

SIMULASI *TUNING* PENGENDALI PID DENGAN ALGORITMA  
GENETIKA PADA SISTEM KENDALI KECEPATAN MOTOR DC



TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan  
Program Strata Satu Departemen Teknik Elektro Fakultas Teknik  
Universitas Hasanuddin  
Makassar*

Oleh

Muhammad Alif Fikri

D041 17 1002

DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
MAKASSAR

2024



**SIMULASI *TUNING* PENGENDALI PID DENGAN ALGORITMA  
GENETIKA PADA SISTEM KENDALI KECEPATAN MOTOR DC**



**TUGAS AKHIR**

*Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan  
Program Strata Satu Departemen Teknik Elektro Fakultas Teknik  
Universitas Hasanuddin  
Makassar*

Oleh

**Muhammad Alif Fikri**

**D041 17 1002**

**DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2024**



**LEMBAR PENGESAHAN SKRIPSI**

**SIMULASI *TUNING* PENGENDALI PID DENGAN  
ALGORITMA GENETIKA PADA SISTEM KENDALI  
KECEPATAN MOTOR DC**

Disusun dan diajukan oleh

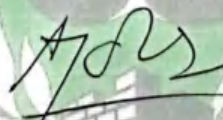
**Muhammad Alif Fikri**

**D041171002**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka  
Penyelesaian Studi Program Sarjana Program Studi Teknik Elektro  
Fakultas Teknik Universitas Hasanuddin  
Pada Tanggal 24 Juni 2024  
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

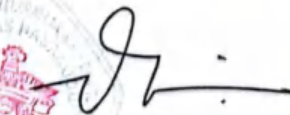
Pembimbing Utama,



Dr. A. Ejah Umraeni Salam, S.T., M.T.

NIP 19720908 199702 2 001

Ketua Program Studi,



Dr. Eng. Ir. Dewiani, M.T., IPM

NIP 19691026 199412 2 001



## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Muhammad Alif Fikri

NIM : D041171002

Program Studi : Teknik Elektro

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**“ SIMULASI *TUNING* PENGENDALI PID DENGAN ALGORITMA  
GENETIKA PADA SISTEM KENDALI KECEPATAN MOTOR DC ”**

Adalah karya tulisan saya sendiri dan bukan merupakan pengambil alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri. Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing dan semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 25 Juni 2024

Yang Menyatakan



METERAI  
TEMPEL  
71CALX249946168

Muhammad Alif Fikri



## KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa yang berkat rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi ini dengan judul “SIMULASI *TUNING* PENGENDALI PID DENGAN ALGORITMA GENETIKA PADA SISTEM KENDALI KECEPATAN MOTOR DC ”. Skripsi ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik pada Departemen Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin.

Penulis bersyukur berkat bimbingan, dukungan, serta bantuan dari berbagai pihak, sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, yang berkat rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan skripsi ini.
2. Kedua orang tua, saudara, serta seluruh keluarga penulis atas doa restu, bantuan, nasehat, dan motivasinya.
3. Ibu Dr. A. Ejah Umraeni Salam, S.T., M.T. selaku pembimbing yang telah banyak meluangkan waktu, tenaga, dan pikirannya selama membimbing dan mengarahkan penulis dalam penyusunan skripsi ini.
4. Ketua Departemen Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin Ibu Dr.Eng.Ir. Dewiani, M.T.
5. Bapak Muh Anshar, S.T. M.Sc (reaserch), Ph. D dan Bapak Prof. Dr. Ing. Faizal Arya Samman, S.T., M.T., selaku dosen penguji yang telah banyak memberikan kritik dan saran dalam penyusunan skripsi ini.
6. Seluruh Dosen dan Staf Akademik Departemen Elektro, Fakultas Teknik, Universitas Hasanuddin atas pengabdian dan pelayanannya.
7. Teman-teman EQUAL17ER atas kebersamaan dan dukungannya selama i.



8. Dan untuk semua pihak yang tak dapat penulis sebutkan satu per satu, yang telah memberikan dukungan baik secara langsung, maupun tidak langsung sehingga penulis dapat menyelesaikan penyusunan skripsi ini.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini yang dapat diperbaiki ataupun dikembangkan kedepannya. Akhir kata, penulis berharap skripsi ini dapat bermanfaat bagi rekan-rekan mahasiswa, dan para pembaca sekalian.

Makassar, 25 Juni 2024

Muhammad Alif Fikri

D041171002





## ABSTRAK

Pengendali *Proportional Integral Derivatif* (PID) adalah salah satu metode kendali yang banyak digunakan dalam sistem kendali kecepatan motor DC. Pada penelitian ini Algoritma Genetika digunakan untuk mengoptimalkan nilai  $K_p$ ,  $K_i$  dan  $K_d$  pada pengendalian empat jenis motor DC yakni motor DC penguatan magnet, motor DC penguatan terpisah, motor DC penguatan shunt, dan motor DC penguatan seri. Digunakan empat jenis indeks performansi yakni ISE, ITSE, IAE dan ITAE yang kemudian akan ditentukan fungsi objektif terbaik untuk digunakan dalam proses *tuning* parameter PID dengan Algoritma Genetika. Terdapat lima skema simulasi yang akan dilakukan untuk membandingkan kinerja pengendali PID kecepatan motor DC hasil *tuning* algoritma genetika terhadap pengendali PID Ziegler Nichols yaitu simulasi fase learning parameter PID, pengujian dengan variasi kecepatan referensi, pengujian variasi beban, pengujian beban tinggi, dan pengujian variasi kecepatan referensi dan beban.

Dari hasil penelitian yang dilakukan, ditentukan bahwa IAE merupakan fungsi objektif terbaik untuk digunakan dalam melakukan tuning parameter pengendali PID dengan algoritma genetika. Didapatkan parameter PID optimal untuk dengan karakteristik respon *unit step* terbaik untuk (1) motor DC magnet : *rise time* = 0.1268 detik, *settling time* = 1.3589 detik, *overshoot* = 4.9306 %, dan *steady state error* = 0%; (2) motor DC penguatan terpisah : *rise time* = 0.0142 detik, *settling time* = 1.0397 detik, *overshoot* = 4.6745%, dan *steady state error* = 0%; (3) motor DC shunt : *rise time* = 0.0045 detik, *settling time* = 1.0108 detik, *overshoot* = 2.6395%, dan *steady state error* = 0%; (4) motor DC seri : *rise time* = 0.0175 detik, *settling time* = 1.0283 detik, *overshoot* = 1.2555%, dan *steady state error* = 0%. Setelah membandingkan metode *tuning* pengendali PID dengan Algoritma Genetika dan metode *tuning* PID Ziegler Nichols, ditentukan bahwa respon *unit step* yang dihasilkan pada tuning Algoritma Genetika jauh lebih baik dan pada kondisi kecepatan dan beban yang berubah, keluaran sistem yang dihasilkan lebih stabil.

Kata kunci : Motor DC, Pengendali PID, Algoritma Genetika



## ABSTRACT

The *Proportional Integral Derivative* (PID) Controller is one of the most widely used control methods for DC motor speed control system. In this research, the Genetic Algorithm is used to optimize the values of  $K_p$ ,  $K_i$  and  $K_d$  on the control of four types of DC motors: magnetic DC motor, separately excited DC motor, shunt-connected DC motor and series-connected DC motor. There are four types of performance index namely ISE, ITSE, IAE and ITAE which will then be determined the best objective function used in PID parameter tuning process with Genetic Algorithm. There are five simulation schemes to compare the performance of the DC motor speed PID controller from the genetic algorithm tuning to the *Ziegler Nichols* PID controllers: PID parameter learning phase simulation, test with reference speed variation, load variation test, high load test, and test of the variation of reference speed and load.

From the results of the research, it was determined that IAE is the best objective function to be used in tuning the parameters of the PID controller with the genetic algorithm. Obtained optimum PID parameters with the best step unit response characteristics for (1) magnetic DC motor : rise time = 0.1268 seconds, settling time = 1.3589 seconds, overshoot = 4.9306 %, and steady state error = 0%; (2) separately excited DC motor : rise time = 0.0142 seconds, settling time = 1.0397 seconds, overshoot = 4.6745%, and steady state error = 0%; (3) shunt-connected DC motor : rise time = 0.0045 seconds, settling time = 1.0108 seconds, overshoot = 2.6395%, and steady state error = 0%; (4) series-connected DC motor : rise time = 0.0175 seconds, settling time = 1.0283 seconds, overshoot = 1.2555%, and steady state error = 0%. After comparing the PID controller tuning method with the Genetic Algorithm and Ziegler Nichols's PID Tuning method, it was determined that the step unit response produced on the genetic algorithm tuning was much better and at changing speed and load conditions, the system output produced was more stable.

Keywords : DC Motor, PID Controller, genetic algorithm





## DAFTAR ISI

HALAMAN JUDUL .....	i
KATA PENGANTAR .....	Iv
ABSTRAK .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xii
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Masalah .....	3
1.4 Batasan Masalah .....	3
1.5 Metode Penelitian .....	3
1.6 Sistematikan Penelitian .....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Motor DC .....	5
2.1.1 Persamaan Umum Motor DC .....	6
2.2 Parameter <i>Step Respon</i> .....	7
2.3 Kontroler Proportional-Integral-Derivative (PID) .....	8
2.3.1 Aksi Proporsional .....	9
2.3.2 Aksi Integral .....	9
2.3.3 Aksi Derivatif .....	9
2.4 Aturan Tuning Kontroler PID Ziegler-Nichols .....	10
2.4.1 Metode <i>Open Loop</i> .....	10
2.4.2 Metode <i>Closed Loop</i> .....	11
2.5 <i>Genetic Algorithm</i> (GA) .....	12
2.5.1 Inisialisasi Populasi .....	13
2.5.2 Fungsi Objektif dan Nilai Fitness .....	14
2.5.3 Seleksi .....	14
2.5.4 <i>Crossover</i> .....	15
2.5.5 Mutasi .....	17
2.6 Indeks Performansi Kesalahan .....	18
2.6.1 <i>Integral Square Error (ISE)</i> .....	18
2.6.2 <i>Integral of Time Multiplied by Square Error (ITSE)</i> .....	19
2.6.3 <i>Integral of Absolute value of Error (IAE)</i> .....	19
2.6.4 <i>Integral of Time Multiplied by Absolute value of Error (ITAE)</i> .....	19
<b>BAB III METODE PENELITIAN</b> .....	<b>20</b>
3.1 Pemodelan Motor DC .....	20
3.1.1 Pemodelan Motor DC Penguatan Magnet Permanen .....	20
3.1.2 Pemodelan Motor DC Penguatan Terpisah .....	21
3.1.3 Pemodelan Motor DC Penguatan Shunt .....	22
3.1.4 Pemodelan Motor DC Penguatan Seri .....	24
3.1.5 Simulasi Motor DC tanpa Pengendali .....	25
3.2 Perancangan Kendali PID Motor DC .....	26
3.2.1 Kendali PID dengan Metode Ziegler-Nichols .....	26
3.2.2 Kendali PID dengan Metode Algoritma Genetika .....	27



3.3 Perancangan Pengujian .....	40
3.3.1 Simulasi Pengujian Variasi Kecepatan Referensi.....	30
3.3.2 Simulasi Pengujian Varasi Beban dan Beban Tinggi .....	30
3.3.3 Simulasi Pengujian Variasi Beban dan Kecepatan .....	31
BAB IV HASIL DAN PEMBAHASAN .....	32
4.1 Simulasi Motor DC Tanpa Pengendali .....	32
4.1.1 Motor DC Penguatan Magnet .....	32
4.1.2 Motor DC Penguatan Terpisah .....	33
4.1.3 Motor DC Penguatan Shunt .....	33
4.1.4 Motor DC Penguatan Seri .....	34
4.2 Simulasi Motor DC dengan PID-Ziegler Nichols .....	35
4.2.1 Simulasi Motor DC Penguatan Magnet dengan PID-Ziegler Nichols .....	35
4.2.2 Simulasi Motor DC Penguatan Terpisah dengan PID-Ziegler Nichols .....	36
4.2.3 Simulasi Motor DC Penguatan Shunt dengan PID-Ziegler Nichols .	37
4.2.4 Simulasi Motor DC Penguatan Seri dengan PID-Ziegler Nichols ...	37
4.3 Simulasi Motor DC dengan PID-GA .....	38
4.3.1 Tahap Learning Algoritma Genetika .....	38
4.3.1.1 Tuning PID-GA pada Motor DC Penguatan Magnet .....	39
4.3.1.2 Tuning PID-GA pada Motor DC Penguatan Terpisah .....	41
4.3.1.3 Tuning PID-GA pada Motor DC Penguatan Shunt.....	43
4.3.1.4 Tuning PID-GA pada Motor DC Penguatan Seri .....	45
4.3.2 Penentuan Metode Tuning PID Terbaik .....	47
4.3.3 Tahap Pengujian .....	48
4.3.3.1 Motor DC Penguatan Magnet .....	48
4.3.3.1.1. Hasil Simulasi Dengan Variasi Kecepatan motor DC Penguatan Magnet .....	48
4.3.3.1.2. Hasil Simulasi Dengan Variasi Beban Motor DC Penguatan Magnet .....	48
4.3.3.1.3. Hasil Simulasi Dengan Variasi Beban dan Kecepatan Motor DC Penguatan Magnet .....	49
4.3.3.1.4. Pengujian Performa Motor DC Penguatan Magnet Pada Beban Tinggi .....	50
4.3.3.2 Motor DC Penguatan Terpisah .....	51
4.3.3.2.1. Hasil Simulasi Dengan Variasi Kecepatan motor DC Penguatan Terpisah .....	51
4.3.3.2.2. Hasil Simulasi Dengan Variasi Beban Motor DC Penguatan Terpisah .....	51
4.3.3.2.3. Hasil Simulasi Dengan Variasi Beban dan Kecepatan Motor DC Penguatan Terpisah.....	52
4.3.3.2.4. Pengujian Performa Motor DC Penguatan Terpisah Pada Beban Tinggi .....	53
	54



4.3.3.3	Motor DC Penguatan Shunt .....	
4.3.3.3.1.	Hasil Simulasi Dengan Variasi Kecepatan motor DC Penguatan Shunt .....	54
4.3.3.3.2.	Hasil Simulasi Dengan Variasi Beban Motor DC Penguatan Shunt .....	55
4.3.3.3.3.	Hasil Simulasi Dengan Variasi Beban dan Kecepatan Motor DC Penguatan Shunt .....	55
4.3.3.3.4.	Pengujian Performa Motor DC Penguatan Shunt Pada Beban Tinggi .....	56
4.3.3.4	Motor DC Penguatan Seri .....	58
4.3.3.4.1.	Hasil Simulasi Dengan Variasi Kecepatan motor DC Penguatan Seri .....	58
4.3.3.4.2.	Hasil Simulasi Dengan Variasi Beban Motor DC Penguatan Seri .....	58
4.3.3.4.3.	Hasil Simulasi Dengan Variasi Beban dan Kecepatan Motor DC Penguatan seri .....	59
4.3.3.4.4.	Pengujian Performa Motor DC Penguatan Seri Pada Beban Tinggi .....	60
BAB IV KESIMPULAN DAN SARAN .....		62
5.1	Kesimpulan .....	62
5.2	Saran .....	63
DAFTAR PUSTAKA .....		64
LAMPIRAN .....		66



## DAFTAR GAMBAR

Gambar 2.1	Rangkaian Ekvivalen dari Motor DC Penguatan : (a) Terpisah; (b) Magnet Permanen; (c)Terhubung-Shunt; dan (d) Terhubung-Seri .....	6
Gambar 2.2	Parameter <i>Step-respon</i> .....	8
Gambar 2.3	Respon <i>Open loop</i> sistem membentuk kurva huruf S .....	10
Gambar 2.4	Respon <i>Closed loop</i> sistem berosilasi permanen dengan periode P .....	11
Gambar 2.5	Representasi Kromosom pada GA .....	13
Gambar 2.6	Diagram Alir GA .....	13
Gambar 2.7	Representasi Kromosom dengan Bilangan Biner .....	14
Gambar 2.8	Representasi metode seleksi <i>Stochastic uniform</i> .....	15
Gambar 2.9	Diagram Alir Proses <i>Crossover</i> .....	16
Gambar 2.10	Representase Metode <i>Scattered Crossover</i> .....	17
Gambar 2.11	Diagram Alir Proses Mutasi .....	18
Gambar 2.12	Representase Metode Mutasi <i>Uniform</i> .....	18
Gambar 3.1	Model Motor DC Penguatan Magnet pada Simulink MATLAB .....	21
Gambar 3.2	Model Motor DC Penguatan Terpisah pada Simulink MATLAB .....	22
Gambar 3.3	Model Motor DC Penguatan Shunt pada Simulink MATLAB .....	24
Gambar 3.4	Model Motor DC Penguatan Seri pada Simulink MATLAB .....	25
Gambar 3.5	Model Motor DC tanpa pengendali .....	25
Gambar 3.6	Model Simulink Motor DC dengan kendali PID .....	27
Gambar 3.7	Reprentasi individu pada proses iterasi Algoritma Genetika .....	28
Gambar 3.8	Model Simulasi Pengendali PID Kecepatan motor DC dangan Algoritma Genetika .....	29
Gambar 3.9	Model Simulink ISE .....	29
Gambar 3.10	Model Simulink ITSE .....	29
Gambar 3.11	Model Simulink IAE .....	29
Gambar 3.12	Model Simulink ITAE .....	29
Gambar 3.13	Sinyal kecepatan referensi untuk simulasi variasi kecepatan .....	30
Gambar 3.14	Sinyal Input Variasi Beban Torka .....	31
Gambar 4.1	Sinyal kecepatan motor DC Penguatan Magnet tanpa pengendali .....	32
Gambar 4.2	Sinyal kecepatan motor DC Penguatan terpisah tanpa pengendali .....	33
Gambar 4.3	Sinyal kecepatan motor DC Penguatan Shunt tanpa pengendali .....	34
Gambar 4.4	Sinyal kecepatan motor DC Penguatan Seri tanpa pengendali .....	35
Gambar 4.5	Sinyal kecepatan motor DC Penguatan Magnet kondisi <i>open loop</i> .....	35
Gambar 4.6	Sinyal kecepatan motor DC Penguatan Terpisah kondisi <i>open loop</i> ....	36
Gambar 4.7	Sinyal kecepatan motor DC Penguatan Shunt kondisi <i>open loop</i> .....	37
Gambar 4.8	Sinyal kecepatan motor DC Penguatan Seri kondisi <i>open loop</i> .....	38
Gambar 4.9	Grafik Konvergensi PID – GA Motor DC Penguatan magnet .....	39
Gambar 4.10	Respon Motor dengan kendali PID Motor DC Penguatan magnet .....	40
Gambar 4.11	Grafik Konvergensi PID – GA Motor DC Penguatan terpisah .....	41
Gambar 4.12	Respon Motor dengan kendali PID Motor DC Penguatan terpisah .....	42
Gambar 4.13	Grafik Konvergensi PID – GA Motor DC Penguatan Shunt.....	43
Gambar 4.14	Respon Motor dengan kendali PID Motor DC Penguatan shunt .....	44
	Grafik Konvergensi PID – GA Motor DC Penguatan Seri .....	45
	Respon Motor dengan kendali PID Motor DC Penguatan Seri .....	46
	Perbandingan respon kendali motor DC penguatan magnet dengan variasi kecepatan. ....	48



Gambar 4.18	Perbandingan respon kendali motor DC penguatan magnet dengan penambahan sinyal beban. ....	49
Gambar 4.19	Perbandingan respon kendali motor DC penguatan magnet mengikuti set-point dengan penambahan sinyal beban.....	49
Gambar 4.20	Respon Motor DC Penguatan Magnet terhadap Beban Tinggi .....	50
Gambar 4.21	Perbandingan respon kendali motor DC penguatan terpisah dengan variasi kecepatan .....	51
Gambar 4.22	Perbandingan respon kendali motor DC penguatan terpisah dengan penambahan sinyal beban.....	52
Gambar 4.23	Perbandingan respon kendali motor DC penguatan terpisah mengikuti <i>set-point</i> dengan penambahan sinyal beban.....	52
Gambar 4.24	Respon Motor DC Penguatan Terpisah terhadap Beban Tinggi .....	53
Gambar 4.25	Respon Motor DC Penguatan Terpisah terhadap Beban Tinggi setelah proses <i>tuning</i> ulang .....	54
Gambar 4.26	Perbandingan respon kendali motor DC penguatan shunt dengan variasi kecepatan .....	54
Gambar 4.27	Perbandingan respon kendali motor DC penguatan shunt dengan penambahan sinyal beban.....	55
Gambar 4.28	Perbandingan respon kendali motor DC penguatan shunt mengikuti <i>set-point</i> dengan penambahan sinyal beban.....	56
Gambar 4.29	Respon Motor DC Penguatan Shunt terhadap Beban Tinggi .....	57
Gambar 4.30	Respon Motor DC Penguatan Shunt terhadap Beban Tinggi setelah proses <i>tuning</i> ulang .....	57
Gambar 4.31	Perbandingan respon kendali motor DC penguatan seri dengan variasi kecepatan.....	58
Gambar 4.32	Perbandingan respon kendali motor DC penguatan seri dengan penambahan sinyal beban.....	59
Gambar 4.33	Perbandingan respon kendali motor DC penguatan seri mengikuti <i>set-point</i> dengan penambahan sinyal beban.....	59
Gambar 4.34	Respon Motor DC Penguatan Seri terhadap Beban Tinggi .....	60
Gambar 4.35	Respon Motor DC Penguatan Seri terhadap Beban Tinggi setelah proses <i>tuning</i> ulang .....	61



## DAFTAR TABEL

Tabel 2.1	Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols Metode <i>Open Loop</i> .....	11
Tabel 2.2	Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols Metode <i>Closed Loop</i> ...	12
Tabel 3.1	Nilai Parameter Motor DC Penguatan Magnet .....	20
Tabel 3.2	Nilai Parameter Motor DC Penguatan Terpisah .....	21
Tabel 3.3	Nilai Parameter Motor DC Penguatan Shunt .....	23
Tabel 3.4	Nilai Parameter Motor DC Penguatan Seri .....	24
Tabel 3.5	Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols Metode <i>Open Loop</i> .....	26
Tabel 3.6	Aturan <i>Tuning</i> Kontroler PID Ziegler-Nichols Metode <i>Closed Loop</i> ...	26
Tabel 4.1	Parameter Respon Motor DC penguatan magnet.....	32
Tabel 4.2	Parameter Respon Motor DC penguatan terpisah .....	33
Tabel 4.3	Parameter Respon Motor DC penguatan shunt .....	34
Tabel 4.4	Parameter Respon Motor DC penguatan seri .....	34
Tabel 4.5	Parameter PID-Ziegler Nichols Motor DC penguatan Magnet.....	36
Tabel 4.6	Parameter PID-Ziegler Nichols Motor DC penguatan Terpisah.....	36
Tabel 4.7	Parameter PID-Ziegler Nichols Motor DC penguatan Shunt.....	37
Tabel 4.8	Parameter PID-Ziegler Nichols Motor DC penguatan Seri .....	38
Tabel 4.9	Parameter PID-GA Motor DC Penguatan Magnet.....	39
Tabel 4.10	Parameter Respon Motor DC dengan Kendali PID-GA Motor penguatan Magnet.....	40
Tabel 4.11	Parameter PID-GA Motor DC Penguatan terpisah.....	41
Tabel 4.12	Parameter Respon Motor DC dengan Kendali PID-GA Motor penguatan terpisah .....	42
Tabel 4.13	Parameter PID-GA Motor DC Penguatan Shunt.....	43
Tabel 4.14	Parameter Respon Motor DC dengan Kendali PID-GA Motor penguatan Shunt .....	44
Tabel 4.15	Parameter PID-GA Motor DC Penguatan Seri.....	45
Tabel 4.16	Parameter Respon Motor DC dengan Kendali PID-GA Motor penguatan Seri .....	46
Tabel 4.17	Hasil metode terbaik tuning PID motor DC dengan Algoritma Genetika .....	47
Tabel 4.18	Parameter respon Motor DC penguatan magnet pada beban tinggi.....	50
Tabel 4.19	Parameter respon Motor DC penguatan terpisah pada beban tinggi.....	53
Tabel 4.20	Parameter hasil <i>tuning</i> ulang PID-GA Motor DC Penguatan Terpisah..	53
Tabel 4.21	Parameter respon Motor DC penguatan shunt pada beban tinggi.....	56
Tabel 4.22	Parameter hasil <i>tuning</i> ulang PID-GA Motor DC Penguatan Shunt.....	57
Tabel 4.23	Parameter respon Motor DC penguatan seri pada beban tinggi.....	60
Tabel 4.24	Parameter hasil <i>tuning</i> ulang PID-GA Motor DC Penguatan Seri .....	61





# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Motor DC (*Direct Current*) atau motor arus searah termasuk dalam kategori jenis motor yang paling banyak digunakan baik dalam lingkungan industri, peralatan rumah tangga hingga ke mainan anak-anak ataupun sebagai piranti pendukung sistem instrumen elektronik. Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dan lain-lain. Dalam mengontrol kecepatan dan performa Motor DC, metode yang banyak digunakan adalah metode pengontrolan PID[1].

Kontrol PID (*Proportional Integral Derivatif*) merupakan teknik kontrol yang sering digunakan dalam rekayasa kontrol, dimana kendali PID ini terdiri dari tiga macam kendali yang dikombinasikan antara lain (*Proportional, Integral dan Derivative*) dimana parameter tersebut dapat menentukan kualitas respon dari suatu sistem kontrol [2]. Saat ini kontrol PID merupakan teknik yang paling banyak diadaptasi pada pengontrolan variabel proses di industri. Dimana dapat dijumpai kenyataan bahwa sebagian besar industri yang bergerak dalam bidang proses (seperti industri kimia, pulp, makanan, minyak dan gas) menggunakan PID sebagai komponen utama dalam pengontrolannya. Kontrol PID telah digunakan untuk kontrol proses di sebagian besar aplikasi teknik selama beberapa dekade [13].

Kontrol PID memiliki struktur sederhana dan memberikan kinerja yang dapat diterima untuk beberapa aplikasi industri. Memilih parameter pengontrol

yang tepat sangat penting karena masing – masing parameter yang ada berpengaruh pada output kontrol yang dihasilkan. Teknik yang paling sederhana dan awam digunakan untuk kontrol PID sederhana adalah proses trial



and error kemudian , Ziegler dan Nichols mengusulkan metode terkenal untuk menemukan parameter kontroler PID. Pada tahun 2022 Aliyah Muthmainnah melakukan penelitian berupa tuning nilai PID dengan metode *Particle Swarm Optimization* (PSO) dimana didapati *tuning* parameter dengan PSO menghasilkan respon motor DC yang lebih baik dibandingkan dengan metode Ziegler-Nichols.

Pada penelitian ini penulis mengajukan proposal berupa Simulasi Sistem Kontrol Kecepatan Motor DC Menggunakan Pengendali PID Dengan Algoritma Genetika (GA) sebagai metode pengoptimalan parameter PID. Algoritma Genetika ini didasarkan pada proses genetik yang ada dalam makhluk hidup; yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau "siapa yang kuat, dia yang bertahan (*survive*)". Dengan meniru teori evolusi ini, Algoritma Genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata. Algoritma Genetika akan digunakan untuk mencari parameter – parameter pada kontrol PID sehingga dihasilkan sistem kendali kecepatan motor DC yang optimal.

## 1.2. Rumusan Masalah

- a. Bagaimana melakukan simulasi tuning parameter PID pada Motor DC dengan Algoritma Genetika ?
- b. Bagaimana pengaruh perbedaan indeks kesalahan performansi sebagai fungsi objektif dalam simulasi Algoritma Genetika terhadap *step respon* motor DC ?
- c. Bagaimana Perbandingan performa sistem kendali motor DC antara PID-Ziegler Nichols dan PID-GA ?



### 1.3. Tujuan Penelitian

- a. Mengembangkan model simulasi PID kontroler dengan *Genetic Algorithm Tuning* untuk kendali kecepatan Motor DC.
- b. Menentukan pengaruh indeks kesalahan performansi sebagai fungsi objektif dalam simulasi Algoritma Genetika terhadap karakteristik steady state motor DC.
- c. Menentukan proses tuning yang paling efektif untuk mendapatkan parameter PID yang optimal.

### 1.4. Batasan Masalah

Untuk mengoptimalkan hasil penelitian, penelitian ini memberikan batasan-batasan masalah yaitu sebagai berikut:

- a. Menggunakan Motor DC yang dimodelkan dalam bentuk fungsi alih.
- b. Menggunakan empat jenis Motor DC berdasarkan sumber penguatannya (Motor DC magnet, motor DC penguatan terpisah, motor DC shunt, dan Motor DC seri).
- c. Performa sistem kendali motor DC dievaluasi pada tanggapan undak satuan. (*rise time, overshoot, settling time, dan steady state error*).

### 1.5. Metode Penelitian

Untuk menghasilkan tugas akhir yang komprehensif, maka dalam penelitian akan digunakan metode sebagai berikut:

- a. Studi literatur

Studi literatur dilakukan untuk membentuk landasan teori yang konkrit berdasarkan literatur terkait, sebelum melakukan implementasi dan pengujian secara langsung

- b. Pengumpulan Data

Upaya pengumpulan data sekunder parameter-parameter motor yang digunakan dalam penelitian ini.



c. Diskusi dan konsultasi

Melakukan dialog secara langsung kepada pembimbing dan pihak - pihak yang berkompeten di bidang terkait untuk mendapatkan pengetahuan mengenai penelitian yang dilakukan

d. Perancangan Sistem

Perancangan Kendali Motor DC menggunakan kendali PID dengan metode Ziegler-Nichols dan Algoritma Genetika menggunakan software MATLAB R2021b.

## 1.6. Sistematika Penelitian

### BAB I : PENDAHULUAN

Bab ini akan menjelaskan mengenai latar belakang pemilihan judul skripsi, deskripsi dan rumusan masalah, tujuan penelitian, batasan masalah, metode penelitian dan sistematika penulisan.

### BAB II : TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang berkaitan dengan penelitian yang telah dilaksanakan

### BAB III : METODOLOGI PENELITIAN

Bab ini berisi pembahasan analisis sistem dan perancangan sistem perangkat lunak.

### BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi ulasan hasil penelitian dan pembahasan terhadap pengujian yang telah dilaksanakan

### BAB V : PENUTUP

Bab ini berisi kesimpulan yang didapat menjalani penelitian ini serta saran yang diharapkan dapat bermanfaat dalam usaha untuk melakukan perbaikan dan pengembangan dalam penelitian ini.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Motor DC

Motor arus searah (motor DC) adalah mesin yang mengubah energi listrik arus searah menjadi energi mekanis. Sebuah motor listrik berfungsi untuk mengubah daya listrik menjadi daya mekanik. Pada prinsip pengoperasiannya, motor arus searah sangat identik dengan generator arus searah. Kenyataannya mesin yang bekerja sebagai generator arus searah akan dapat bekerja sebagai motor arus searah. Oleh sebab itu, sebuah mesin arus searah dapat digunakan baik sebagai motor arus searah maupun generator arus searah [1].

Bentuk fisik motor DC pada dasarnya sama dengan generator DC, dimana komponen utamanya terdiri dari tiga bagian, yakni:

- Kumparan (belitan) jangkar yang terletak pada rotor
- Kumparan (belitan) medan yang terletak pada stator, dan
- celah udara antara kumparan jangkar dan kumparan medan.

berdasarkan sumber penguatannya (*exciter*), Penggolongan Motor DC menjadi empat jenis yaitu [2] :

a. Motor DC Penguatan Magnet Permanen

Jenis Motor DC yang medan penguatannya disuplai dengan magnet permanen. Sehingga aliran arus listrik yang mengalir hanya pada rangkaian rotor.

b. Motor DC Penguatan Terpisah

Jenis Motor DC yang sumber arus kumparan armatur dan kumparan medan disuplai dari sumber tegangan DC yang berbeda dan tidak berhubungan satu dengan yang lain.

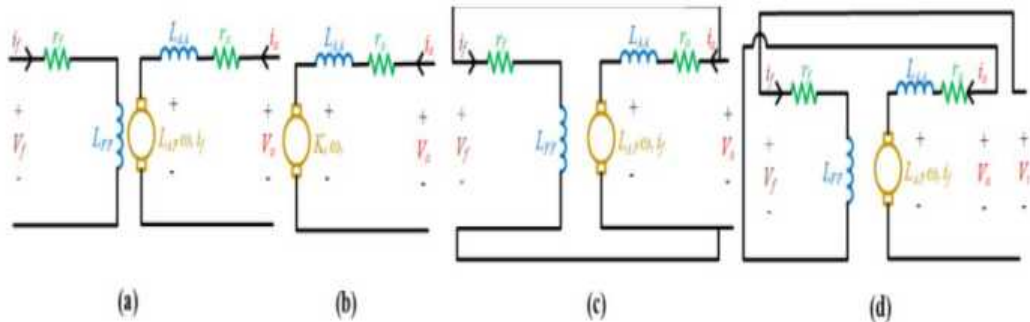
c. Motor DC Penguatan Shunt

Motor DC yang sumber arus kumparan armatur dan kumparan medan disuplai dari sumber DC yang sama dan kumparan – kumparannya dihubungkan secara paralel.



d. Motor DC Penguatan Seri

Jenis Motor DC yang sumber arus kumparan armatur dan kumparan medan disuplai dari sumber DC yang sama dan kumparan – kumparannya dihubungkan secara seri.



Gambar 2.1 Rangkaian Ekivalen dari Motor DC Penguatan : (a) Terpisah; (b) Magnet Permanen; (c)Terhubung-Shunt; dan (d) Terhubung-Seri [2]

2.1.1. Persamaan umum motor DC

Karakteristik dari motor DC dapat digambarkan melalui persamaan elektromagnetik dari sistemnya [2].

- Persamaan *Back Emf*

$$e(t) = \omega(t)K_m i_f(t) \quad (2.1)$$

- Persamaan Tegangan Armatur dan Tegangan Jangkar :

$$\begin{aligned} v_a(t) &= (r_a + \frac{dL_a}{dt}) i_a(t) + e(t) \\ &= (r_a + \frac{dL_a}{dt}) i_a(t) + \omega(t)K_m i_f(t) \end{aligned} \quad (2.2)$$

$$v_f(t) = (r_f + \frac{dL_f}{dt}) i_f(t) \quad (2.3)$$

- Persamaan Torka Elektromagnetik :

$$T_m(t) = K_t i_a(t) i_f(t) \quad (2.4)$$

- Persamaan Mekanik Motor

$$J \frac{d\omega(t)}{dt} + B\omega(t) + T_l \quad (2.5)$$





Dimana,  $v_a(t)$  = Tegangan Jangkar;  $v_f(t)$  = Tegangan Penguatan;  $r_a$  = Resistansi Jangkar;  $r_f$  = Resistansi Penguatan;  $L_a$  = Induktansi Jangkar;  $L_f$  = Induktansi Penguatan;  $i_a(t)$  = Arus Jangkar;  $i_f(t)$  = Arus Penguatan;  $\omega(t)$  = putaran rotor;  $K_m$  = Konstanta Motor;  $J$  = Momen Inersia Rotor;  $B$  = Beban Gesekan Motor;  $T_m$  = Torka yang dibangkitkan;  $T_l$  = Torka Beban

## 2.2 Parameter Step-Respon

Beberapa parameter yang digunakan untuk mengukur performa suatu sistem dalam domain waktu adalah *rise time*, *maximum overshoot*, *settling time*, dan *steady state error*, yang dijelaskan sebagai berikut [3] :

- *Rise Time*

*Rise time*,  $T_r$ , didefinisikan sebagai waktu yang dibutuhkan oleh respon *unit step* untuk naik dari 10% nilai *steady-state* ke 90% nilai *steady-state*. Pendapat lain mengatakan, *rise time* dapat didefinisikan sebagai waktu yang diperlukan untuk mencapai nilai 95% *steady-state* dari 5% *steady state* .

- *Overshoot*

*Overshoot* merupakan besar nilai *step respon* yang melebihi nilai *steady-state* [3]. Persentasi *overshoot* didefinisikan dengan persamaan berikut:

$$\% \text{ overshoot} = 100 \times \frac{\text{nilai}_{\text{maximum}} - \text{nilai}_{\text{steady state}}}{\text{nilai}_{\text{steady state}}} \quad (2.6)$$

Persentasi *overshoot* berada diantara nilai 0 sampai 100.

- *Settling Time*

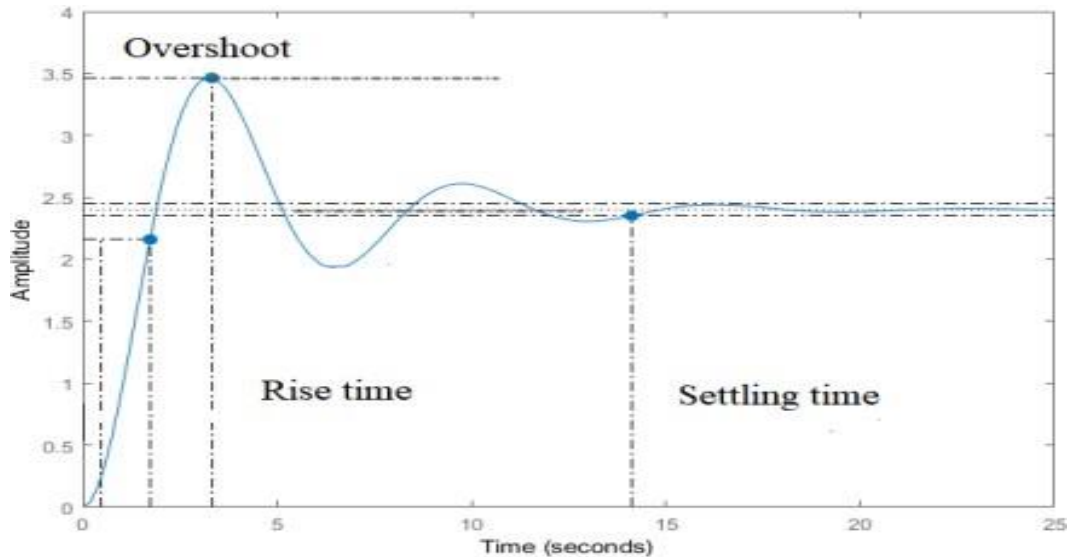
*Settling time*,  $T_s$ , didefinisikan sebagai waktu minimum yang dibutuhkan agar sinyal respon bernilai tetap tidak lebih dari  $\pm 5\%$  dari nilai *steady-state* .



### *Steady State Error*

output sistem pada keadaan *steady-state* tidak persis sama dengan inputnya maka sistem tersebut dinyatakan memiliki *steady-state error*. *Error*

ini merupakan tolak ukur akurasi sistem. *Steady-state error* merupakan selisih atau perbedaan antara output sistem dan input sistem pada keadaan *steady-state*. Penjelasan secara grafis dapat dilihat pada gambar 2.2.



Gambar 2.2 Parameter Step-response

### 2.3 Kontroler *Proportional-Integral-Derivative* (PID)

Kontroler PID merupakan salah satu kontroler yang banyak digunakan untuk memperbaiki performa sistem di industri. Struktur kontroler PID yang sederhana menjadi salah satu faktor mengapa kontroler tersebut diminati.  $u(t)$  merupakan sinyal kontrol dan  $e(t)$  adalah sinyal kesalahan. Sinyal kontrol dari kontroler tersebut merupakan penjumlahan dari 3 aksi yaitu, aksi proporsional ( $P$ ), integral ( $I$ ) dan derivatif ( $D$ ). Kontroler PID memiliki fungsi untuk memperbaiki kesalahan melalui aksi proporsional, menghilangkan kesalahan pada keadaan tunak melalui aksi integral dan mengantisipasi kesalahan akan datang melalui aksi derivatif. Parameter kontroler PID adalah penguatan proporsional  $K_p$ , waktu integral  $T_i$ , dan waktu derivatif  $T_d$  [10]. Algoritma kontroler PID dapat dijelaskan pada Persamaan (2.7).



$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{d}{dt} e(t) \right] \quad (2.7)$$

### 2.3.1. Aksi Proporsional

Aksi proporsional memberikan sinyal kontrol yang proporsional terhadap sinyal kesalahan seperti pada Persamaan (2.8) Penggunaan aksi proporsional saja tidak menghilangkan kesalahan keadaan tunak [5].

$$u(t) = K_p e(t) \quad (2.8)$$

$K_p$  adalah penguatan proporsional yang dapat mempercepat dan menyebabkan osilasi pada respon sistem apabila nilainya diperbesar.

### 2.3.2. Aksi Integral

Aksi integral berfungsi untuk memastikan bahwa output sistem mencapai nilai setpoint pada keadaan tunak. Sinyal kesalahan positif akan menyebabkan sinyal kontrol meningkat, sedangkan sinyal kesalahan negatif akan menyebabkan sinyal kontrol menurun pada kontroler dengan aksi integral. Kedua efek aksi integral akan membuat kesalahan pada keadaan tunak bernilai nol [5]. Aksi integral memberikan sinyal kontrol yang proporsional terhadap nilai kesalahan pada waktu sebelumnya seperti pada Persamaan (2.9)

$$u(t) = K_p \left( \frac{1}{T_i} \int_0^t e(t) dt \right) \quad (2.9)$$

$T_i$  merupakan waktu integral. Semakin besar nilai waktu integral, maka respon sistem akan semakin lambat untuk mencapai setpoint. Nilai waktu integral yang kecil dapat mempercepat dan menyebabkan osilasi pada respon sistem.

### 2.3.3. Aksi Derivatif

Aksi derivatif bertujuan untuk meningkatkan kestabilan pada sistem *closed loop*. Dinamika proses umumnya menimbulkan waktu tunda pada perubahan sinyal kontrol, sehingga kontroler terlambat dalam memperbaiki sinyal kesalahan.

besar nilai waktu derivatif, maka redaman sistem akan membesar. Redaman menurun saat waktu derivatif terlalu besar [5]. Aksi derivatif memberikan sinyal kontrol yang proporsional terhadap kesalahan akan datang



seperti dirumuskan pada Persamaan (2.10)

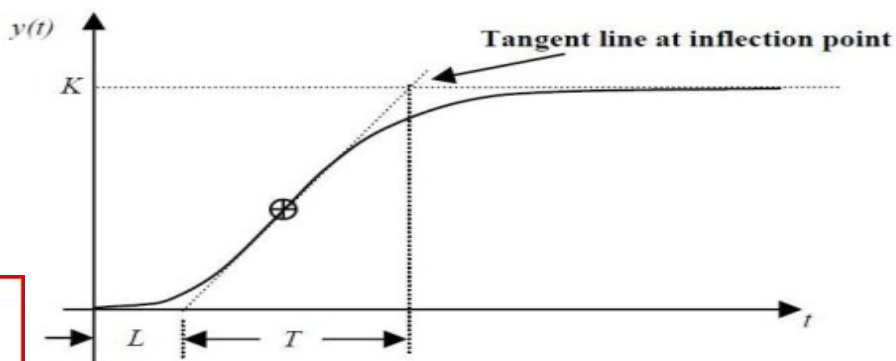
$$u(t) = K_P \left( T_d \frac{d}{dt} e(t) \right) \quad (2.10)$$

## 2.4 Aturan *Tuning* Kontroler PID Ziegler-Nichols

Teknik desain PID yang banyak digunakan di industri adalah metode Ziegler-Nichols. Ziegler-Nichols mengusulkan aturan dalam menentukan nilai parameter kontroler PID berdasarkan karakteristik respon sistem. Ziegler-Nichols memiliki dua metode dalam penggunaan aturan *tuning*-nya.

### 2.4.1. Metode *Open Loop*

Aturan *tuning* metode *open loop* dilakukan dengan mendapatkan respon sistem terhadap masukan sinyal uji *unit step*. Sistem yang tidak memiliki karakteristik integrator akan menghasilkan respon sistem menyerupai kurva huruf S. Jika respon sistem tidak memiliki bentuk seperti kurva tersebut, maka metode ini tidak dapat diterapkan. Respon sistem menyerupai kurva huruf S seperti pada Gambar 2.3, memiliki dua buah karakteristik, yaitu waktu tunda,  $L$  dan waktu konstan,  $T$  yang dapat dilihat. Kedua karakteristik diperoleh dengan menggambar garis singgung yang memotong sumbu waktu dan garis keluaran pada kondisi tunak [4]. Nilai parameter kontroler PID diperoleh dengan mensubstitusikan  $L$  dan  $T$  ke dalam Tabel 2.1.



Gambar 2.3 Respon *Open loop* sistem membentuk kurva huruf S

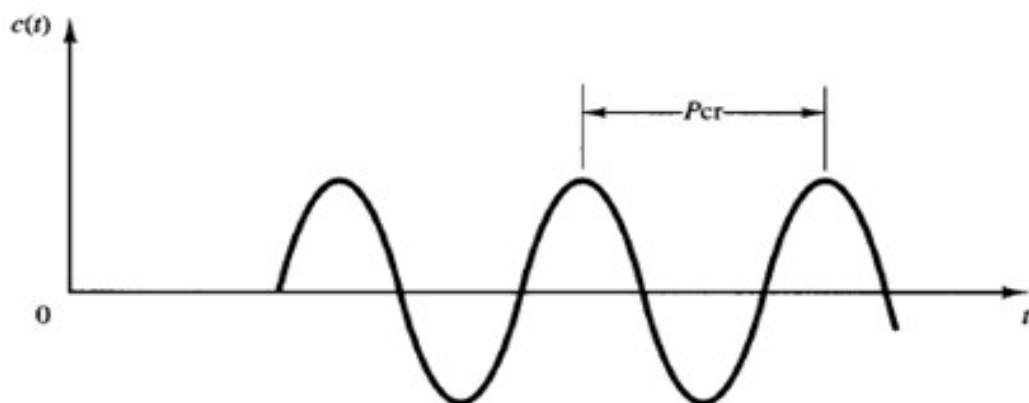


Tabel 2.1 Aturan *Tuning* Kontroler PID Ziegler-Nichols Metode *Open Loop*

Kontroler/ Parameter	$K_p$	$T_i$	$T_d$
P	$T/L$		
PI	$0,9 T/L$	$L/0,3$	
PID	$1,2 T/L$	$2L$	$L/2$

#### 2.4.2. Metode *Closed Loop*

Aturan tuning Ziegler-Nichols metode *closed loop* dilakukan dengan memperoleh penguatan minimal ketika respon sistem mengalami osilasi permanen. Langkah pertama untuk melakukan tuning kontroler PID yaitu mengatur nilai  $T_i = \infty$  dan  $T_d = 0$ , kemudian tingkatan nilai parameter  $K_p$  mulai dari 0 sampai diperoleh nilai penguatan kritis,  $K_{cr}$  yaitu ketika respon sistem mengalami osilasi permanen untuk semua nilai, maka metode ini tidak dapat digunakan. Langkah kedua adalah menentukan periode kritis,  $P_{cr}$ , seperti pada Gambar 2.4, yaitu periode ketika sistem mengalami osilasi permanen pertama kali atau dapat disebut periode respon sistem ketika  $K_p = K_{cr}$  [4]. Aturan *tuning* Ziegler-Nichols metode kedua dapat dilihat pada Tabel 2.2.



2.4 Respon *Closed loop* sistem berosilasi permanen dengan periode  $P_{cr}$



Tabel 2.2 Aturan *Tuning* Kontroler PID Ziegler-Nichols Metode *Closed Loop*

Kontroler/ Parameter	Kp	Ti	Td
P	$0,5K_{cr}$		
PI	$0,45K_{cr}$	$K_{cr} / 1,2$	
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125 P_{cr}$

## 2.5 *Genetic Algorithm* (GA)

*Genetic Algorithm* atau algoritma genetika merupakan metode adaptif yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup; yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau "siapa yang kuat, dia yang bertahan (*survive*)". Dengan meniru teori evolusi ini, Algoritma Genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata [6].

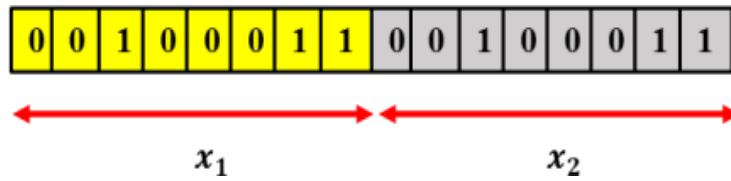
Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-individu, yang masing-masing individu mempresentasikan sebuah solusi yang mungkin bagi persoalan yang ada. Dalam kaitan ini, individu dilambangkan dengan sebuah nilai fitness yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada [7].

Solusi permasalahan direpresentasikan oleh pengkodean individu dalam kromosom yang terdiri dari beberapa karakter, sehingga nilai kromosom. Representasi GA paling banyak menggunakan pengkodean bilangan biner, meskipun terdapat cara representasi lain seperti *ternary*, bilangan bulat, atau bilangan riil. Contohnya, permasalahan ditentukan memiliki solusi optimal berupa variabel  $x_1$  dan  $x_2$  yang dipetakan pada kromosom seperti pada Gambar 2.5.

Proses reproduksi meliputi *crossover* dan mutasi. Setelah melalui proses ini, maka terbentuk generasi baru. Jika solusi masih belum optimal, maka melalui proses seleksi dan reproduksi kembali [6].

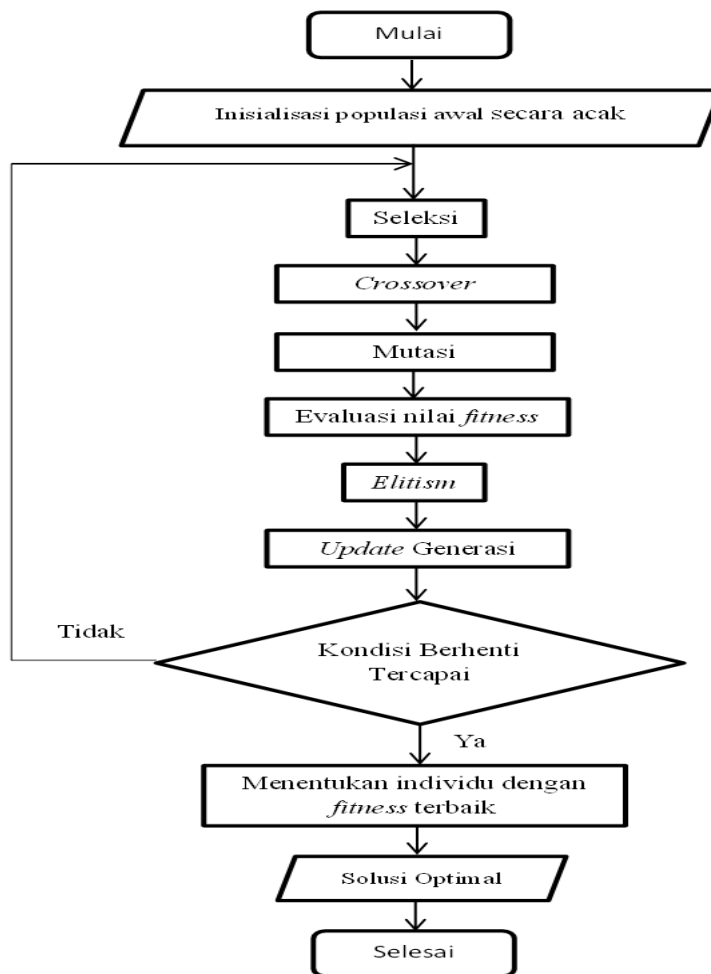






Gambar 2.5 Representasi Kromosom pada GA

Dengan demikian, populasi akan semakin mendekati solusi yang optimal. Mekanisme GA dapat dijelaskan melalui Gambar 2.6



Gambar 2.6 Diagram Alir GA

### 2.5.1. Inisialisasi Populasi

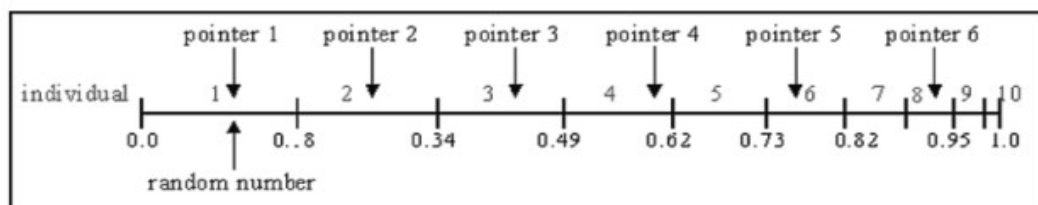


Membangkitkan populasi awal adalah proses membangkitkan sejumlah secara acak atau melalui prosedur tertentu. Ukuran untuk populasi pada masalah yang akan diselesaikan dan jenis operator genetika yang



Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Nilai *fitness* ini yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Masing-masing individu dalam wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai obyektif dari semua individu dalam wadah seleksi tersebut.

Salah satu metode yang digunakan untuk seleksi induk yakni *Stochastic uniform selection*. Metode ini dieksekusi dengan menarik garis dengan panjang yang telah ditentukan kemudian jumlah pointer N ditentukan secara acak untuk menentukan jumlah induk dalam fase seleksi tersebut. Induk ditunjuk beruntun dari garis yang telah dibentuk dengan interval antara *pointer* yang nilainya  $\frac{1}{N}$  sehingga pemilihan induk tidak terdapat bias terhadap nilai *fitness* yang memungkinkan setiap nilai dalam garis yang telah ditentukan dapat terpilih dengan kemungkinan yang sama sebagai induk [8]. Adapun seleksi *Stochastic uniform* dapat dilihat pada gambar 2.8.



Gambar 2.8 Representasi metode seleksi *Stochastic uniform* [8]

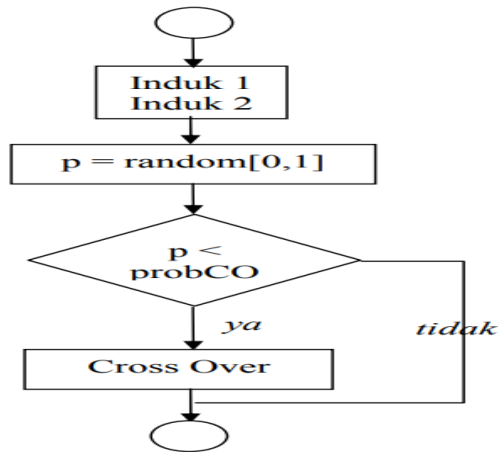
#### 2.5.4. Crossover

Kawin silang (*crossover*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan titik baru dalam ruang pencarian yang siap untuk diuji. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan crossing dengan  $P_c$  antara 0,6 s/d 0,95. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan [6].

Prinsip dari pindah silang ini adalah melakukan operasi (pertukaran, a) pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan kromosom baru. Proses crossover dilakukan pada setiap individu dengan

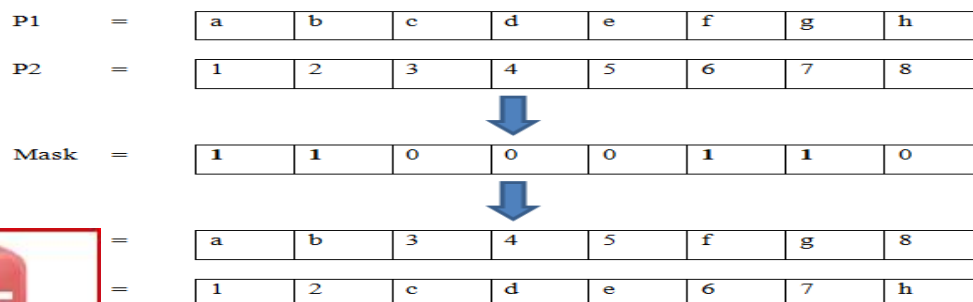


probabilitas *crossover* yang ditentukan. Pada Gambar 2.9 diilustrasikan diagram alir penggunaan probabilitas *crossover* pada proses *crossover*. Operator *crossover* ini bergantung pada representasi kromosom yang dilakukan.



Gambar 2.9 Diagram Alir Proses *Crossover*

Salah satu metode *crossover* yang dapat digunakan dalam algoritma genetika yakni *scattered crossover*. Metode ini dieksekusi dengan membangkitkan suatu nilai biner acak yang panjangnya sesuai dengan kromosom yang telah ditentukan yang disebut *mask*. Nilai *mask* ini kemudian memilih representasi nilai kromosom induk seperti sebuah topeng. Kemudian dari kedua induk nilai yang direpresentasikan 1 pada induk pertama akan dikombinasikan dengan nilai yang direpresentasikan 0 pada induk kedua dan begitu pula sebaliknya sehingga akan membentuk individu baru [8]. Metode ini dapat dilihat pada gambar 2.10



Gambar 2.10 Representase Metode *Scattered Crossover*

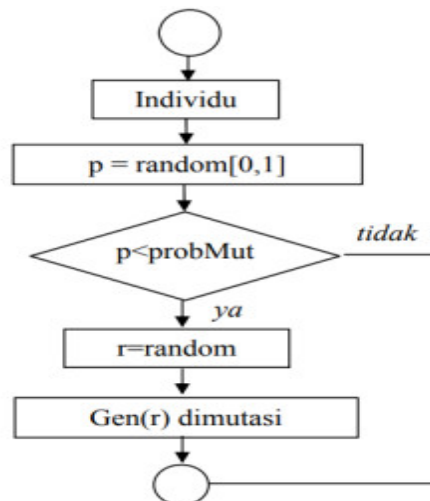


### 2.5.5. Mutasi

Mutasi pada evolusi makhluk hidup merupakan proses pergantian satu alel suatu gen dengan alel gen lain untuk menghasilkan struktur genetika yang baru. Operator ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi [6].

Kromosom anak dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah. Peluang mutasi ( $p_m$ ) didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya 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 anak akan kehilangan kemiripan dari induknya, dan juga algoritma kehilangan kemampuan untuk belajar dari histori pencarian.

Kromosom hasil mutasi harus diperiksa, apakah masih berada pada domain solusi, dan bila perlu bisa dilakukan perbaikan.

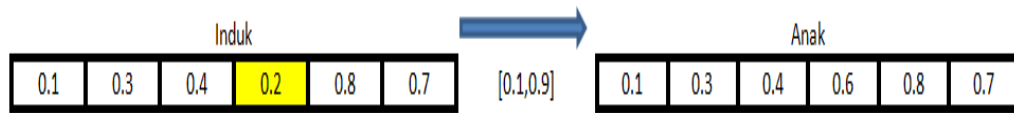


Gambar 2.11 Diagram Alir Proses Mutasi



Salah satu metode mutasi adalah mutasi uniform. Metode ini dilakukan dengan menentukan sebuah nilai acak pada suatu interval batas nilai yang telah ditentukan. Salah satu nilai gen pada sebuah kromosom induk akan digantikan

pada nilai tersebut yang kemungkinan terjadinya telah ditentukan [8]. Metode *mutasi uniform* ini dapat dilihat pada gambar 2.12.



Gambar 2.12 Representase Metode *Mutasi Uniform*

## 2.6 Indeks Kesalahan Performansi

Kontrol optimal tidak dapat didefinisikan dengan tepat. Solusi yang menurut sebuah masalah adalah kondisi optimal, mungkin bukan nilai yang optimal bagi permasalahan lain. Indeks performansi banyak digunakan oleh kalangan praktisi dan juga akademisi untuk membantu dalam menentukan kualitas sebuah sistem. Indeks performansi sendiri sebenarnya adalah fungsi hubungan dimana beberapa karakteristik sistem seperti kondisi optimal dari sistem didefinisikan [9]. Indeks performansi secara umum dapat dirumuskan pada Persamaan (2.11) dimana  $J$  adalah indeks performansi dan  $e$  adalah kesalahan (error).

$$J = \int_0^{\infty} f(e)dt \quad (2.11)$$

Nilai indeks performansi ini mengindikasikan seberapa bagus performansi dari suatu sistem. Pada penjelasan selanjutnya akan dibahas mengenai beberapa indeks performansi untuk kriteria kesalahan yang biasa digunakan dalam perancangan kontroler.

### 2.6.1. *Integral Square Error (ISE)*

Indeks performansi ISE ditunjukkan pada Persamaan (2.12). Indeks performansi ini berhubungan dengan *Mean Square Error (MSE)* [9].

$$\int_0^{\infty} e^2(t)dt \quad (2.12)$$





ISE selain dapat digunakan untuk mendesain sebuah kontroler, juga dapat digunakan sebagai kriteria analisa mengenai seberapa baik sebuah sistem dengan kontroler dalam mengatasi pembebanan.

### 2.6.2. *Integral of Time Multiplied by Square Error (ITSE)*

Indeks performansi ITSE ditunjukkan pada Persamaan (2.13). Bobot pengali waktu pada ITSE membuatnya lebih cepat untuk meminimalkan kesalahan dibandingkan ISE [9].

$$J = \int_0^{\infty} t e^2(t) dt \quad (2.13)$$

### 2.6.3. *Integral of Absolute value of Error (IAE)*

Indeks performansi IAE ditunjukkan pada Persamaan (2.14). IAE memiliki struktur kriteria yang paling mudah diaplikasikan dibandingkan dengan kriteria kesalahan lainnya, karena secara matematis indeks performansi tersebut mudah dioperasikan [9].

$$J = \int_0^{\infty} |e(t)| dt \quad (2.14)$$

Kriteria ini dapat menghilangkan kesalahan yang besar dan kesalahan yang kecil jika dibandingkan dengan ISE. Sistem optimal berdasarkan kriteria ini mempunyai nilai redaman dan fasa transien yang sangat baik

### 2.6.4. *Integral of Time Multiplied by Absolute value of Error (ITAE)*

Indeks performansi dari ITAE ditunjukkan pada Persamaan 2.15. Bobot pengali waktu pada ITAE membuatnya lebih cepat untuk meminimalkan kesalahan dibandingkan IAE [9].

$$J = \int_0^{\infty} t |e(t)| dt \quad (2.15)$$

IAE memiliki nilai kesalahan awal yang besar dan kemudian kesalahan menghilang secara perlahan.

