

Metode Algoritma Genetika dengan Sistem Fuzzy Logic untuk Penentuan Parameter Pengendali PID

Bhakti Yudho Suprpto dan Sariman

Jurusan Teknik Elektro Fakultas Teknik Universitas Sriwijaya
Jl. Raya Palembang – Prabumulih km.32 Inderalaya, Ogan Ilir
Email : bhakti_yudho@yahoo.com, sariman78@yahoo.com

Abstrak— Proportional-Integral-Derivative (PID) merupakan pengendali populer yang banyak digunakan di industri dan peralatan instrumentasi. Walaupun dalam prosedur desainnya yang sederhana, tetapi tuning parameter-parameter PID (K_p , K_i dan K_d) untuk memperoleh nilai yang optimal tersebut tidak mudah. Untuk mendapatkan parameter-parameter PID sehingga menghasilkan performa yang lebih optimal digunakan metode Algoritma Genetika dengan sistem Logika Samar (Fuzzy Logic). Algoritma Genetika merupakan metode yang digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi berdasarkan pada seleksi alam dan evolusi genetika. Sedangkan sistem fuzzy logic digunakan untuk menentukan parameter genetika seperti probabilitas crossover dan mutasi. Dalam penelitian ini, Algoritma Genetika dengan sistem Logika Samar (Fuzzy Logic) pada sistem motor arus searah dipergunakan untuk mendapatkan parameter pengendali PID yang didasarkan pada nilai mean square error (MSE). Pengujian ini menghasilkan respon sistem yang stabil dengan nilai MSE berkisar 0.0027 sampai 0.0028 dan steady state error berkisar 0.004 sampai 0.001.

Kata Kunci : PID, Algoritma Genetika, Fuzzy Logic, Motor Arus Searah, MSE

Abstract—Proportional-Integral-Derivative (PID) represents the popular controller which is frequently used in industry and instrumentation equipments. Although in simple design procedure, tuning of PID parameters (K_p , K_i And K_d) to get the optimal value is not easy and simple. Genetic Algorithm method with the Fuzzy Logic system is used to get the PID parameters with optimal performance result. Genetic Algorithm is a method that used to solve a problem of optimization based on selection and genetic evolution, while system of fuzzy logic is used to determine the genetic parameter like crossover probability and mutation. In this research, Genetic Algorithm with the Fuzzy Logic system is forming model of direct-current motor to get the parameters of PID controller based on value of mean square error (MSE). This Examination yielded the stable response system with the value MSE gyrate 0.0027 until 0.0028 and steady state error gyrate 0.004 until 0.001.

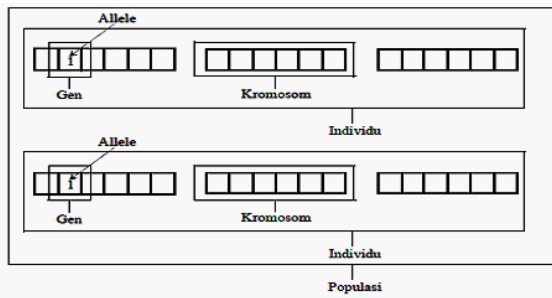
Keywords: Fuzzy Logic, Genetic Algorithm, PID, Mean Square Error (MSE)

I. PENDAHULUAN

Teori evolusi menghasilkan prinsip seleksi alam pada perkembangan generasi dalam sebuah populasi yang alami. Dimana masing-masing dari setiap individu akan diberikan kesempatan untuk mempertahankan diri dan melakukan reproduksi melalui perkawinan silang dengan individu lain dalam populasi tersebut. Kemudian akan dihasilkan individu baru (keturunan) yang membawa beberapa sifat dari induknya. Sedangkan individu dalam populasi yang tidak terseleksi dalam reproduksi akan mati dengan sendirinya[1]. Dengan proses ini, generasi dengan karakteristik yang baik akan bermunculan dalam populasi tersebut untuk kemudian ditukar dengan karakter yang lain. Dengan mengawinkan semakin banyak individu, maka akan semakin banyak kemungkinan terbaik yang dapat diperoleh.

Algoritma Genetika sebagai cabang dari Algoritma Evolusi merupakan metode yang digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah

optimasi yaitu permasalahan-permasalahan yang tak linier[1]. Dengan meniru teori evolusi, Algoritma Genetika bekerja dengan sebuah populasi dari individu-individu yang masing-masing individunya merepresentasikan sebuah solusi bagi persoalan yang ada. Individu dilambangkan dengan sebuah nilai *fitness* yang digunakan untuk mencari solusi terbaik dari suatu persoalan. Untuk penentuan parameter-parameter yang digunakan dalam Algoritma Genetika dihasilkan dengan sistem *Fuzzy Logic*. Dalam Penelitian ini, penentuan parameter pengendali merupakan hal yang penting dalam suatu sistem kendali. Salah satu pengendali yang paling populer karena keefektifannya, sederhana dalam implementasi dan luas penggunaannya adalah *Proportional-Integral-Derivative* (PID). Walaupun dalam prosedur desainnya yang sederhana, tetapi *tuning* parameter-parameter PID (K_p , K_i dan K_d) untuk memperoleh nilai yang optimal tersebut tidak mudah dan memerlukan waktu. Beberapa metode *tuning* PID konvensional yang telah dikembangkan, seperti metode Ziegler-Nichols, Cohen-Coen, dan sebagainya. Metode ini menggunakan asumsi bahwa proses yang



Gambar 1. Ilustrasi representasi struktur dalam algoritma genetika

dikontrol memiliki dinamika minimum, *linier*, dan tidak ada *noise*. Pada kenyataannya, banyak proses kontrol yang *nonlinier*, *time-variant*, dan sangat kompleks.

Berdasarkan hal tersebut diatas, maka penulis mengambil topik mengenai metode Algoritma Genetika dengan sistem *Fuzzy Logic* untuk penentuan parameter pengendali PID. Untuk Penelitian ini, model yang dipilih adalah pengendalian kecepatan motor arus searah. Hal ini dikarenakan motor arus searah memiliki torsi yang tinggi, tidak memiliki daya reaktif dan tidak menimbulkan harmonisa pada sistem tenaga listrik yang mensuplainya.

II. LANDASAN TEORI

A. Algoritma Genetika

Algoritma Genetika atau *Genetic Algorithm* (GA) sebagai cabang dari Algoritma Evolusi merupakan metode yang digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi yaitu permasalahan-permasalahan yang tak linier. Algoritma ini pertama kali diperkenalkan oleh John Holland pada tahun 1975 sebagai *whitepaper di Universitas Michigan* yang didasarkan pada proses genetik pada makhluk hidup. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika[1]. Dengan meniru teori evolusi, Algoritma Genetika bekerja dengan sebuah populasi dari individu-individu yang masing-masing individunya merepresentasikan sebuah solusi bagi persoalan yang ada. GA telah diaplikasikan untuk berbagai permasalahan seperti bioinformatika, ilmu komputer, teknik, ekonomi, matematika, fisika, dan bidang lainnya.

B. Struktur Umum Algoritma Genetika

Pada Algoritma Genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen (kromosom) disebut dengan *genotype* (gen). Dalam Algoritma Genetika gen ini bisa berupa nilai biner, *float*, *integer* maupun karakter atau kombinatorial. Sedangkan nilai dari gen tersebut disebut dengan *allele*. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi[1].

Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas dari kromosom dalam populasi

tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*) dan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

Umumnya, Algoritma Genetika memiliki lima komponen dasar menurut Michalewicz [1], yaitu :

1. Representasi genetika untuk solusi-solusi suatu masalah
2. Cara menentukan inisialisasi populasi awal
3. Penentuan fungsi evaluasi berdasarkan nilai *fitness* nya
4. Operator genetika yang mengubah gen-gen anak selama proses reproduksi

C. Representasi Genetika

Representasi mengacu pada cara solusi untuk masalah tertentu yang dikodekan menjadi struktur data yang dapat diproses oleh komputer digital. Teknik pengkodean meliputi pengkodean gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel.

Dengan demikian kromosom dapat direpresentasikan dengan menggunakan :

- *String bit* : 10011, 01101, 11110 dan sebagainya
- *Bilangan real* : 65.65 ; 45.78 ; 345.23 dan sebagainya
- *Elemen permutasi* : E2, E10, E5 dan sebagainya
- *Daftar aturan* : R1, R2, R3 dan sebagainya

D. Inisialisasi Populasi Awal

Untuk menentukan populasi awal, gen-gen yang mengisi masing-masing kromosom dibangkitkan secara *random* dengan N kromosom dimana ukuran populasi ini tergantung dari permasalahan yang ada dan sesuai dengan domain solusi.

E. Fungsi Fitness

Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai *fitness* ini yang dijadikan acuan dalam mencapai nilai optimal dalam Algoritma Genetika. Algoritma Genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi. Secara umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai tidak negatif. Apabila fungsi objektif memiliki nilai negatif, maka perlu ditambahkan konstanta C agar nilai *fitness* yang terbentuk tidak negatif.

F. Seleksi

Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses kawin silang dan mutasi. Seleksi digunakan untuk mendapatkan calon induk yang baik. Semakin tinggi nilai *fitness* suatu individu semakin besar kemungkinannya untuk terpilih.

1) Seleksi Roda Roulette (*Roulette Wheel Selection*)

Metode seleksi Roda *Roulette* sering juga dikenal dengan nama *stochastic sampling with replacement*. Individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen

individu memiliki ukuran yang sama dengan ukuran *fitness* nya. Sebuah bilangan *random* dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan *random* tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.

Metoda ini menirukan permainan *roulette wheel* di mana masing-masing kromosom menempati potongan lingkaran pada roda *raulette* secara proporsional sesuai dengan nilai *fitness*nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah

2) Stochastic Universal Sampling

Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*nya seperti halnya pada seleksi Roda *Roulette*. Kemudian diberikan sejumlah *pointer* sebanyak individu yang ingin diseleksi pada garis tersebut. Andaikan N adalah jumlah individu yang akan diseleksi, maka jarak antar *pointer* adalah $1/N$ dan posisi *pointer* pertama diberikan secara acak pada *range* $[0, 1/N]$.

Crossover adalah operator Algoritma Genetika yang utama karena beroperasi pada dua kromosom pada suatu waktu dan membentuk offspring dengan mengkombinasikan dua bentuk kromosom. Dimana kromosom offspring (anak) yang terbentuk akan mewarisi sebagian sifat kromosom induknya. Kawin silang hanya bisa dilakukan dengan suatu probabilitas tertentu, artinya kawin silang bisa dilakukan hanya jika suatu bilangan *random* yang dibangkitkan kurang dari probabilitas yang ditentukan tersebut. Pada umumnya probabilitas tersebut diset mendekati 1. Adapun beberapa operator crossover yang didasarkan pada representasi kromosom, yaitu :

1. *Single Point Crossover* (Penyilangan satu titik)
Pada crossover satu titik, posisi crossover k ($k = 1, 2, \dots, N-1$) dengan $N =$ panjang kromosom diseleksi secara *random*. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan *offspring* (anak).
2. *Multi Point Crossover* (Penyilangan banyak titik)
Pada penyilangan banyak titik, m posisi penyilangan k_i ($k = 1, 2, \dots, N-1, i = 1, 2, \dots, m$) dengan $N =$ panjang kromosom diseleksi secara *random* dan tidak diperbolehkan ada posisi yang sama, serta diurutkan naik. Variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak.
3. *Uniform/Scattered Crossover* (Penyilangan Seragam)
Pada penyilangan seragam, setiap lokasi memiliki potensi sebagai tempat penyilangan. Sebuah *mask* penyilangan dibuat sepanjang panjang kromosom secara *random* yang menunjukkan bit-bit dalam *mask* yang mana induk akan mensuplai anak dengan bit-bit yang ada. Induk yang akan menyumbang bit ke anak dipilih secara *random* dengan probabilitas yang sama.
4. *Intermediate Crossover* (Penyilangan Menengah)
Pada penyilangan menengah, hanya dapat digunakan untuk variabel *real*. Dimana nilai variabel anak (*offspring*) dipilih disekitar dan antara nilai – nilai variabel induk. Anak dihasilkan menurut aturan sebagai berikut[1]:

$$\text{Anak} = \text{induk1} + \alpha (\text{induk2} - \text{induk1}) \quad (1)$$

dimana α adalah faktor skala dari rumus berikut.

$$\alpha = \text{ratio} * (\text{bilangan } \textit{random}) \quad (2)$$

dimana *ratio* merupakan nilai dari probabilitas *crossover* (P_c). Tiap – tiap variabel anak merupakan penyilangan dari variabel induk dengan nilai α dipilih secara ulang untuk tiap variabel.

Pada *crossover*, parameter yang sangat penting yaitu peluang *crossover* (P_c). Peluang *crossover* menunjukkan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi. Metode yang paling sering digunakan pada GA sederhana dengan kromosom berbentuk *string* biner adalah *crossover* satu titik.

G. Mutasi

Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Mutasi dapat dilakukan dari semua gen yang ada dengan probabilitas mutasi tertentu. Jika bilangan *random* yang dibangkitkan kurang dari probabilitas mutasi yang ditentukan maka ubah gen tersebut menjadi nilai kebalikannya. Bilamana probabilitas mutasi adalah $(1/12)$ maka sebanyak 1 gen akan dimutasi dari kromosom yang terdiri dari 12 gen (*bits*). Pada Algoritma Genetika yang sederhana, nilai probabilitas mutasi adalah tetap selama evolusi.

Peluang mutasi (P_m) didefinisikan sebagai persentase dari jumlah total gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga akan kehilangan kemiripan dari induknya. Proses mutasi lain dapat dilakukan mutasi pada gen sebanyak probabilitas mutasi * jumlah gen, dimana posisi gen yang akan dilakukan mutasi dipilih secara acak.

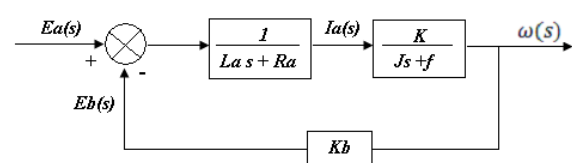
H. Motor Arus Searah

Pada penelitian ini digunakan objek untuk proses pengendalian yaitu motor arus searah. Motor arus searah ini termasuk dalam kategori jenis motor yang paling banyak digunakan baik dalam lingkungan industri dan instrumen elektronik. Motor arus searah lebih cocok digunakan pada aplikasi yang menggunakan kecepatan tinggi dan torsi yang cukup besar.

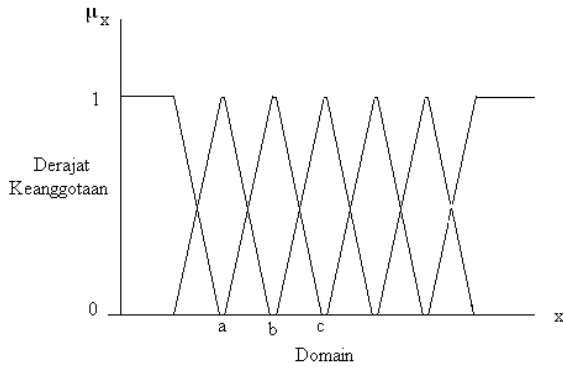
Fungsi Alih motor arus searah sebagai berikut :[2,5]

$$\frac{\omega(s)}{E_a(s)} = \frac{K}{(R_a + L_a s)(Js + f) + K K_b} \quad (3)$$

Adapun didapatkan diagram blok seperti Gambar 2.



Gambar 2. Diagram blok motor arus searah



Gambar 3 Representasi kurva fungsi keanggotaan bentuk bahu

I. Fuzzy Logic

Fuzzy Logic diperkenalkan tahun 1965 oleh Prof. Lotfi A. Zadeh dari University of California USA. Secara umum, fuzzy logic adalah metodologi berhitung dengan variabel kata-kata (*linguistik variable*), sebagai pengganti berhitung dengan bilangan[4]. Fuzzy logic merupakan logika yang memiliki nilai keaburan atau kesamaran antara benar dan salah. Perbedaan fuzzy logic dengan logika klasik dimana logika klasik biasa memiliki nilai tidak = 0 dan ya = 1 sedangkan fuzzy logic memiliki nilai antara 0 hingga 1[3].

Fuzzy logic dapat digunakan untuk memodelkan berbagai sistem serta memecahkan masalah pemetaan *non-linier*. Fuzzy logic yang berdasar pada bahasa manusia, dapat diterapkan dalam desain sistem kontrol dimana tanpa harus menghilangkan teknik desain sistem kontrol konvensional yang terlebih dahulu sudah ada.

J. Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) merupakan fungsi yang memetakan ruang input menjadi bobot atau derajat keanggotaan antara 0 dan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Dalam Tugas Akhir ini digunakan fungsi representasi kurva bentuk bahu dikarenakan diperlukan permukaan yang linier [4].

Fungsi Keanggotaan adalah :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x - a}{b - a}; & a \leq x \leq b \\ \frac{c - x}{c - b}; & b \leq x \leq c \end{cases} \quad (4)$$

K. Fuzzifikasi dan defuzzifikasi

Fuzzifikasi merupakan proses dimana nilai input crisp dibuat dalam himpunan fuzzy. Untuk mengubah input crisp menjadi input fuzzy, terlebih dahulu menentukan fungsi keanggotaan tiap input crisp, kemudian fuzzifikasi akan mengambil input crisp dan membandingkan dengan fungsi keanggotaan untuk menghasilkan input fuzzy. Jika masukan dari fuzzifikasi adalah sebuah bilangan tunggal, yaitu variabel masukan, dan keluarannya adalah derajat keanggotaan dalam suatu himpunan fuzzy dalam *antecedent*, maka masukan dan keluaran defuzzifikasi adalah kebalikannya. Jenis bilangan tunggal yang paling populer adalah *centre of area* atau *centroid* dibawah kurva dari himpunan fuzzy hasil kombinasi semua rule meenjadi himpunan fuzzy tunggal[4]. Cara menghitung bilangan ini sama dengan menghitung pusat massa dari suatu kurva tertutup. Persamaan metode titik tengah ini sebagai berikut

$$z^* = \frac{\int \mu(z)z dz}{\int \mu(z)dz} \quad (5)$$

III. METODE PENELITIAN

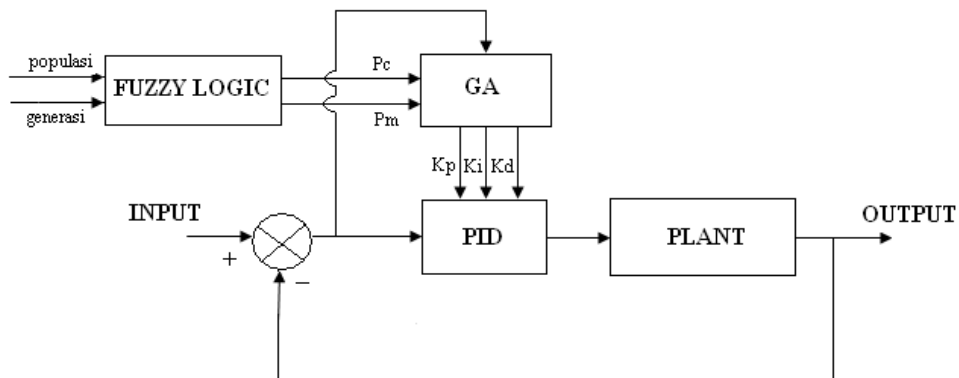
Penelitian ini dilakukan melakukan simulasi terhadap plant yang diambil yakni sebuah Motor Arus Searah terkendali Jangkar Merk U.S. Electrical Motors Type Dripproof dengan spesifikasi seperti tertulis pada Tabel 1 [2] :

Sedangkan langkah yang akan ditempuh selanjutnya adalah :

1. Pengambilan data
Data yang diambil yaitu data motor, kemudian mencari referensi tentang algoritma genetika dan fuzzy logic yang akan dipergunakan
2. Pemodelan sistem
Dari data motor yang telah ada maka akan dicari fungsi alihnya yang nantinya akan dipergunakan pada saat simulasi.
3. Perancangan
Tahap ini dilakukan dengan menyusun sistem menjadi seperti pada Gambar 4.

Pada Gambar 4 tersebut pertama kali yang harus ditentukan yaitu : input fuzzy logic.

Input didesain untuk mengambil dua masukan, yaitu populasi dan generasi yang menghasilkan dua keluaran, yaitu probabilitas *crossover* (P_c) dan probabilitas mutasi



Gambar 4 Blok diagram sistem

TABEL I
DATA MOTOR ARUS SEARAH

| No | Parameter | Simbol | Satuan | Nilai |
|-----|--------------------------|-------------|------------------------|--------|
| 1. | Tegangan jangkar nominal | $e_d(t)$ | Volt | 500 |
| 2. | Kecepatan nominal | $\omega(t)$ | Rad/s | 183,26 |
| 3. | Arus jangkar nominal | $i_a(t)$ | Amp | 12,7 |
| 4. | Daya mekanik | P | Watt | 5595 |
| 5. | Resistansi jangkar | R_a | Ω | 1,6 |
| 6. | Induktansi jangkar | L_a | H | 0,047 |
| 7. | Inersia | J | Kg.m ² /rad | 0,0282 |
| 8. | Beban (gesekan) | f | Kg.m ² /s | 0,1666 |
| 9. | Konstanta motor | K | N.m/Amp | 2,404 |
| 10. | Konstanta back-up | K_b | Volt.s/rad | 2,6175 |
| 11. | Torsi | T | N.m/rad | 30,53 |

(P_m). Nilai linguistik variabel masukan untuk populasi dibagi dalam tiga himpunan *fuzzy* : *small*, *medium* dan *large* dengan interval [0 100]. Nilai linguistik variabel masukan untuk generasi dibagi dalam tiga himpunan *fuzzy* : *short*, *medium* dan *long* dengan interval [0 300]. Nilai linguistik variabel keluaran P_c dibagi empat, yaitu : *small*, *medium*, *large* dan *very large*. Dan nilai linguistik variabel keluaran P_m dibagi empat, yaitu : *very small*, *small*, *medium* dan *large*. Untuk variabel keluaran P_c dan P_m dibuat dengan menggunakan interval [0 1]. Selain itu aturan *fuzzy* juga ditentukan yaitu pada Tabel 2.

Selanjutnya ditentukan pula pada algoritma genetika yaitu : Sebelum proses iterasi dilakukan, terlebih dahulu menginisialisasikan populasi awal secara *random* dengan *type double vector*. Ini dikarenakan untuk menghasilkan parameter K_p , K_i dan K_d dalam bilangan *real*. Dimana

untuk satu individu diwakilkan 3 gen yang meliputi parameter K_p , K_i dan K_d .

- **PopulationSize** : Penentuan ukuran populasi pada umumnya berkisar antara 20 sampai 100 individu. Ini ditentukan secara acak. Pada penelitian ini, digunakan ukuran populasi 20, dan 100.
- **Variable Bounds** : Variabel batasan dari parameter K_p , K_i dan K_d yang digunakan pada proses evaluasi GA. Dalam penelitian ini, tidak didapatkan suatu batasan variabel sehingga ditulis dalam bentuk [].
- **Constraint Function** : Dalam penelitian ini tidak digunakan suatu fungsi pembatas, sehingga ditulis dalam bentuk [].
- **Termination Function** : Didasarkan pada jumlah maksimum generasi. Proses iterasi akan berhenti ketika didapatkan nilai minimum dari fungsi objektif atau telah mencapai maksimum generasi. Untuk proses simulasi digunakan ukuran generasi 100 dan 300.

4. Pengujian sistem

Pada langkah ini akan dilakukan pengujian yang berupa pencarian parameter terbaik dari parameter pengendali sehingga respon yang didapatkan lebih stabil dan cepat.

TABEL II
ATURAN UNTUK NILAI PROBABILITAS CROSSOVER

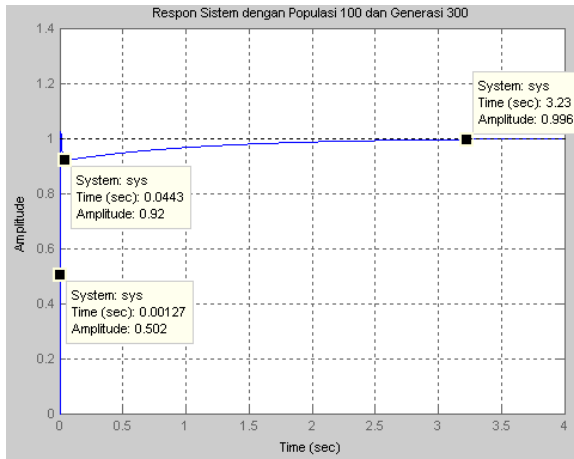
| P_c | Populasi | | |
|----------|------------|----------|------------|
| Generasi | Small | Generasi | Small |
| Short | Medium | Short | Medium |
| Medium | Large | Medium | Large |
| Long | Very large | Long | Very large |

TABEL III
ATURAN UNTUK NILAI PROBABILITAS MUTASI

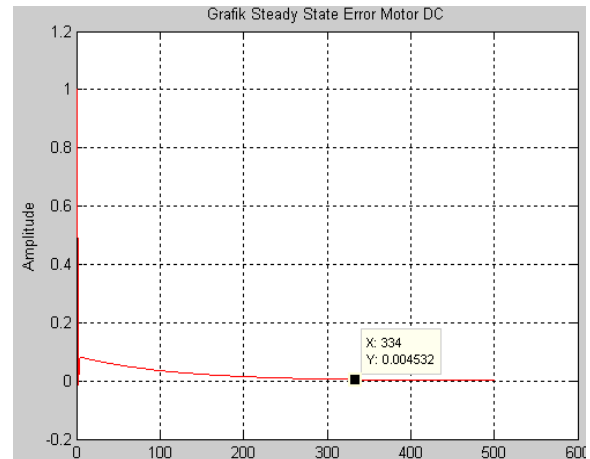
| P_m | Populasi | | |
|----------|----------|----------|--------|
| Generasi | Small | Generasi | Small |
| Short | Large | Short | Large |
| Medium | Medium | Medium | Medium |
| Long | Small | Long | Small |

IV. HASIL DAN PEMBAHASAN

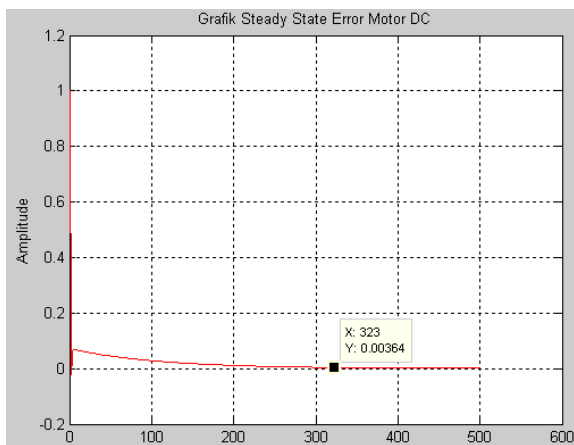
Pada penelitian ini parameter yang dipilih merupakan parameter yang memiliki respon cepat dan steady error cepat dan time delay diambil saat respon bergerak hingga ke 50% dari set point yang ditentukan, jadi penulis hanya menampilkan 4 model yaitu : populasi 100, generasi 300 dengan metode *Roulette*, populasi 100, generasi 300 dengan metode *Stochastic Uniform* dan populasi 20 generasi 100 dengan metode *Roulette* serta populasi 20 generasi 100 dengan metode *Stochastic Uniform*. Pada model ke 1 digunakan ukuran populasi 100, generasi 300 yang menghasilkan P_c sebesar 0.65 dan P_m sebesar 0.124 dan menggunakan metode seleksi *Roulette* yang menggunakan populasi yang sama pada model ke-1. Bentuk respon sistem model ke-64 dapat dilihat pada Gambar 5.



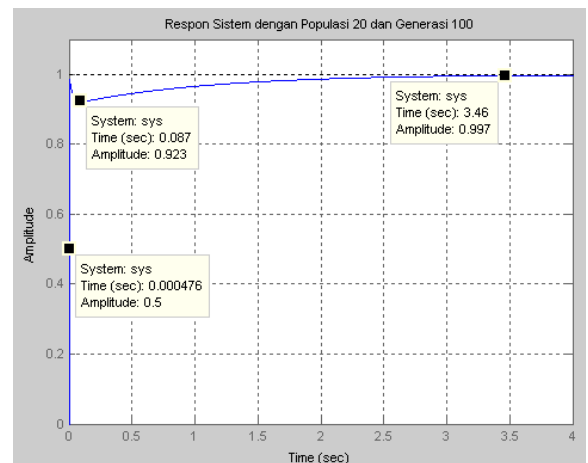
Gambar 5. Respon *closed-loop* sistem motor arus searah dengan $K_p = 29.9593$, $K_i = 29.5056$ dan $K_d = 0.293$



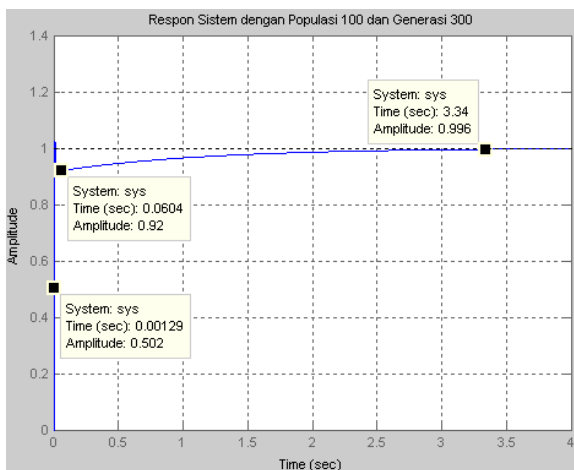
Gambar 8. Grafik *steady state error* motor arus searah model ke-2



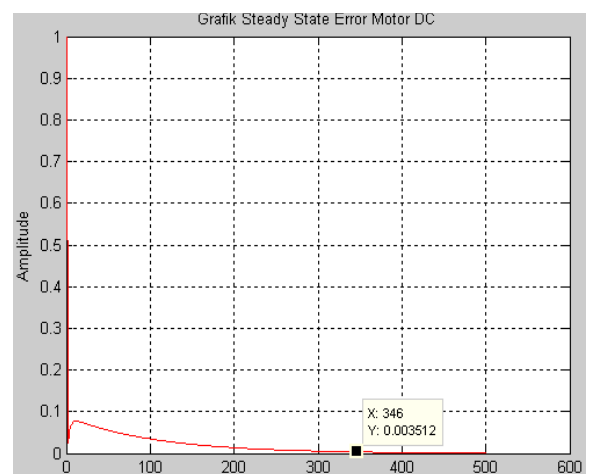
Gambar 6. Grafik *steady state error* motor arus searah model ke -1



Gambar 9. Respon *closed-loop* sistem motor arus searah dengan $K_p = 29.8898$, $K_i = 29.6209$ dan $K_d = 0.8872$



Gambar 7. Respon *closed-loop* sistem motor arus searah dengan $K_p = 29.833$, $K_i = 28.3834$ dan $K_d = 0.2871$

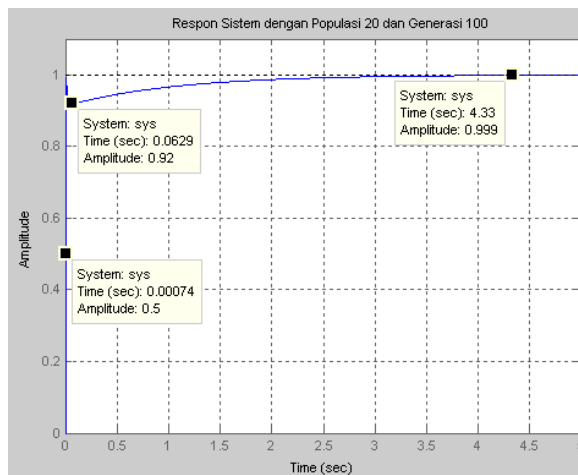


Gambar 10. Grafik *steady state error* motor arus searah model ke-3

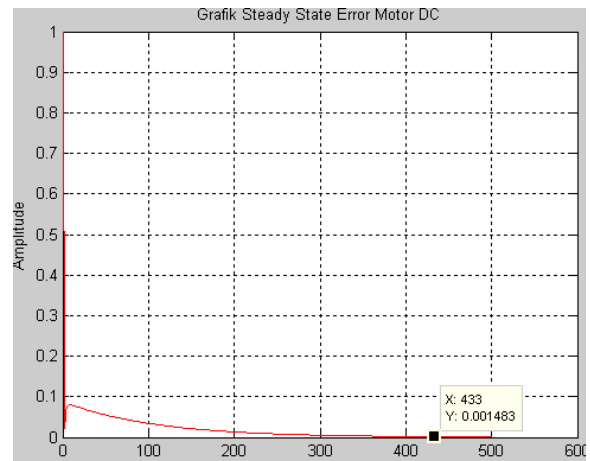
Pada model ini terlihat bahwa sistem mencapai kestabilan pada 3.23 sec dengan nilai *steady state error* mencapai 0.004 dan besar MSE nya adalah 0.0027. Terlihat bahwa t_d mencapai 0.00127 sec dan t_r mencapai 3.23 sec dengan *amplitude* 0.996. Grafik dari *steady state error* terhadap waktu ditampilkan dalam Gambar 6. Model ke 2 yaitu digunakan ukuran populasi 100, generasi 300 dengan

metode *Stochastic Uniform* menghasilkan P_c sebesar 0.65 dan P_m sebesar 0.124. Bentuk respon sistem dapat dilihat pada Gambar 7.

Pada model ini terlihat bahwa sistem mencapai kestabilan pada 3.34 sec dengan nilai *steady state error* mencapai 0.004 dan besar MSE nya adalah 0.0028. Terlihat



Gambar 11. Respon *closed-loop* sistem motor arus searah dengan $K_p = 29.2707$, $K_i = 29.7097$ dan $K_d = 0.5368$



Gambar 12. Grafik *steady state error* motor arus searah model ke-4

bahwa t_d mencapai 0.00129 sec dan t_r mencapai 3.34 sec dengan *amplitude* 0.996. Sedangkan untuk grafik respon *steady state* dapat dilihat pada Gambar 8.

Pada model ke 3 digunakan ukuran populasi 20, generasi 100 yang menghasilkan P_c sebesar 0.503 dan P_m sebesar 0.701 dan menggunakan metode seleksi *Stochastic Uniform*. Dimana proses simulasi respon sistem motor arus searah diujikan dalam diagram blok sistem dari *simulink* dan menggunakan *editor window* yang menampilkan melalui *figure window*. Bentuk respon sistem dapat dilihat pada Gambar 7 dan Gambar 8.

Pada model ini terlihat bahwa sistem mencapai kestabilan pada 3.46 sec dengan nilai *steady state error* mencapai 0.003 dan besar MSE nya adalah 0.0028. Terlihat bahwa t_d mencapai 0.000476 sec dan t_r mencapai 3.46 sec dengan *amplitude* 0.997. Sehingga kecepatan motor arus searah pada model ini, saat *start* sangat cepat dengan waktu 3.46 sec mencapai kestabilan. Pada $t = 1000$ sec, *steady state error* mencapai 0.003. Grafik dari *steady state error* terhadap waktu ditampilkan dalam Gambar 8.

Sedangkan model ke 4 digunakan ukuran populasi 20, generasi 100 yang menghasilkan P_c sebesar 0.503 dan P_m sebesar 0.701 dan menggunakan metode seleksi *Roulette* dimana grafik responnya dapat dilihat pada Gambar 11.

Pada model ini terlihat bahwa sistem mencapai kestabilan pada 4.33 sec dengan nilai *steady state error* mencapai 0.001 dan besar MSE nya adalah 0.0028. Terlihat bahwa t_d mencapai 0.00074 sec dan t_r mencapai 4.33 sec dengan *amplitude* 0.999. Sedangkan grafik *steady state* dapat dilihat pada Gambar 12.

Dari hasil pengujian diatas dengan menggunakan Algoritma Genetika dengan sistem *Fuzzy Logic* didapatkan respon sistem motor arus searah yang stabil dengan mempercepat dan meminimumkan nilai *steady state error* nya. Hasil simulasi parameter pengendali PID dengan menggunakan Algoritma Genetika ini, baik digunakan pada saat *start* motor arus searah, kemudian dengan menggunakan metode Algoritma Genetika dengan sistem *Fuzzy Logic* didapatkan besar *amplitude* mendekati nilai 1 yang berkisar antara 0.996 sampai 0.999 sehingga respon sistem motor arus searah dalam keadaan stabil dan didapatkan pula waktu yang cepat untuk mencapai *amplitude* tersebut yaitu pada 3,23 sec sampai 4,33 sec,

maka dapat dikatakan bahwa metode ini mampu menghasilkan kestabilan sistem yang baik. Karena hanya dengan meminimumkan fungsi *fitness* (fungsi objektif) yaitu MSE yang berkisar 0.0027 sampai 0.0028, dan mengatur parameter yang disesuaikan dengan sistem *Fuzzy Logic* sudah didapatkan respon sistem yang stabil dan cepat meminimumkan *steady state error* sistem mencapai 0.004 sampai 0.001. Oleh karena itu, metode ini menghasilkan kecepatan motor arus searah yang konstan karena metode Algoritma Genetika merupakan metode yang bersifat adaptif yang dapat memodelkan sistem sesuai dengan pengendali PID.

V. KESIMPULAN

Berdasarkan hasil pengujian dan analisa, dapat disimpulkan sebagai berikut :

1. Metode Algoritma Genetika dengan sistem *Fuzzy Logic* pada sistem motor arus searah menghasilkan nilai *steady state error* mendekati 0 berkisar 0.004 sampai 0.001.
2. Proses seleksi dengan metode *Roulette* dan metode *Stochastic Uniform* dapat digunakan dengan baik karena menghasilkan perbedaan waktu (T_r dan T_d) yang kecil.
3. Semakin besar ukuran populasi dan generasi maka sistem tersebut akan lebih cepat menuju kestabilan.

DAFTAR PUSTAKA

- [1] Gen, Mitsuo & Cheng, Runwei, *Genetic Algorithms and Engineering Optimization*. Ashikaga, Japan: Institute of Technology, 2000.
- [2] Ibrahim, Mohamed S., *The PID Controller Design using Genetic Algorithm*. Research Project. University of Southern Queensland, 2005.
- [3] Joelianto, Endra & Tansri, Oberli, Sistem Kontrol Hibrida Sinyal Rujukan Berbasis Logika Fuzzy untuk Peningkatan Kinerja Transien Pengontrol PID. Bandung : ITB, 2007.
- [4] Kusumadewi, Sri, *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu, 2003.
- [5] Ogata, Katsuhiko. *Modern Control Engineering Third Edition*. New Jersey: Universitas of Minnesota, 1997.
- [6] Zebulum, Ricardo S., dkk., *Evolutionary Electronics : Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*. Florida: CRC Press LLC, 2002.