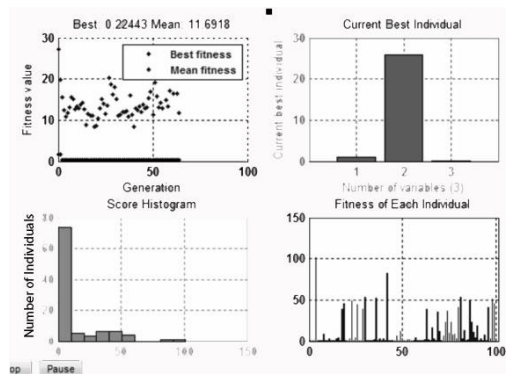


Gambar 11. Respon sistem tuning PID dengan AG pada Pengujian 3

Seperti yang terlihat pada pengujian sebelumnya untuk populasi 90, respon sistem juga tidak memiliki *overshot*, *settling time* sebesar 0.418 detik. Ini merupakan hasil yang lebih baik dari ketiga pengujian sebelumnya, termasuk hasil yang diberikan oleh metode Zieger-Nichols, dimana sistem mencapai kondisi stabil pada waktu 0.458 detik. Sedangkan *rise time* 10-90% sebesar 0.18461 detik. Meskipun menunjukkan grafik respon yang lebih baik, namun indeks performasi sistemnya tidak lebih baik daripada pengujian sebelumnya. Oleh karena itu, pada pengujian selanjutnya, jumlah populasi akan kembali diperbanyak.

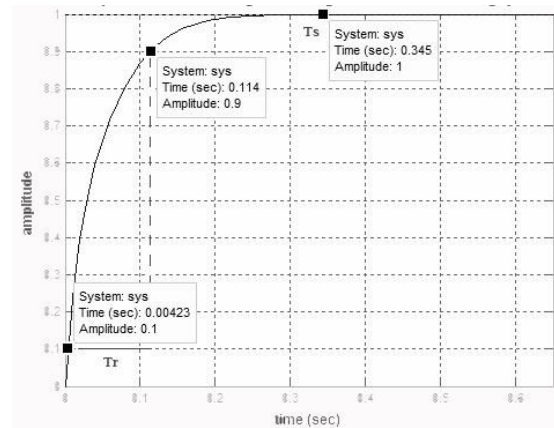
3.4 Pengujian 4

Pada pengujian ini, digunakan populasi yang berukuran 100. Hasil dari proses iterasi dapat dilihat pada Gambar 12. Masing-masing parameter PID terdapat pada subplot 2, dimana K_p bernilai 1.0309, K_i bernilai 25.9346 dan K_d bernilai 0.0186. Sedangkan subplot 1 menunjukkan nilai fitness terbaik, yaitu 0.22443 pada generasi ke 64, dengan nilai fitness rata-rata 11.6918. Dapat dilihat bahwa indeks performasi sistem pada pengujian ini merupakan indeks performasi terbaik.



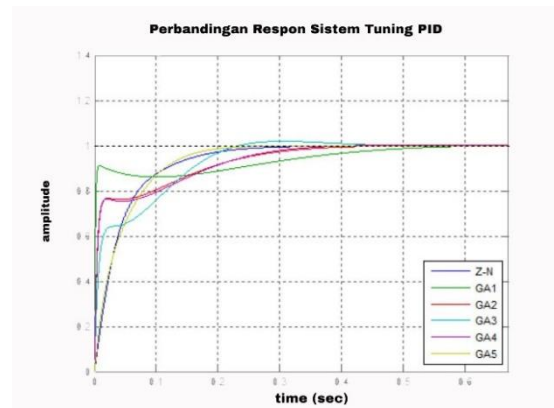
Gambar 12. Grafik hasil AG pada Pengujian 4

Pada subplot 3 grafik kawasan nilai fitness untuk masing-masing individu pada generasi terakhir. Terlihat bahwa nilai fitness terbanyak terdapat pada kawasan dibawah 10, yaitu lebih dari 70 individu. Selain itu, nilai fitness masing-masing individu pada generasi ke 64 ditunjukkan oleh subplot 4. Dengan menggunakan hasil K_p , K_i dan K_d diatas, respon yang didapatkan yaitu



Gambar 13. Respon sistem tuning PID dengan AG pada Pengujian 4

Untuk ukuran populasi 100 seperti pada Gambar 13, respon sistem yang dihasilkan juga tidak memiliki overshoot, tidak memiliki *peak time*, *settling time* 0.345 detik, dan *rise time* 10-90% sebesar 0.10977 detik. Pada pengujian terakhir ini, selain indeks performasi yang semakin baik, respon sistemnya juga merupakan yang terbaik dari keempat pengujian dengan Algoritma Genetika.



Gambar 14. Perbandingan Respon Sistem Penalaan Pengendali PID

Berdasarkan pada Gambar 14, hasil respon menunjukkan bahwa metode Algoritma Genetika dan jaringan syaraf tiruan telah dapat digunakan untuk menentukan parameter K_p , K_i , dan K_d pada pengendali PID dengan sangat baik. Hal ini memberikan pilihan lain bagi kita dalam mengoptimalkan besarnya ketiga parameter tersebut, terutama untuk sistem dengan orde tinggi. Karena pada sistem orde tinggi, jika menggunakan metode Ziegler-Nichols, kita akan menemui kesulitan dalam menentukan K_u dan K_i awal untuk menghasilkan respon sistem yang berisolasi dengan magnitudo tetap. Hal ini dapat menghabiskan waktu yang banyak, tidak demikian jika kita menggunakan metode

optimasi Algoritma Genetika. Jika dilihat pada Gambar 14 maka parameter pengendali yang didapatkan pada pengujian 4 dengan jumlah populasi 100 didapatkan hasil yang lebih baik dibandingkan metode Ziegler-Nichols. Untuk *settling time* didapatkan 0.345 detik dibandingkan metode Ziegler-Nichols sebesar 0.458 detik. *Rise time* 10-90% sebesar 0.10977 detik, ini lebih baik dibandingkan dengan metode Ziegler-Nichols sebesar 0.11096 detik. Hasil respon dapat diperbaiki dengan menambah jumlah populasi.

4. KESIMPULAN

Pada paper ini penalaan parameter pengendali PID telah dapat dilakukan dengan metode lain selain Ziegler-Nichols yaitu dengan metode Jaringan Syaraf Tiruan yang dikombinasikan dengan Algoritma Genetika. Jaringan Syaraf Tiruan dapat digunakan untuk menentukan peluang mutasi dan peluang crossover pada metode Algoritma Genetika. Berdasarkan hasil pengujian didapatkan hasil terbaik ditunjukkan pada pengujian 4 dengan ukuran populasi 100, dimana *settling time* pada waktu 0.345 detik dan *rise time* yang dicapai yaitu 0.10977 detik. Hal ini sedikit lebih baik dibandingkan dengan metode Ziegler-Nichols yakni *settling time* didapatkan 0.458 detik. *Rise time* sebesar 0.11096 detik. Penelitian yang akan dilakukan pada masa yang akan datang yaitu pemanfaatan metode ini pada beberapa aplikasi.

DAFTAR PUSTAKA

- García-Martínez, C., Rodríguez, F. J., & Lozano, M. (2018). Genetic algorithms. In *Handbook of Heuristics*. https://doi.org/10.1007/978-3-319-07124-4_28
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*. <https://doi.org/10.1613/jair.4992>
- Hecht-Nielsen, R. (1988). Theory of the backpropagation neural network. *Neural Networks*. [https://doi.org/10.1016/0893-6080\(88\)90469-8](https://doi.org/10.1016/0893-6080(88)90469-8)
- Heryanto, M. A., Suprijono, H., Suprpto, B. Y., & Kusumoputro, B. (2017). Attitude and altitude control of a quadcopter using neural network based direct inverse control scheme. *Advanced Science Letters*. <https://doi.org/10.1166/asl.2017.8328>
- K. Ogata. (2002). *Modern Control Engineering*. *Control Engineering*. <https://doi.org/10.1109/TAC.1972.1100013>
- Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D. B., Wang, Y., & Iqbal, F. (2018). Malware Classification with Deep Convolutional Neural Networks. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018 - Proceedings*. <https://doi.org/10.1109/NTMS.2018.8328749>
- Rojas, R., & Rojas, R. (2011). The Backpropagation Algorithm. In *Neural Networks*. https://doi.org/10.1007/978-3-642-61068-4_7
- Sait, S. M., & Youssef, H. (2000). Iterative computer algorithms with applications in engineering: solving combinatorial optimization problems. the IEEE Computer Society. *Los Alamitos*.
- Suprpto, B. Y., Heryanto, M. A., Suprijono, H., & Kusumoputro, B. (2017). Altitude Control of Heavy-Lift Hexacopter using Direct Inverse Control Based on Elman Recurrent Neural Network. In *Proceedings of the 8th International Conference on Computer Modeling and Simulation - ICCMS '17* (pp. 135-140). <https://doi.org/10.1145/3036331.3036354>
- Suprpto, B. Y., & Sariman, D. (2012). Metode Algoritma Genetika dengan Sistem Fuzzy Logic untuk Penentuan Parameter Pengendali PID. *Jurnal Rekayasa Elektrika*. <https://doi.org/10.17529/jre.v10i1.147>
- Thomas, N., & Poongodi, D. P. (2009). Position control of DC motor using genetic algorithm based PID controller. In *Proceedings of the World Congress on Engineering* (Vol. 2, pp. 1-3).