

IMPLEMENTASI ALGORITMA PARTICLE SWARM OPTIMIZATION (PSO) UNTUK OPTIMASI PARAMETER PENGENDALI *PROPORTIONAL INTEGRAL DERIVATIVE (PID)* PADA PENGENDALIAN STIR OTOMATIS

Fathi Muhammad¹, Ahmad Hatta Novera¹, Hari Handika Setiawan¹, Suci Dwijayanti¹, Zainuddin Nawawi¹, Djulil Amri¹, Bhakti Yudho Suprapto^{1*}

¹Teknik Elektro, Universitas Sriwijaya, Palembang

*Corresponding author e-mail: bhakti@ft.unsri.ac.id

ABSTRAK: Kemajuan yang terus meningkat di bidang teknologi menyebabkan perkembangan teknologi *autonomous vehicle* semakin meningkat. Salah satu komponen penting pada *Autonomous vehicle* adalah *control steering* untuk mengendalikan gerakannya. Dalam proses pengontrolan kemudi terdapat banyak gangguan dan ketidakpastian sehingga berbagai metode dalam pengontrolan kemudi telah banyak dilakukan oleh peneliti. Salah satu metode pengontrolan yang paling populer saat ini adalah dengan metode pengontrolan Proportional Integral Derivative (PID). Namun, terdapat permasalahan utama dalam kontrol PID yaitu penyesuaian parameter yang harus cepat dan akurat untuk mendapatkan kontrol yang diinginkan. Untuk mengatasi permasalahan tersebut dikembangkan suatu metode untuk optimisasi nilai parameter yang dikenal dengan Particle Swarm Optimization (PSO). Berdasarkan penelitian yang dilakukan, nilai kontrol PID didapat dengan metode Ziegler-Nichols dan metode algoritma PSO. Metode Ziegler-Nichols mendapat nilai $K_p = 4,842$, $K_i = 8,327$, dan $K_d = 0,704$. Nilai rise time 0,162 detik dan settling time yang didapat sebesar 1,2 detik. Sistem juga mencapai keadaan steady state pada waktu 2 detik. Metode PSO mendapat nilai $K_p = 7,0339$, $K_i = 20,0074$, dan $K_d = 0,6734$. Nilai rise time 0,0925 detik dan settling time yang didapat 0,148 detik. Sistem juga mencapai keadaan steady state pada waktu 0,7 detik. Hasil ini menunjukkan bahwa metode PSO lebih baik dibanding metode Ziegler-Nichols.

Kata Kunci: Pengendali Stir, PID, Particle Swarm Optimization, Ziegler-Nichols

ABSTRACT: Progress in the field of technology has led to the increasing development of autonomous vehicle technology. One of the important components in an autonomous vehicle is the control steering to manage its movement. During the steering control process, there are many disturbances and uncertainties, so various methods of steering control have been widely studied by researchers. One of the control methods is the Proportional Integral Derivative (PID) control. However, the main problem in PID control is the adjustment of parameters that must be fast and accurate to achieve the desired control. To address this problem, a method for optimizing parameter values was developed, known as Particle Swarm Optimization (PSO). Based on the research conducted, PID control values were compared with the Ziegler-Nichols method and the PSO algorithm method. The Ziegler-Nichols method yielded $K_p = 4.842$, $K_i = 8.327$, $K_d = 0.704$, 0.162 seconds for the rise time, and the settling time was 1.2 seconds. The system also reached steady state at 2 seconds. The PSO method obtained $K_p = 7.0339$, $K_i = 20.0074$, and $K_d = 0.6734$; the rise time value was 0.0925 seconds, and the settling time was 0.148 seconds, steady state at 0.7 seconds. These results show that the PSO method is better compared to the Ziegler-Nichols method.

Keywords: Control Steering, PID, Particle Swarm Optimization, Ziegler-Nichols

1 Pendahuluan

Electric Vehicle (EV) telah muncul pada akhir abad 18 dan popular sampai pada tahun 1918, penggunaan *Electric Vehicles* terus berkurang disebabkan perkembangan transportasi menggunakan *Internal Combustion Engine*

(ICE) [1]. Namun, perkembangan signifikan pada elektronik dan juga teknologi *mikrokontroller* membuat EV dapat bersaing dengan ICE. Alasan lingkungan juga di pertimbangkan karena ICE menyebabkan polusi udara serta memiliki keterbatasan pada sumber bahan bakar sehingga membuat orang-orang kembali melihat EV

sebagai transportasi alternatif ramah lingkungan, biaya energi rendah dan energi terbarukan [2]. *Electric Vehicle* memiliki potensi untuk meningkatkan efisiensi energi, mengurangi gas rumah kaca, dan mengurangi ketergantungan energi fosil yang terbatas [3].

Hal lain yang mendukung terus berkembangnya *Electric Vehicle* adalah penelitian yang terus meningkat di bidang sains dan teknologi dalam beberapa tahun terakhir, khususnya teknologi penginderaan dan komputasi bersama berdampak pada penelitian ke arah *Autonomous Vehicle* [4]. Secara umum, *Autonomous Vehicles* merupakan suatu teknologi pada kendaraan yang memungkinkan untuk mengendarai dirinya sendiri secara otomatis dengan menggunakan kecerdasan buatan. Teknologi-teknologi dasar yang dimiliki oleh *Autonomous Vehicles* terdiri dari empat teknologi dasar yaitu *modeling*, *motion control*, *map building*, dan *path planning* [5]. Komponen penting pada *motion control* adalah sistem *Steer-by-Wire* yang menghilangkan hubungan konvensional antara kemudi mobil dengan komponen mekanis digantikan oleh modul kontrol yang menerima input dari sensor untuk membelokkan roda depan menggunakan tenaga listrik [6], [7].

Steer-by-Wire membuat kemudi menjadi lebih baik dengan menyesuaikan sudut kemudi roda depan secara *real-time* melalui kontrol unit elektronik sehingga mengurangi beban pengemudi dan meningkatkan stabilitas serta keamanan pengemudi [8]. Dalam proses pengontrolan kemudi terdapat banyak gangguan dan ketidakpastian sehingga metode kontrol tradisional berdasarkan model tidak efektif [9]. Berbagai metode dalam pengontrolan kemudi telah banyak dilakukan oleh peneliti dan terus berkembang menjadi bidang riset yang populer. Salah satu metode pengontrolan yang paling populer saat ini adalah dengan metode pengontrolan *Proportional Integral Derivative* (PID) [10]. Kontrol *Proportional Integral Derivative* (PID) banyak digunakan dalam kontrol proses, kontrol motor, kontrol penerbangan, serta instrumentasi pada industri meskipun teori-teori kontrol telah banyak dikembangkan. Alasan utama luasnya penggunaan PID dikarenakan kehandalan dalam pengontrolan, kinerja yang kuat, memberikan solusi yang efektif dan strukturnya yang sederhana sehingga mudah dipahami dan diimplementasikan [11], [12].

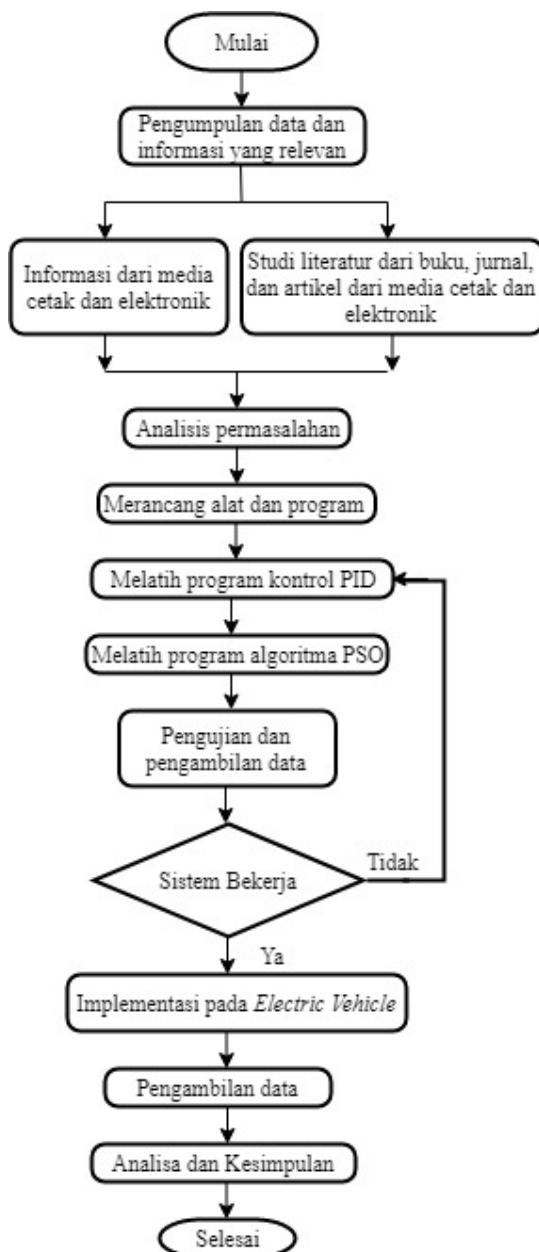
Kontrol PID beroperasi secara langsung pada sinyal *error* yang merupakan respon dari perbedaan antara *output* yang diinginkan dan *output* aktual. Terdapat permasalahan utama dalam kontrol PID yaitu penyesuaian parameter yang harus cepat dan akurat untuk mendapatkan kontrol

yang diinginkan [13]. Penelitian telah banyak dikembangkan untuk mendapatkan nilai parameter PID yang optimal dengan berbagai metode baik metode konvensional maupun metode dengan kecerdasan buatan. Metode konvensional *Ziegler-Nichols* banyak digunakan untuk menentukan nilai parameter PID, namun nilai parameter yang diberikan masih belum optimal, menghasilkan respon dengan tingkat *osilasi* dan *overshoot* yang tinggi [13], [14]. Metode konvensional lainnya *Cohen-Coon* juga menghasilkan respon dengan tingkat *osilasi* dan *overshoot* yang besar [15]. Metode konvensional *Trial-Error* juga memiliki kelemahan dalam penyesuaian parameternya karena membutuhkan waktu tuning yang lama dan tingkat akurasi kontrol rendah [16]. Mengatasi masalah kurang efektifnya metode konvensional tersebut dalam menentukan nilai parameter, dikembangkan metode dengan menggunakan *Artificial Intelligence* (AI) salah satunya pengontrolan logika *Fuzzy*. Namun, pengontrolan logika *Fuzzy* masih menghasilkan *osilasi* yang cukup besar [17].

Dalam rangka mengatasi permasalahan yang ditimbulkan oleh metode-metode sebelumnya, dikembangkan suatu metode untuk optimisasi nilai parameter yang dikenal dengan *Particle Swarm Optimization* (PSO). Algoritma *Particle Swarm Optimization* merupakan sebuah algoritma yang diadaptasi dari perilaku sekelompok burung dalam migrasi dan mencari makanan [18].

2 Metode

Penelitian yang dilakukan untuk menguji performa *Automatic Control Steering* menggunakan pengontrol *Proportional Integral Derivative* (PID). Algoritma yang digunakan untuk menentukan parameter PID adalah algoritma *Particle Swarm Optimization* (PSO). Objek penelitian yang digunakan untuk menguji performa *Automatic Control Steering* adalah *Electric Vehicle* yang berada pada Laboratorium Riset Kendali dan Robotika, Jurusan Teknik Elektro Fakultas Teknik Universitas Sriwijaya. Adapun tahapan penelitian yang dilakukan untuk menguji performa *Automatic Control Steering* adalah sebagai berikut :

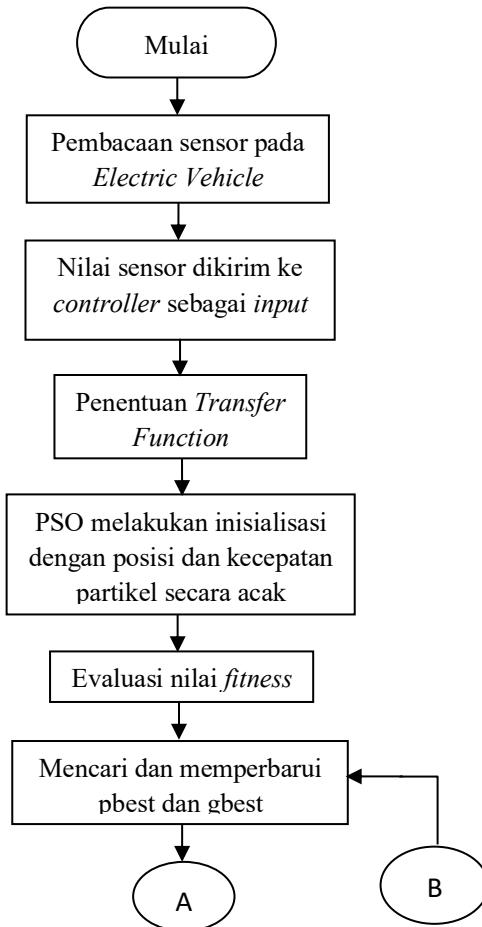


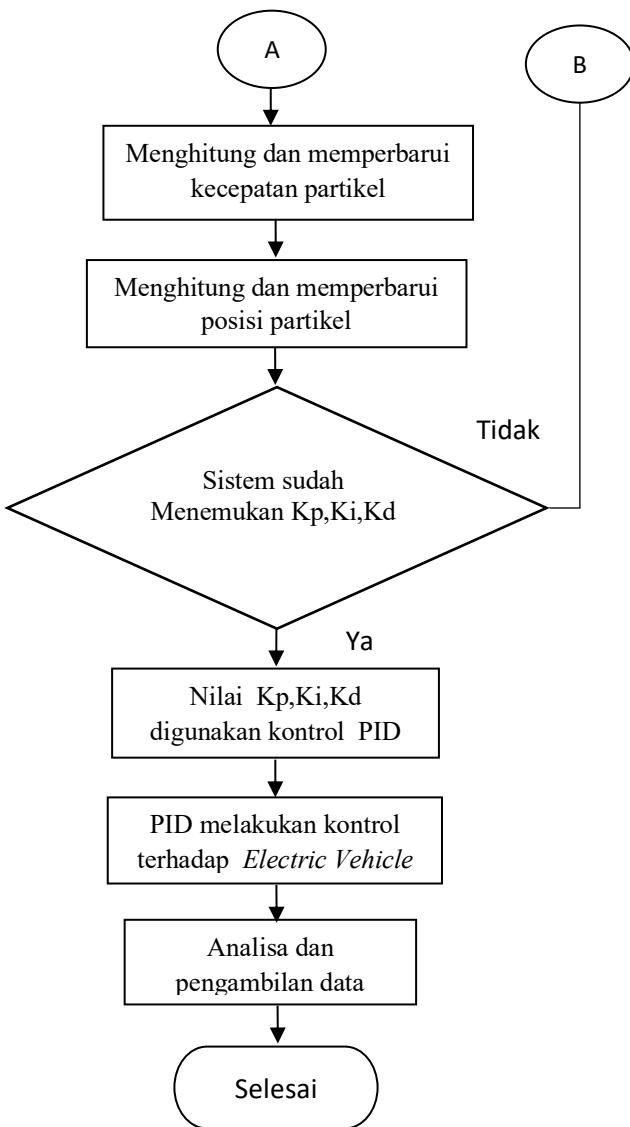
Gambar 1. Diagram alir penelitian

Penelitian diawali dengan pengumpulan data dan informasi yang relevan dari berbagai macam sumber baik media cetak seperti buku, jurnal, ataupun artikel maupun media elektronik. Setelah informasi yang dikumpulkan dirasa cukup, tahap selanjutnya dengan merancang alat serta program untuk menemukan solusi dari permasalahan yang timbul. Pengontrol PID dilakukan evaluasi dengan penentuan parameterternya menggunakan algoritma PSO. Apabila sistem bekerja maka pengontrol akan langsung di

implementasikan pada Elecric Vehicle, namun apabila sistem tidak bekerja, maka proses perancangan pengontrol PID ditinjau kembali. Untuk mengetahui tingkat keberhasilan dari pengontrol PID dengan penentuan parameter menggunakan algoritma PSO, maka dilakukan pengambilan data dan dianalisa untuk mendapatkan kesimpulan terhadap penelitian.

Dalam perancangan sistem pada penelitian ini terdapat dua sistem perancangan yaitu perancangan perangkat lunak dan perancangan perangkat keras. Untuk perancangan perangkat lunak digunakan algoritma menentukan parameter PID adalah algoritma *Particle Swarm Optimization* (PSO). Parameter yang digunakan untuk pembuatan fungsi alih adalah jarak sebagai input yang merupakan hasil pembacaan sensor jarak, kemudian output-nya berupa sudut dari ban terhadap sumbu kendaraan. Gambar 2 merupakan diagram alir dari *Automatic Control Steering* dengan pengontrol PID yang akan dilakukan oleh peneliti.





Gambar 2. Flowchart perancangan penelitian

Sistem dirancang untuk dapat memenuhi pengimplementasian dari *Particle Swarm Optimization* (PSO) untuk menentukan parameter dari *Proportional Integral Derivative* (PID) sebagai kontrol terhadap Automatic Steering. Penelitian diawali dengan pembacaan sensor pada *Electric Vehicle* sebagai input terhadap controller. Didalam controller terjadi proses komputasi oleh komputer untuk ditentukan Transfer Function dari sistem. *Transfer Function* yang didapat selanjutnya akan diproses oleh algoritma PSO untuk melakukan inisialisasi posisi serta kecepatan partikel. Setelah itu, PSO akan melakukan evaluasi nilai fitness serta mencari dan memperbarui pbest dan gbest untuk selanjutnya dilakukan

perhitungan terhadap kecepatan dan posisi partikel. Apabila PSO telah menemukan nilai parameter dari Kp, Ki, dan Kd yang optimal, maka nilai tersebut akan digunakan oleh kontrol PID. Namun apabila nilai parameter belum optimal, maka PSO akan melakukan evaluasi terhadap nilai fitness dan proses kembali berulang. Setelah mendapatkan parameter yang optimal, PID akan melakukan kontrol terhadap Stir untuk selanjutnya dilakukan pengambilan data dan diambil kesimpulan.

Particle Swarm Optimization (PSO) adalah salah satu metode optimisasi berbasis populasi yang terbarukan dan tidak menggunakan operasi penyaringan karena anggota dari seluruh populasi dipertahankan melalui prosedur pencarian. Algoritma *Particle Swarm Optimization* (PSO) diperkenalkan oleh Kennedy dan Eberhard pada tahun 1995 yang terinspirasi dari perilaku kawan burung dalam mencari makanan atau berimigrasi [18]. Algoritma *Particle Swarm Optimization* (PSO) meniru perilaku sosial organisme dari tindakan individu-individu lain dalam suatu kelompok. Setiap individu atau partikel berperilaku dengan cara memanfaatkan kecerdasan sendiri secara terdistribusi serta dipengaruhi perilaku kelompok kolektifnya. Dengan kondisi ini, ketika suatu individu atau partikel menemukan jalan terbaik untuk menuju ke sumber makanan, maka siswa kelompok yang lain akan mengikuti jalan tersebut. Metode optimasi *Swarm Optimization* memiliki konteks optimisasi *multivariabel* dimana setiap partikel diasumsikan mempunyai kecepatan tertentu dan memiliki posisi yang terletak di suatu lokasi yang acak dalam ruang. Setiap partikel memiliki karakteristik posisi dan kecepatan dimana partikel bergerak dalam ruang tertentu dan mengingat posisi terbaik yang pernah ditemukan atau nilai fungsi objektif. Setiap partikel memberikan informasi tentang posisi terbaiknya kepada partikel yang lain serta menyesuaikan kecepatan dan posisi berdasarkan informasi yang didapat [19].

Algoritma PSO terdapat dua varian yang dikembangkan, yaitu partikel dengan lingkungan lokal dan partikel dengan lingkungan global. Menurut lingkungan global, setiap partikel bergerak menuju posisi terbaik sebelumnya dan menuju partikel terbaik di seluruh kawannya disebut dengan model gbest. Sedangkan menurut varian lokal, setiap partikel bergerak menuju posisi terbaik sebelumnya dan menuju partikel terbaik dalam lingkungan yang terbatas. Menurut Sevkli dan Guner [20], beberapa langkah pada algoritma *Particle Swarm Optimization* (PSO) dalam melakukan tuning PID :

1. Inisialisasi nilai parameter awal dari *Particle Swarm Optimization* (PSO) seperti kecepatas dan posisi.

2. Menentukan nilai awal dari Kp, Ki, dan Kd dengan menggunakan metode seperti metode Ziegler Nichols.
3. Tetapkan banyaknya partikel *swarm* pada PSO (n), dimana tiap partikel masing-masing memiliki tiga nilai (Kp, Ki, dan Kd)
4. Tetapkan nilai dari posisi awal tiap *swarm* untuk Kp, Ki, dan Kd.
5. Lakukan *test* dari setiap *swarm* yang dipilih (Kp, Ki, dan Kd) dan lihat respon yang dihasilkan.
6. Lihat nilai *Rise Time*, *Settling Time*, dan *Overshoot*, kemudian cari partikel yang menghasilkan nilai ITAE yang paling kecil dan menjadi *Gbest* sebagai acuan basi partikel lain pada iterasi berikutnya.
7. Jika iterasi telah maksimum atau algoritma mencapai posisi terbaik maka algoritma akan berhenti, namun jika tidak maka akan terus melakukan iterasi.

Secara umum algoritma PSO dapat direpresentasikan dengan rumus dibawah ini:

$$v_i^{k+1} = wv_i^k + c_1 * \text{Rand}() * (\text{Pbest}_k^i - x_i^k) + c_2 * \text{Rand}() * (Gbest^k - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

dimana,

v_i^k : kecepatan partikel ke-i pada iterasi yang ke-k.

x_i^k : posisi partikel ke-i pada iterasi yang ke-k.

w : faktor bobot inersia.

c_1, c_2 : konstanta akselerasi.

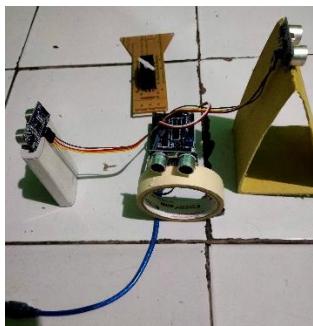
Rand : bilangan random dari 0 sampai 1.

Pbest : posisi terbaik partikel sebelum partikel ke -i.

Gbest : partikel terbaik diantara semua partikel dalam *swarm*.

3 Hasil dan Pembahasan

Perancangan alat perlu dilakukan agar dapat menguji nilai dari parameter PID yang telah didapatkan. Peralatan dibuat tersusun atas komponen-komponen seperti mikrokontroler, motor servo, dan sensor jarak yang dapat dilihat pada gambar 3.



Gambar 3. Hardware In The Loop (HIL)

Hardware In The Loop (HIL) yang akan digunakan untuk pengambilan data menggunakan tiga buah sensor ultrasonik dan sebuah motor servo dengan mengukur jarak objek benda didepannya. Nilai parameter kontroler PID dapat ditentukan dengan mencari nilai *transfer function* untuk motor servo terlebih dahulu. Berdasarkan penelitian yang dilakukan oleh Fiktor Sihombing [21] dalam melakukan pengujian fisik untuk *plant* motor servo dengan $K_p = 20$, diperoleh maksimum *overshoot* (M_p) adalah 0,0292 pada *peak time* (t_p) = 0,875 detik. Nilai-nilai yang didapat dimasukkan kedalam rumus sehingga didapat nilai parameter faktor redaman (ζ) :

$$\begin{aligned} M_p &= e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi} \\ e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi} &= 0,0292 \\ -\frac{\pi\zeta}{\sqrt{1-\zeta^2}} \ln e &= \ln 0,0292 \\ -\frac{\pi\zeta}{\sqrt{1-\zeta^2}} &= \ln 0,0292 \text{ dimana } \ln e = 1 \\ -\frac{3,14\zeta}{\sqrt{1-\zeta^2}} &= -3,533 \\ \zeta &= 0,747 \end{aligned}$$

Kemudian dicari perhitungan Frekuensi Alamiah Tidak Teredam (ω_n) :

$$\begin{aligned} t_p &= \frac{\pi}{\omega_d} \\ \omega_d &= 3,588 \\ \omega_n &= \frac{\omega_d}{\sqrt{1-\zeta^2}} \\ &= \frac{3,588}{\sqrt{1-(0,747)^2}} \\ \omega_n &= 5,397 \text{ rad/sec} \end{aligned}$$

Selanjutnya dicari perhitungan *Transfer Function* Untuk Harga $K_p = 20$:

$$\begin{aligned} \frac{C_{(s)}}{R_{(s)}} &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ \frac{C_{(s)}}{R_{(s)}} &= \frac{29,2}{s^2 + 8,07s + 29,2} \end{aligned}$$

Transfer function yang didapat sebelumnya akan dihitung menggunakan metode kedua dari metode Ziegler-Nichols, yaitu metode osilasi. Metode ini dilakukan pada *plant* tertutup yang disusun seri dengan kontroler PID saat parameter integrator disetel tak berhingga ($T_i = \infty$) dan parameter diferensial disetel nol ($T_d = 0$) [22].

$$\frac{C_{(s)}}{R_{(s)}} = \frac{29,2}{s^2 + 8,07s + 29,2}$$

Input unit step

$$\frac{C(s)}{s} = \frac{29,2}{s^2 + 8,07s + 29,2}$$

$$C(s) = \frac{29,2}{s^3 + 8,07s^2 + 29,2s}$$

Berdasarkan teori dari metode kedua Ziegler-Nichols maka didapat:

$$C(s) = \frac{K_{CR}}{\frac{1}{29,2}s^3 + \frac{8,07}{29,2}s^2 + s + K_{CR}}$$

Kemudian nilai K_{CR} dapat ditentukan dari tabel 1.

Tabel 1. Judul tabel ditempatkan rata kiri dan ditempatkan

s^3	$\frac{1}{29,2}$	1
s^2	$\frac{8,07}{29,2}$	K_{CR}
s^1	$\frac{8,07 - K_{CR}}{8,07}$	0
s^0	K_{CR}	0

Berdasarkan tabel dapat ditentukan nilai K_{CR}

$$\frac{8,07 - K_{CR}}{8,07} = 0$$

$$K_{CR} = 8,07$$

$$K_p = 0,6 K_{CR}$$

$$K_p = 4,842$$

Berdasarkan nilai K_p yang didapat, maka dapat dicari nilai ω :

$$\frac{1}{29,2}s^3 + \frac{8,07}{29,2}s^2 + s + K_{CR} = 0$$

Substitusikan nilai s menjadi $j\omega$:

$$\frac{1}{29,2}(j\omega)^3 + \frac{8,07}{29,2}(j\omega)^2 + (j\omega) + 8,07 = 0$$

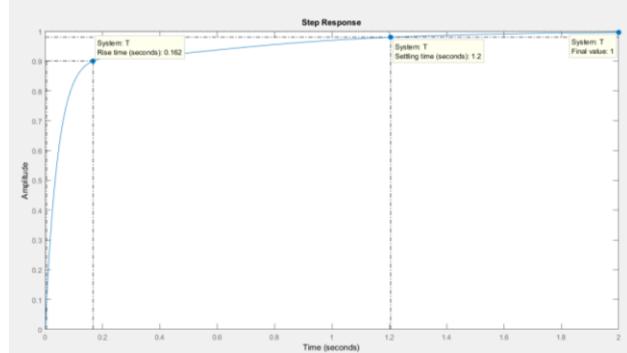
$$-\frac{1}{29,2}j\omega^3 - \frac{8,07}{29,2}\omega^2 + j\omega + 8,07 = 0$$

$$\omega = 5,4$$

Setelah nilai ω didapat, maka kita dapat menentukan nilai K_p , K_i , dan K_d . Jadi didapat nilai parameter PIDnya, yaitu $K_p=4,842$, $K_i= 8,327$, dan $K_d=0,704$. Dengan memasukkan parameter K_p , K_i , dan K_d yang didapat kedalam aplikasi MATLAB, respon sistem terhadap perubahan parameter PID dapat dilihat pada gambar 4.

Pada gambar 4 rise time yang diperlukan oleh sistem sebesar 0,162 detik dan settling time yang didapat sebesar

1,2 detik. Sistem juga mencapai keadaan steady state pada waktu 2 detik. Input unit step dipilih karena sistem lebih stabil dan dapat dicari nilai parameter PID. Pemilihan input unit impuls tidak bisa dilakukan disebabkan orde dari persamaan genap, sehingga nilai ω tidak bisa didapat yang mengakibatkan nilai parameter PID tidak bisa didapatkan. Pemilihan input unit ramp tidak bisa dilakukan disebabkan orde dari persamaan genap dan nilai kestabilan routhnya memberikan nilai $K_p = 0$ yang mengakibatkan nilai parameter PID tidak bisa didapat.

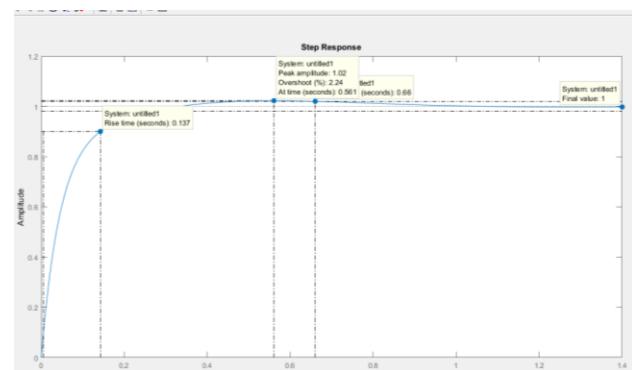


Gambar 4. Respon sistem terhadap perubahan parameter PID dengan metode Ziegler-Nichols

Kemudian pada bagian ini, transfer function akan dihitung menggunakan algoritma Particle Swarm Optimization (PSO) untuk mendapatkan parameter PID yang sesuai dengan mengubah jumlah partikel PSO meningkat dari 20, 60, dan 100 partikel.

3.1 Pelatihan 20 artikel

Training parameter PID dengan jumlah partikel 20 dan jumlah iterasi maksimal 40, didapatkan nilai $K_p = 4,1987$, $K_i = 20,0626$, dan $K_d = 0,6768$. Dengan memasukkan parameter K_p , K_i , dan K_d didapat kedalam aplikasi MATLAB, respon sistem terhadap perubahan parameter PID dapat dilihat pada gambar 5.

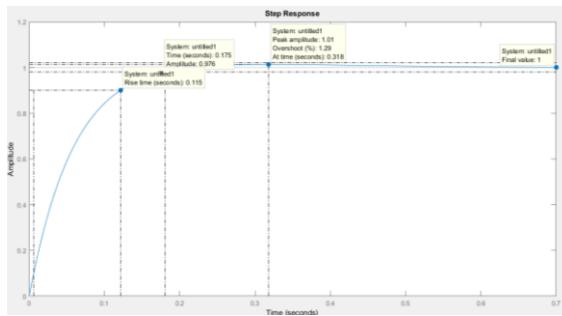


Gambar 5. Respon sistem terhadap perubahan parameter PID dengan metode Particle Swarm Optimization dengan 20 partikel

Pada gambar 5 diatas rise time yang diperlukan oleh sistem sebesar 0,137 detik dan settling time yang didapat sebesar 0,66 detik. Sistem juga mencapai keadaan steady state pada waktu 1 detik.

3.2 Pelatihan 60 Partikel

Pelatihan parameter PID dengan jumlah partikel 60 dan jumlah iterasi maksimal 40, didapatkan nilai $K_p = 5,1987$, $K_i = 20,0492$, dan $K_d = 0,6021$. Dengan memasukkan parameter K_p , K_i , dan K_d didapat kedalam aplikasi MATLAB, respon sistem terhadap perubahan parameter PID dapat dilihat pada gambar 6.

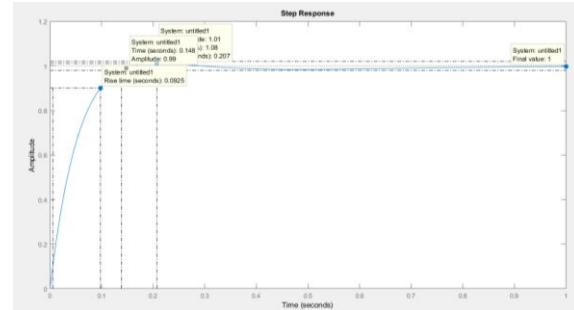


Gambar 6. Respon sistem terhadap perubahan parameter PID dengan metode Particle Swarm Optimization dengan 60 partikel

Pada gambar 4.4 diatas rise time yang diperlukan oleh sistem sebesar 0,115 detik dan settling time yang didapat sebesar 0,175 detik. Sistem juga mencapai keadaan steady state pada waktu 0,7 detik.

3.3 Pelatihan 100 Partikel

Training parameter PID dengan jumlah partikel 100 dan jumlah iterasi maksimal 40, didapatkan nilai $K_p = 7,0339$, $K_i = 20,0074$, dan $K_d = 0,6734$. Dengan memasukkan parameter K_p , K_i , dan K_d didapat kedalam aplikasi MATLAB, respon sistem terhadap perubahan parameter PID dapat dilihat pada gambar 7.



Gambar 4.5 Respon sistem terhadap perubahan parameter PID dengan metode Particle Swarm Optimization dengan 100 partikel.

Pada gambar 4.5 diatas rise time yang diperlukan oleh sistem sebesar 0,0925 detik dan settling time yang didapat sebesar 0,148 detik. Sistem juga mencapai keadaan steady state pada waktu 0,7 detik.

3.4 Pengujian Sistem

Pengujian sistem dilakukan dengan menguji respon sistem menggunakan parameter PID yang telah didapatkan pada alat secara langsung. Input sistem berupa sensor jarak dengan objek atau rintangan yang akan diletakkan pada jarak dan posisi berubah-ubah untuk mendapatkan data putar setir terhadap objek seperti yang terlihat pada tabel 2. Tabel 1. Posisi Objek Terhadap Sensor Jarak

No	Sensor Ultrasonik Sebelah Kiri	Sensor Ultrasonik Tengah	Sensor Ultrasonik Sebelah Kanan
1	>100 cm	>100 cm	>100 cm
2	>35 cm	>35 cm	>35 cm
3	<35 cm	<35 cm	<35 cm
4	<10 cm	<10 cm	<10 cm
5	>100 cm	>35 cm	>35 cm
6	>100 cm	<35 cm	<35 cm
7	>100 cm	<10 cm	<10 cm
8	>35 cm	>35 cm	>100 cm
9	<35 cm	<35 cm	>100 cm
10	<10 cm	<10 cm	>100 cm

Setelah posisi objek ditentukan, maka pengujian akan dilakukan secara berurutan dimulai dari pengujian yang pertama dengan kondisi sensor kanan >100 cm, sensor tengah >100 cm, dan sensor kiri >100 cm. Hasil yang didapat dari pengujian kondisi pertama dapat dilihat pada gambar 8.

```

pid = 90.00
108 cm
144 cm
116 cm
pid = 90.00
103 cm
280 cm
110 cm
pid = 90.00
109 cm
279 cm
113 cm
pid = 90.00
  
```

Gambar 8. Hasil Pengujian Kondisi Pertama

Pada gambar 8 dapat dilihat sudut motor servo yang didapat sebesar 90 derajat dan merupakan kondisi lurus. Hal ini dikarenakan setir tidak perlu merespon karena objek jauh dari sensor yang menjadi input sistem. Pengujian kedua dengan kondisi sensor kanan >35 cm, sensor tengah >35 cm, dan sensor kiri >35 cm didapat hasil seperti yang terlihat pada gambar 9 dibawah.

```

39 cm
49 cm
45 cm
pid = 68.19
39 cm
49 cm
45 cm
pid = 68.52
39 cm
48 cm
46 cm
pid = 68.52
39 cm
49 cm
45 cm
pid = 68.52
  
```

Gambar 9. Hasil Pengujian Kondisi Kedua

Pada gambar 9 dapat dilihat sudut motor servo yang didapat berada dikisaran 68,52 derajat dan kondisinya sedikit berbelok kekanan. Pengujian ketiga dengan kondisi sensor kanan <35 cm, sensor tengah <35 cm, dan sensor kiri <35 cm didapat hasil seperti yang terlihat pada gambar 10 dibawah.

```

28 cm
30 cm
pid = 98.76
25 cm
28 cm
30 cm
pid = 101.61
25 cm
28 cm
30 cm
pid = 104.13
26 cm
28 cm
30 cm
pid = 103.80
  
```

Gambar 10. Hasil Pengujian Kondisi Ketiga

Pada gambar 10. dapat dilihat sudut motor servo yang didapat berada dikisaran 104,13 derajat dan kondisinya sedikit berbelok kekiri.

4 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat terlihat bahwa tunning PID menggunakan metode algoritma Particle Swarm Optimization (PSO) menghasilkan rise time sebesar 0,0925 detik dan settling time sebesar 0,148 detik serta keadaan steady state pada waktu 0,7 detik. Hasil ini lebih baik dibandingkan dengan metode Ziegler-Nichols yang menghasilkan rise time sebesar 0,162 detik dan settling time yang didapat sebesar 1,2 detik serta keadaan steady state pada waktu 2 detik.

Daftar Pustaka

- [1] B. Bates, “Electric Vehicles - A decade of Transition,” *SAE/PT-40*, 1992.
- [2] C. C. Chan, “An Overview of Electric Vehicle Technology,” *IEEE Log Number 9211745*, vol. 81, no. 9, 1993.
- [3] “Environmental Assessment of Plug-in Hybrid Electric Vehicles - Volume 1: Nationwide Greenhouse Gas Emissions,” Electric Power Research Institute.”
- [4] Y. Cheng, D. Wu, X. Wu, and D. Zhang, “Implementation of an Arduino Obstacle Avoidance Car for Automatic Drawing a Path Map,” vol. 0958, pp. 13–16, 2019.
- [5] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, “Overview of environment perception for intelligent vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2584–2601, 2017.
- [6] B. Avek, “Modeling and Control of a Superimposed Steering System,” *Thesis, Master Sci. Sch. Electr. Comput. Eng. Georg. Inst. Technol.*, 2004.

- [7] E. Hansen, "Perencanaan prototipe steer by wire pada mobil," pp. 1–3, 2013.
- [8] X. Chen, "Wire steering system simulation active steering control strategy," *J. Hubei Automot. Ind. Inst.*, 2013.
- [9] Y. Nianjiang and A. C. S. System, "Design and Simulation for Steer-by-Wire System Based on Fuzzy-PID," *2015 7th Int. Conf. Intell. Human-Machine Syst. Cybern.*, pp. 2–5, 2015, doi: 10.1109/IHMSC.2015.43.
- [10] E. Yip, "Seam Tracking FUZZY-PID control," *Weld. J.*, pp. 97–101, 2005.
- [11] B. Nagaraj, "A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO," pp. 305–313, 2010.
- [12] V. Mukherjee and S. P. Ghoshal, "Intelligent particle swarm optimized fuzzy PID controller for AVR system," *Electr. Power Syst. Res.*, vol. 77, pp. 1689–1698, 2007, doi: 10.1016/j.epsr.2006.12.004.
- [13] M. Ahmed and M. Hassan, "Tuning PID Controllers Using Artificial Intelligence Techniques Applied to Tuning PID Controllers Using Artificial Intelligence Techniques Applied To DC-Motor and AVR System," *Asian J. Eng. Technol.*, vol. 02, no. April, 2016.
- [14] A. Jalilvand, A. Kimiyaghaham, A. Ashouri, and H. Kord, "PID Controllers: Theory, Design and Tuning," *IJTPE J.*, 2011.
- [15] G. . Cohen and G. . Coon, "Theoretical investigation of retarded control," *Trans. Am. Soc. Mech. Eng.*, vol. 75, 1953.
- [16] M. Ruswandi Djalal and Rahmat, "Optimisasi Kontrol PID untuk Motor DC Magnet Permanen Menggunakan Particle Swarm Optimization," *J. TAM (Technology Accept. Model.)*, vol. 8, pp. 117–122, 2017.
- [17] A. Kaushal, N. Thakur, and D. Nagaria, "Comparison of Speed Control of DC Motor Using Fuzzy- PID and PSO-PID Technique," vol. 4, no. 6, pp. 553–558, 2014.
- [18] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. Neural Networks*, vol. 4, pp. 1942-1948., 1995.
- [19] E. Dermawan, A. Firmanto, S. N. Ambo, E. Diniardi, and A. I. Ramadhan, "DESAIN FREKUENSI KONTROL PADA HYBRID WIND-DIESEL DENGAN PID - PARTICLE SWARM OPTIMIZATION (PSO)," no. November, pp. 8–9, 2016.
- [20] M. Sevkli and A. R. Guner, "Ant Colony Optimization and Swarm Intelligence: A continuous Particle Swarm Optimization for Uncapacitated Facility Location Problem," *Lect. Notes Comput. Sci.*, vol. 4, pp. 316–323.
- [21] F. Sihombing and D. Lumbantoruan, "STUDI ANALISIS DAN REKAYASA SISTEM KENDALI MOTOR SERVO DC DI LABORATORIUM DASAR SISTEM KENDALI DENGAN MENGGUNAKAN METODA POLE PLACEMENT," 2011.
- [22] M. Sholihin, L. Tack, and M. Kean, "Tuning of PID Controller Using Particle Swarm Optimization (PSO)," *Proceeding Int. Conf. Adv. Sci.*, pp. 458-461., 2011.