

Penerapan Metode *Fuzzy Logic*-PID Sebagai Kontrol Gerak Robot Berkaki Enam (*Hexapod*)

Joko Subur¹, Suryadhi², Eki Arohman³

^{1,2,3}Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Kelautan,
Universitas Hang Tuah Surabaya

ARTICLE INFO

Article history:

DOI:

[10.30595/pspfs.v6i.858](https://doi.org/10.30595/pspfs.v6i.858)

Submitted:

August 05, 2023

Accepted:

September 29, 2023

Published:

October 13, 2023

Keywords:

Hexapod Robot, Motion Navigation, Fuzzy Logic Controller, PID Control

ABSTRACT

Robotics technology is currently developing rapidly and has become one of the sophistications that are relied upon by every country, one of which is Indonesia. One of the activities for the development of robotics technology in Indonesia is the Indonesian Robot Contest (KRI). KRI was held as an event for students' creativity in designing robots. One of the divisions contested at KRI is the Indonesian Search and Rescue Robot Contest (KRSRI) division. In the KRSRI division, it is required to make a legged robot that can navigate the motion of walking in the competition area and pass obstacles on the track. Of course, a navigation motion control system is needed that can regulate the movement of the legged robot's steps properly, so that the robot can pass the path in the arena. In this study, the answer to this problem is the application of legged robot navigation motion control using the Fuzzy Logic-PID method. The Fuzzy Logic-PID method is a combination of two methods between the Fuzzy Logic Controller method and the Proportional Integral Derivative (PID) control method. The Fuzzy method's role is to produce constant values P, I, and D based on the error value and the incoming error delta. Furthermore, the resulting constant value is used in the calculation of the PID control. From the experimental data, the Fuzzy method has been able to produce a PID constant value and the PID method can also be applied to regulate the robot's motion steps. The test results on trials using obstacles and without obstacles were successful, except for uphill obstacles with a success percentage of 60% successfully passing incline obstacles and 40% failing to pass uphill obstacles.

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



Corresponding Author:

Joko Subur

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Kelautan, Universitas Hang Tuah Surabaya
Jl. Arif Rahman Hakim No. 150 Sukolilo, Surabaya, Jawa Timur, Indonesia, 60111

Email: joko.subur@hangtuah.ac.id

1. PENDAHULUAN

Di Indonesia terdapat ajang perlombaan robotika yang diadakan oleh Pusat Prestasi Nasional – Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi yang disediakan bagi Mahasiswa untuk beradu kreatifitas dalam perlombaan bidang robotika. Pada tahun 2023 Terdapat 7 divisi perlombaan yaitu KRAI, KRSRI, KRSBI Beroda, KRSBI Humanoid, KRSTI, KRTMI dan KRBAL. Salah satu divisi yaitu Kontes Robot *Search And Rescue* (SAR) Indonesia atau KRSRI. Robot pada perlombaan divisi KRSRI pada umumnya menggunakan model robot berkaki empat (*Quadpod*) atau bisa juga robot berkaki enam (*Hexapod*). Di Universitas Hang Tuah (UHT) Surabaya saat ini

mengikuti robot divisi KRSRI menggunakan robot *Hexapod* yaitu robot berkaki enam, yang merupakan pengembangan dari robot *Quadpod* berkaki empat sebelumnya.

Kontes Robot SAR Indonesia (KRSRI) merupakan pengganti Kontes Robot Pemadam Api Indonesia (KRPAI). Untuk saat ini KRSRI ditugaskan tidak hanya meniuip lilin juga harus menyelamatkan calon korban dari potensi kebakaran ke tempat yang lebih aman, Seiring dengan nama kontes robot untuk mengedepankan fungsi SAR dengan robot berkaki, setiap jalur lintasan diberikan rintangan yang lebih menantang yang bisa dihadapi robot SAR pada wilayah bencana [1]. Oleh karena itu robot harus mampu bernavigasi melewati rintangan, tentunya juga harus mampu membuat robot sesuai dengan rintangan yang setiap tahunnya dirubah.

Robot KRSRI Universitas Hang Tuah diperlukan pengembangan dalam bernavigasi mengenai pergerakan kaki robot, pertama kali menggunakan metode *Proportional Integral Derivative (PID)* untuk bernavigasi gerak robot yang sebelumnya pada penelitian [2]. Pada penelitian tersebut robot UHT menggunakan robot berkaki empat dengan menggunakan metode *PID trial and error*, Robot dapat mengikuti dinding akan tetapi kestabilan pada robot masih kurang, robot sering tersangkut belum mampu melewati gundukan sehingga dikembangkan menjadi robot *Hexapod*, pada keandalan menyisiri dinding robot juga masih sering terbentur oleh dinding arena maka butuh penambahan metode pada sistem robot, Robot *Hexapod* memiliki manfaat lebih dalam mekaniknya untuk melalui rintangan jalan tak rata dan kasar dibandingkan robot beroda [3].

Pengembangan selanjutnya ditambahkan metode lain yaitu *Fuzzy Logic Controller*, PID diterapkan pada robot berkaki maka akan memiliki banyak kekurangan karena robot berkaki menggunakan penggerak motor servo yang berjumlah banyak sehingga memiliki respon yang lambat [4]. Pada penelitian tersebut menggunakan metode *Fuzzy-PID* pada robot *Hexapod* dengan penggabungan antara PID dan *Fuzzy Logic Controller* disaat *error* terlalu besar atau kondisi *error* yang mendadak terjadi osilasi diperlukan *Fuzzy* untuk respon robot yang lebih cepat dan tidak mengalami osilasi yang besar, akan tetapi pada penelitian tersebut belum menggunakan *auto-tunning* parameter PID menggunakan logika *Fuzzy*. Algoritma kendali PID, sangat sederhana dan mudah untuk diimplementasikan pada sistem *LTI (Linear Time Invariant)*, namun jika sistem yang dikendalikan *Non-LTI* penggunaan PID memerlukan algoritma tambahan agar dapat dicapai sistem kendali yang stabil [5].

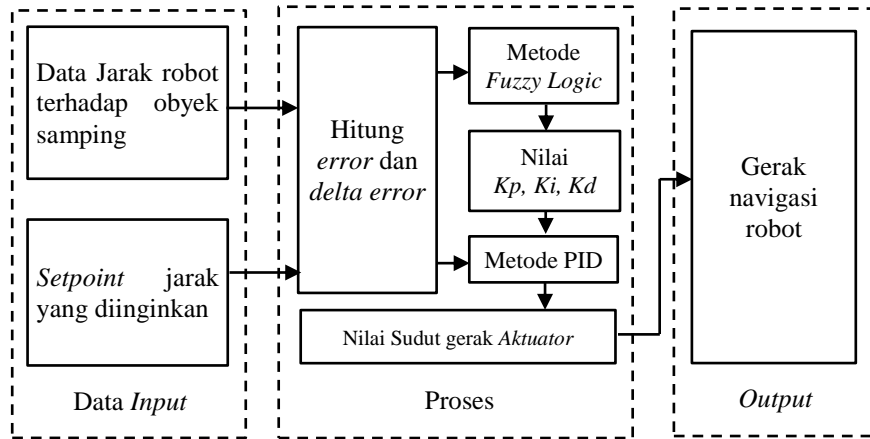
Pada nilai konstanta PID saling mempunyai kelebihan sendiri-sendiri untuk melengkapi dari sistem kontrol rumus pencarian konstanta P, I, dan D dalam satu kondisi, Penyetelan konstanta menggunakan K_p , K_i , dan K_d akan mengakibatkan reaksi dari respon yang berbeda-beda [6], Sehingga diperlukan *auto-tunning* PID yang sesuai menggunakan *Fuzzy Logic Controller* untuk menghasilkan *variable* yang sesuai pada konstanta PID pemberian nilai ini memiliki memiliki konsep dasar pengetahuan secara heuristik misalnya apabila respon memiliki *error steady state* lalu apabila pada kontroler diberi konstanta K_i maka *error steady state* akan terhapus, lalu misalnya apa bila terdapat *overshoot* maka nilai K_d harus diberi nilai yang lebih besar sehingga menambah transien pada respon [7]. Untuk memaksimalkan performansi sistem, umumnya teori *Ziegler and Nichols* sangat populer digunakan dalam menentukan parameter PID, menggantikan teori *trial and error* [8]. Pada *Fuzzy Inference System* Mamdani membutuhkan 5 langkah untuk menghasilkan output: Fuzzyfikasi, Operasi fungsi keanggotaan, Implikasi fungsi, Agregasi, dan DeFuzzyfikasi [9].

2. METODE PENELITIAN

Pada penelitian ini langkah yang disusun berdasarkan referensi dari literatur lain untuk menjawab sebuah permasalahan dari penelitian ini. terdapat beberapa tahapan pada penelitian ini yaitu rancangan diagram blok, pembuatan *hardware*, pembuatan *software* perintah program, dan ujicoba sistem yang dibuat.

2.1. Gambaran Umum Perancangan Sistem Alat

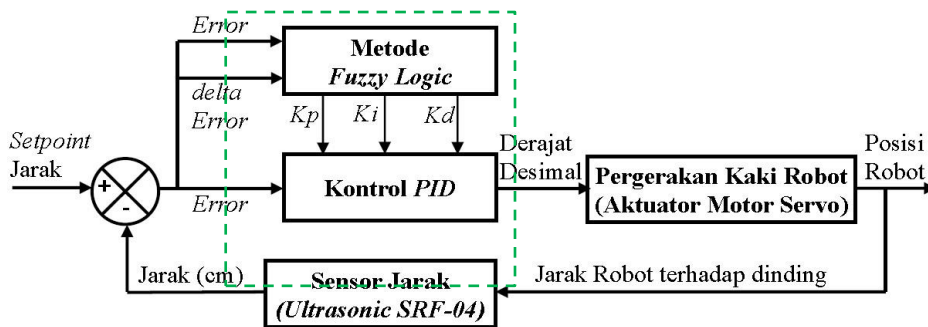
Terdapat beberapa rencana yang disajikan dalam diagram blok yang mempunyai alur untuk membedakan antara metode pada *software* dan *hardware*. Perencanaan data *input* mengacu pada data dari sensor jarak yang terpasang pada badan robot bagian samping dan penentuan nilai *setpoint* jarak yang diinginkan. Dari data input diteruskan pada bagian proses untuk dihitung nilai *error* dan *delta error*. Nilai *error* merupakan selisih antara nilai *setpoint* dengan nilai data jarak samping robot terhadap objek, sedangkan nilai *delta error* merupakan selisih *error* sekarang dengan nilai *error* sebelumnya. Dari nilai *error* dan *delta error* yang sudah didapat selanjutnya diproses melalui perhitungan metode *fuzzy logic* untuk menghasilkan (*generate*) nilai konstanta *Proportional (K_p)*, *Integral (K_i)*, dan *Derivatif (K_d)*. Tahapan proses masih berlanjut yaitu proses perhitungan metode PID, dimana pada perhitungan PID tersebut menggunakan nilai konstanta K_p , K_i , dan K_d hasil dari *generate* pada proses metode *fuzzy logic*. Dari proses tahapan perhitungan metode PID menghasilkan nilai sudut gerak aktuator untuk selanjutnya akan dikirim data pada tiap tiap aktuator pada tiap kaki robot. Pada bagian *output* adalah gerak kaki robot untuk navigasi jalan sesuai dengan data data nilai derajat pada tiap aktuator motor pada kaki kaki robot. Diagram blok sistem yang dibuat terlihat pada gambar 1 berikut:



Gambar 1. Diagram blok sistem.

2.2. Perancangan sistem kontrol Fuzzy-PID

Penelitian ini terdapat alur sistem yang mendasar dan disajikan dalam bentuk diagram blok sistem dari Fuzzy-PID yang akan diimplementasikan pada pergerakan robot.

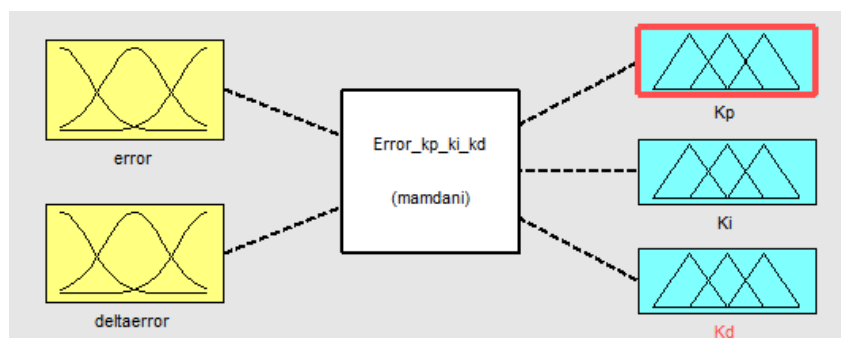


Gambar 2. Diagram blok sistem metode Fuzzy-PID.

Pada gambar 2 menunjukkan penggabungan metode fuzzy logic dan metode kontrol PID, dengan cara optimalisasi metode PID caranya penentuan nilai konstanta yang telah diatur oleh fuzzy logic. Penentuan fuzzy logic menghasilkan nilai konstanta yang sesuai dengan posisi robot dan pergerakan dari robot akan lebih teratur dan lebih cepat, karena pergerakan tidak sering mengalami osilasi maupun berbenturan dengan dinding yang akan menghasilkan efisiensi waktu yang ditempuh untuk mencapai titik tujuan tertentu pada arena perlombaan.

2.3. Perancangan sistem kontrol fuzzy logic

Penentuan input fuzzy menggunakan variable error dan variable delta error sebagai kendali perhitungan dari error dihitung berdasarkan pengukuran dikurangi dengan setpoint, dari perhitungan tersebut selisih error nya dikurangi terhadap perhitungan sebelumnya.

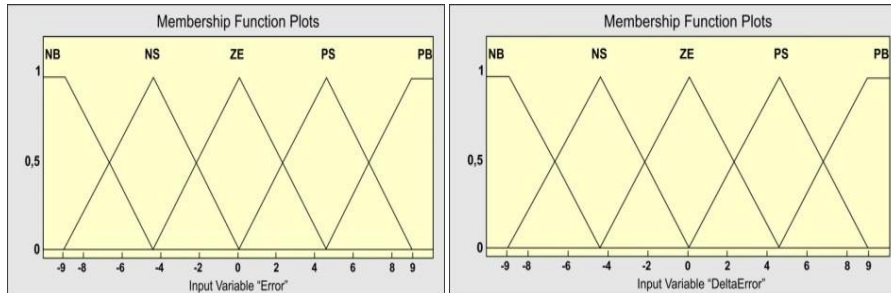


Gambar 3. Design fuzzy logic yang digunakan.

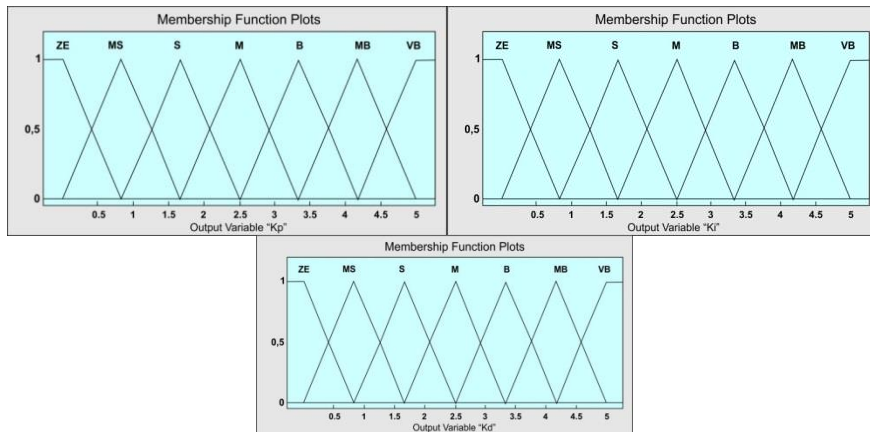
Parameter PID dari setiap parameternya terdapat konstanta sebagai faktor kali *output* yaitu K_p , K_i dan K_d yang akan digunakan sebagai konstanta kendali PID.

a. *Membership Function*

Penentuan *membership function*, *input* berdasarkan pada *setpoint* robot untuk penentuan *error* terhadap jarak dengan menghitung antara *body* robot dengan dinding sehingga dapat dihitung kondisi robot berada pada kondisi tengah jalur lintasan agar robot dapat bergerak dengan *fleksibel* dengan jarak kanan kiri agar tidak terlalu dekat dengan *body* robot.



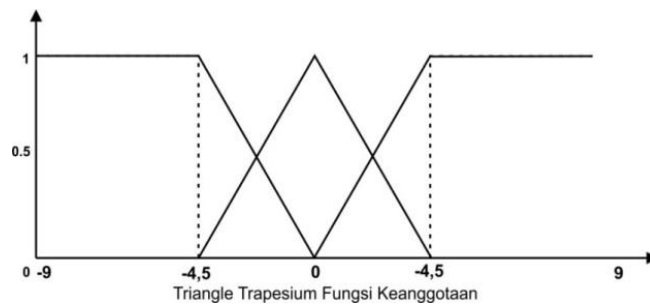
Gambar 4. *Membership function input error dan delta error.*



Gambar 5. *Membership function Output Kp, Ki, dan Kd.*

b. *Fuzzyfikasi*

Proses *fuzzyfikasi* digunakan ketika telah mendapatkan nilai *input* x yang telah masuk kedalam salah satu himpunan fungsi keanggotaan, nilai *input* tersebut bersifat tegas (*crisp*) yang kemudian dirubah menjadi *variable* data *linguistik* yang dinyatakan dalam bahasa alami dengan jenis nilai input samar (*fuzzy*) antara rentang nilai 0 - 1. Pada proses *fuzzyfikasi* ini yang menghubungkan antara *output* dengan *input* dari suatu himpunan fungsi keanggotaan dengan mengikuti aturan-aturan dasar dari *fuzzy logic*.



Gambar 6. *Triangle trapezium function*

c. *Inverensi rule base*

Penentuan *rule base* berdasarkan aksi dari masukan *error* dan *delta error* dengan keluaran dari *variable Kp*, *Ki* dan *Kd*. Pada sistematis penentuan *rule base* juga sesuai dengan fungsi tiap konstanta PID yang mempunyai fungsi berbeda-beda sehingga *rulebase* pada konstanta harus disesuaikan dengan fungsi yang terkait kontrol PID.

Tabel 1. *Rule base Kp.*

$\Delta e/e$	NB	NS	ZE	PS	PB
NB	VB	VB	VB	VB	VB
NS	B	B	B	MB	VB
ZE	ZE	ZE	MS	S	S
PS	B	B	B	MB	VB
PB	VB	VB	VB	VB	VB

Tabel 2. *Rule base Ki.*

$\Delta e/e$	NB	NS	ZE	PS	PB
NB	M	M	M	M	M
NS	S	S	S	S	S
ZE	MS	MS	ZE	MS	MS
PS	S	S	S	S	S
PB	M	M	M	M	M

Tabel 3. *Rule base Kd.*

$\Delta e/e$	NB	NS	ZE	PS	PB
NB	ZE	S	M	MB	VB
NS	S	B	MB	VB	VB
ZE	M	MB	MB	VB	VB
PS	B	VB	VB	VB	VB
PB	VB	VB	VB	VB	VB

d. *Agregasi kombinasi*

Pada tahap *agregasi* menghasilkan nilai perhitungan yang diambil dari nilai *rule* yang masih *fuzzy* (samar) kemudian dijadikan sebagai nilai *crisp* (tegas) sebagai nilai masukan dari *defuzzifikasi*, input dari kombinasi *agregasi* berupa nilai *min* dan *max* dari *output defuzzifikasi*.

e. *Defuzzifikasi*

Pada penentuan jenis *Defuzzifikasi* yang digunakan menggunakan jenis COA (*Center Of Area*), untuk mengambil titik pusat area *Fuzzy* akan digunakan sebagai pengambilan nilai *crisp* sebagai penentuan *Kp*, *Ki* dan *Kd*. sehingga secara umum dirumuskan sebagai berikut.

$$V_o = \frac{\sum_{k=1}^m V_k \mu_v(V_k)}{\sum_{k=1}^m \mu_v(V_k)} \quad (1)$$

V_o = nilai keluaran

m = tingkat kuantisasi

V_k = elemen ke – k

$\mu_v(V_k)$ = derajat keanggotaan pada fuzzy set v

V = semesta pembicaraan

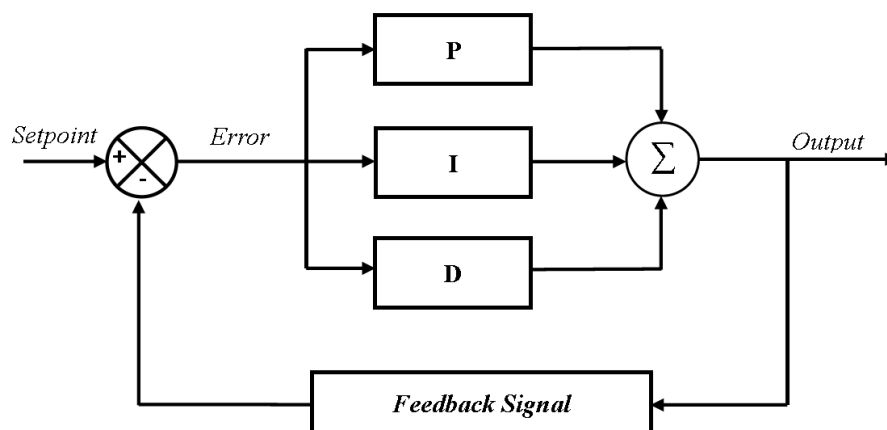
Tabel 4. Hasil *Fuzzy* untuk mendapatkan K_p , K_i dan K_d .

No	Data Input				Hasil generate		
	Jarak	kondisi	Error	Delta Error	K_p	K_i	K_d
1	24	NB	-9	-7	7,66	4,24	2,34
2	19	NS	-5	-2	4,46	2,43	6,8
3	14	ZE	0	0	1,67	0,56	8,33
4	9	PS	5	1	4,83	2,13	9,42
5	5	PB	9	0	3,33	1,67	9,75

Fuzzy logic menghitung untuk mendapatkan konstanta K_p , K_i dan K_d . Dari hasil *fuzzy logic* tersebut kemudian perhitungan digunakan sebagai konstanta pada PID untuk pengoptimalan dengan kondisi *error* dan *delta error* yang berbeda-beda kondisi *input* tersebut sudah digunakan *rule base* untuk menentukan hasil *output* dengan kondisi *input* yang berbeda.

2.4. Perancangan sistem kontrol PID

Kontrol PID mempunyai peran penting pada sistem pengendalian dengan memanfaatkan sistem *error* pada setiap parameter PID, pada kontrol *Proportional* merubah kenaikan *overshoot* untuk mencapai *setpoint* yang ditentukan, pada kontrol *Integral* berfungsi untuk menurunkan atau meredam kenaikan dari perubahan *proportional* ketika mencapai *setpoint* pada waktu mendatang, untuk kontrol *Derivative* berfungsi untuk menurunkan atau menaikkan sesuai dengan arah *setpoint* sehingga jika nilai telah mencapai kondisi stabil maka sistem akan berusaha menjaga kondisi stabil tersebut dalam waktu penerkaan lampau atau *error* sebelumnya.



Gambar 7. Diagram blok sistem PID

$$U(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2)$$

K_p = Gain Proportional, Parameter tuning

K_i = Gain Integral, Parameter tuning

K_d = Gain Derivative, Parameter tuning

e = Error = setpoint – baca jarak

t = waktu

τ = Variable integrasi, nilainya diambil dari waktu nol sampai t

Perhitungan PID kemudian diterapkan ke robot dengan cara menuliskan program dengan rumus yang sesuai dengan PID ke arduino pada rumus tersebut terdapat konstanta K_p , K_i dan K_d , sehingga pada *variable* konstanta yang nantinya akan diisi oleh tuning konstanta pada *output fuzzy logic* dan nilai dari konstanta PID dari *fuzzy logic* berubah sesuai dengan kondisi *rule base* yang dibuat.

```

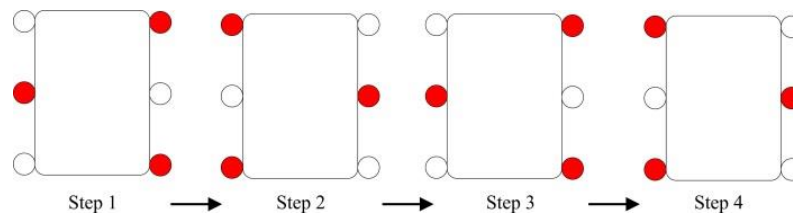
error = setpoint - nilai_sensor;
akumulasi_error += error;
delta_error = error - error_sebelumnya;
//---
P = Kp * error;
I = Ki * akumulasi_error * T;
D = Kd * (delta_error / T);
//---
hasil_PID = P + I + D;
//---
error_sebelumnya = error;

```

Gambar 8. Contoh program perhitungan PID

2.5. Pola pergerakan robot

Pola gerak robot digunakan untuk menentukan koordinasi pada gerakan robot, penelitian ini menggunakan pola gerak *tripod gait*, *Tripod gait* digunakan untuk kordinasi pola gerak robot berkaki enam dengan konfigurasi pergerakan pada kaki bergantian tiga kaki untuk tumpuan pergerakan maju.



Gambar 9. Pola gerak *tripod gait*.

Pergerakan robot *hexapod* terdiri dari enam kaki yang bergerak sesuai dengan keinginan kontrol yang dirancang. Servo pada kaki robot dapat dikontrol melalui pengalamatan pada tiap-tiap servo yang membentuk persendian kaki pada robot, tiap kaki robot berjumlah 3 servo dengan jumlah keseluruhan servo yaitu 18 servo. pola pergerakan tersebut pada tiap servo di *set* sesuai dengan alamat yang kita tentukan terlebih dahulu. Pada struktur program dari servo yang telah ditentukan alamatnya dipanggil kembali dalam sebuah *type data* dan menampung dari *variable* alamat yang dituju.

Pada pergerakan robot membutuhkan koordinasi dari pola langkah untuk diterapkan pada metode *Fuzzy-PID*, sehingga konstanta yang telah diatur tersebut digunakan untuk pengoptimalan pada parameter konstanta PID dan hasil dari perhitungan PID tersebut digunakan untuk pergerakan robot untuk menyesuaikan kondisi *error*.

```

//Gerak 1
//Untuk coxa depan kiri dan belakang kiri
derajat_akhir = derajat_tengah - (selisih - hasil_PID);
//Untuk coxa tengah kiri
derajat_akhir = derajat_tengah + (selisih + hasil_PID);
//Untuk coxa depan kanan dan belakang kanan
derajat_akhir = derajat_tengah - (selisih + hasil_PID);
//Untuk coxa tengah kanan
derajat_akhir = derajat_tengah + (selisih - hasil_PID);
//gerak 2
//Untuk coxa depan kiri dan belakang kiri
derajat_akhir = derajat_tengah + (selisih - hasil_PID);
//Untuk coxa tengah kiri
derajat_akhir = derajat_tengah - (selisih - hasil_PID);
//Untuk coxa depan kanan dan belakang kanan
derajat_akhir = derajat_tengah + (selisih + hasil_PID);
//Untuk coxa tengah kanan
derajat_akhir = derajat_tengah - (selisih + hasil_PID);

```

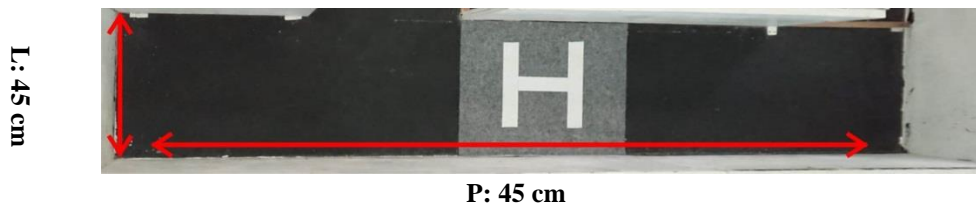
Gambar 10. Contoh program penerapan hasil PID.

3. HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai analisa data hasil pengujian alat yang dibuat, pengambilan data diambil dari hasil *input* dan *output* digunakan untuk menganalisa dari sistem yang dirancang.

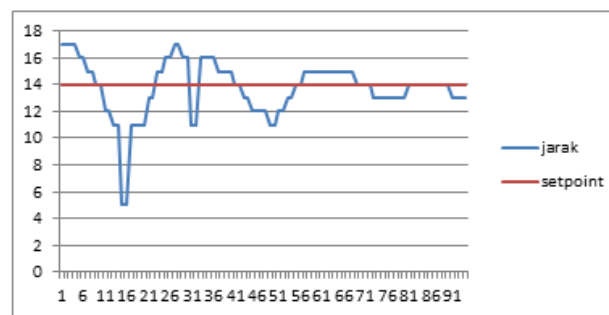
3.1. Percobaan pada lintasan lurus tanpa rintangan sejauh 200 cm

Percobaan dilakukan dengan jalan lurus berjarak 200 cm tanpa rintangan, robot bernavigasi menyisir dinding. Robot bergerak maju mempertahankan jarak dengan dinding, uji coba tanpa rintangan ini dicoba dengan kondisi yang tanpa gangguan sehingga diperoleh jika kondisi jalan yang datar karakteristik dari metode *fuzzy*-PID dapat diketahui.



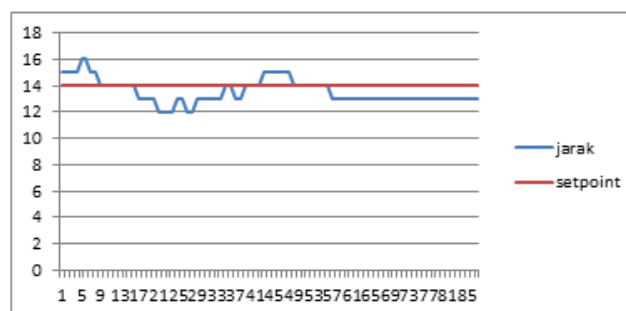
Gambar 11. Jalan lurus tanpa rintangan.

Uji coba dilakukan dengan 2 metode yaitu PID dan *fuzzy*-PID menggunakan sisir kanan atau menyisir dinding dengan sensor bagian kanan, dengan menampilkan pada setiap metode 1 model tabel untuk mengetahui data yang diambil dan 5 percobaan dengan grafik sehingga dapat diketahui perbedaan dari setiap metode yang diterapkan untuk robot berkaki enam, perbedaan dari keduanya terdapat pada penentuan konstantanya yang memiliki peranan tersendiri untuk peran kontrol PID.



Gambar 12. Grafik percobaan 1 tanpa rintangan menggunakan PID.

Pada pengamatan hasil percobaan pada gambar 12 menunjukkan terdapat 94 data pada saat robot bernavigasi dengan jarak tempuh 200 cm menempuh waktu 17:19:06.499 - 17:20:21.080 sehingga diperoleh dalam waktu yang tercatat 1 menit 15 detik, pada baca jarak robot terhadap dinding *osilasi* saat bernavigasi jarak terdekat yaitu 8 cm dan jarak terjauh 17 cm terhadap dinding.



Gambar 13. Grafik percobaan 1 tanpa rintangan menggunakan *Fuzzy*-PID.

Pada gambar 13 merupakan hasil grafik dari percobaan 1 *Fuzzy*-PID tanpa rintangan diperoleh 87 data yang diambil dari pergerakan robot dari jarak tempuh 200 cm bergerak maju dengan kondisi robot mempertahankan jarak dengan dinding. Waktu tempuh diperoleh dari 16:54:29.554 - 16:55:38.487 sehingga tercatat perolehan waktu 1 menit 9 detik. Dari beberapa percobaan diambil 5 percobaan untuk tiap metode dengan beberapa data yang masuk saat

pengiriman ke *serial monitor*, terdapat waktu pengiriman dengan begitu dapat dilihat waktu tercepat dari pergerakan robot saat bernavigasi.

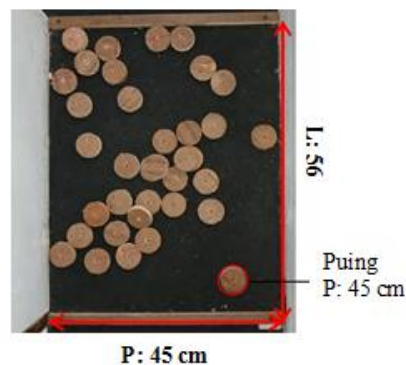
Tabel 5. Hasil uji percobaan metode PID dan *Fuzzy*-PID.

Uji Coba Jalan Lurus Tanpa Rintangan(<i>Fuzzy</i> -PID)					
No	Percobaan ke-	Metode	Jumlah data	Waktu	keterangan
1	1	PID	94	1 menit 15 detik	Berhasil
		<i>Fuzzy</i> -PID	87	1 menit 9 detik	
2	2	PID	90	1 menit 11 detik	Berhasil
		<i>Fuzzy</i> -PID	92	1 menit 10 detik	
3	3	PID	93	1 menit 14 detik	Berhasil
		<i>Fuzzy</i> -PID	92	1 menit 12 detik	
4	4	PID	94	1 menit 15 detik	Berhasil
		<i>Fuzzy</i> -PID	90	1 menit 11 detik	
5	5	PID	96	1 menit 16 detik	Berhasil
		<i>Fuzzy</i> -PID	92	1 menit 13 detik	

Dari hasil perbandingan metode PID dan *Fuzzy*-PID dari tabel 5 dan grafik pada gambar 12 dan gambar 13 menunjukkan bahwa hasil yang didapat antara jumlah data dan juga waktu tempuh pada jarak 200 cm. Metode *Fuzzy*-PID lebih sedikit untuk jumlah data dan waktu tempuh lebih singkat, sehingga jika dilihat dari grafik dan juga data pada metode *Fuzzy*-PID lebih efisien karena pengoptimalan pada konstanta PID.

3.2. Percobaan pada rintangan lurus dengan rintangan puing sejauh 150 cm

Percobaan dilakukan dengan jalan lurus berjarak 150 cm dengan ditambahkan rintangan puing, rintangan puing merupakan rintangan di kondisi area tertentu dengan ditambahkan puing-puing yang berjumlah 40 pcs, dengan panjang area rintangan puing 56 cm dan lebar rintangan puing 45 cm. Susunan dari rintangan puing tak beraturan dan dilempar dari atas dengan ketinggian 50 cm sehingga menghasilkan rintangan puing dengan diameter 5 cm yang tersusun secara *random*. Tujuan dari percobaan ini untuk menguji kestabilan gerak kontrol sistem kontrol *Fuzzy*-PID pada robot dalam menjaga jarak terhadap dinding sesuai setpoint yang ditentukan, ketika melalui jalan dengan ada rintangan puing.



Gambar 14. Rintangan puing pada lintasan jalan robot.

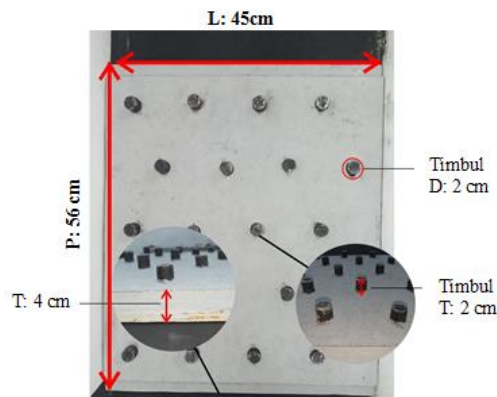
Tabel 6. Hasil uji percobaan rintangan puing menggunakan metode *Fuzzy*-PID.

Uji Coba Jalan Lurus Rintangan Puing (<i>Fuzzy</i> -PID)					
No	Percobaan ke-	Metode	Jumlah data	Waktu	keterangan
1	1	<i>Fuzzy</i> -PID	54	43 detik	Berhasil
2	2	<i>Fuzzy</i> -PID	52	41 detik	Berhasil
3	3	<i>Fuzzy</i> -PID	58	46 detik	Berhasil
4	4	<i>Fuzzy</i> -PID	50	39 detik	Berhasil
5	5	<i>Fuzzy</i> -PID	52	41 detik	Berhasil

Dari hasil Percobaan robot terhadap rintangan puing dari tabel 6 menunjukkan bahwa hasil yang didapat antara jumlah data dan juga waktu tempuh pada jarak 150 cm. Dari 5 uji coba yang dilakukan semua berhasil dengan waktu tercepat yaitu pada percobaan ke 4 dengan waktu tempuh 39 detik dan perolehan 50 data.

3.3. Percobaan pada lintasan lurus dengan rintangan timbul sejauh 150 cm

Percobaan dilakukan dengan jalan lurus sejauh 150 cm ditambahkan dengan rintangan timbul. Rintangan timbul adalah rintangan yang dibuat naik dengan ketinggian 4 cm dan diatas rintangan terdapat timbulan dengan diameter 2 cm dengan ketinggian 2 cm, *Dimensi* pada rintangan timbul memiliki panjang 56 cm dengan lebar 45 cm.



Gambar 15. Rintangan timbul pada lintasan jalan robot.

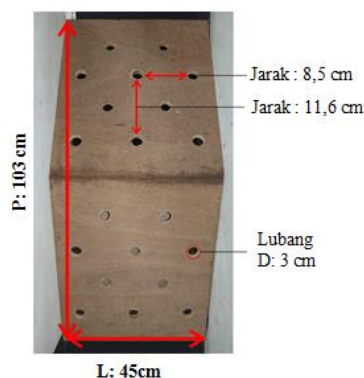
Tabel 7. Hasil uji percobaan rintangan timbul menggunakan metode *Fuzzy*-PID.

PERCOBAAN Jalan Lurus Rintangan Timbul (<i>Fuzzy</i> -PID)					
No	Percobaan ke-	Metode	Jumlah data	Waktu	keterangan
1	1	<i>Fuzzy</i> -PID	48	38 detik	Berhasil
2	2	<i>Fuzzy</i> -PID	52	41 detik	Berhasil
3	3	<i>Fuzzy</i> -PID	46	36 detik	Berhasil
4	4	<i>Fuzzy</i> -PID	50	39 detik	Berhasil
5	5	<i>Fuzzy</i> -PID	52	41 detik	Berhasil

Dari hasil percobaan robot terhadap rintangan timbul dari tabel 7 menunjukkan bahwa hasil yang didapat antara jumlah data dan juga waktu tempuh pada jarak 150 cm. Dari 5 uji coba yang dilakukan semua berhasil dengan waktu tercepat yaitu pada percobaan ke 3 dengan waktu tempuh 46 detik dan perolehan 36 data.

3.4. Percobaan pada lintasan lurus rintangan menanjak dengan kemiringan sejauh 170 cm

Percobaan pada rintangan menanjak dilakukan pada lintasan lurus 170 cm dengan selisih antara lintasan dengan rintangan menanjak yaitu 67 cm untuk gerak awal dan gerak akhir dari robot, panjang rintangan menanjak 103 cm dan lebar rintangan menanjak 45 cm dengan jarak antar lubang *vertikal* yaitu 11,6cm dan jarak *horizontal* antar lubang 8,5 cm dengan begitu robot harus melewati rintangan agar tidak terkena lubang.



Gambar 16. Rintangan jalan tanjakan dan miring pada lintasan jalan robot.

Tabel 8. Hasil uji percobaan rintangan menanjak menggunakan metode *Fuzzy*-PID.

Uji Coba Jalan Lurus Rintangan Menanjak (<i>Fuzzy</i> -PID)					
No	Percobaan ke-	Metode	Jumlah data	Waktu	keterangan
1	1	<i>Fuzzy</i> -PID	72	1 menit 1 detik	Berhasil
2	2	<i>Fuzzy</i> -PID	54	45 detik	Gagal
3	3	<i>Fuzzy</i> -PID	47	37 detik	Gagal
4	4	<i>Fuzzy</i> -PID	73	1 menit 2 detik	Berhasil
5	5	<i>Fuzzy</i> -PID	76	1 menit	Berhasil

Dari hasil percobaan robot terhadap rintangan menanjak dari tabel 8 menunjukkan bahwa hasil yang didapat antara jumlah data dan juga waktu tempuh pada jarak 150 cm, dari 5 uji coba yang dilakukan semua berhasil dengan waktu tercepat yaitu pada percobaan ke 3 dengan waktu tempuh 37 detik dan perolehan 47 data.

4. SIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa pengoptimalan metode PID dengan *fuzzy*-PID pada robot berkaki dari jarak tempuh 200 cm menunjukkan bahwa metode *fuzzy*-PID lebih stabil dibandingkan dengan kontrol PID biasa. Hal tersebut dikarenakan adanya generate secara realtime nilai K_p , K_i , dan K_d berdasarkan kondisi nilai error dan delta error. Ujicoba semua rintangan dapat diselesaikan dengan baik kecuali rintangan menanjak terbilang 60% berhasil.

DAFTAR PUSTAKA

- Puspresnas-Kemendikbud. "*Kontes Robot SAR Indonesia (KRSRI) - 2021*", Jakarta: Direktorat Kemahasiswaan, 2021.
- K. Cholid. "*Implementasi Metode Kontrol Proportional Integral Derivative (PID) Pada Gerak Robot Quadpod*". Surabaya: Universitas Hang Tuah, 2019.
- Y. Yang, L. Ding and A. Rovetta. "*Analysis of typical locomotion of a symmetric Hexapod robot*," *Robotica*, vol. 28, no. 6, p. 1, 2010.
- A. Priswanto, Sugiarto dan J. Prasjo. "*Sistem Kendali Hybrid PID-FLC Pada Studi Kasus Robot Penjejak Dinding Berkaki*". Yogyakarta: Institute Teknologi Nasional Yogyakarta, 2020.
- A. T. Kusuma, I. Agustian, F. Hadi dan A. Suandi. "*Sistem Kendali Fuzzy-Pid Pada Robot Wall Follower Ackerman Steering*". Bengkulu: Universitas Bengkulu, 2017.
- B. Raafiu. "*Implementasi Fuzzy - Pid Untuk Kendali Four-Wheeled Mobile Robotic : Studi Kasus Pada Jalan Tanjakan*". Surabaya: Institut Teknologi Sepuluh Nopember, 2018.
- H. Hibban. "*Desain Kontroler Fuzzy Pid Gain Scheduling Untuk Pengaturan Kecepatan Motor Dc Tanpa Sikat*". Surabaya: Institut Teknologi Sepuluh Nopember, 2015.
- K. Sharma and D. K. Palwalia. "*A modified PID control with adaptive Fuzzy controller applied to DC motor*". Rajasthan, India: Rajasthan Technical University, 2017.
- I. Robandi and B. Kharisma. "*Design of Interval Type-2 Fuzzy Logic Based Power System Stabilizer*". *World Academy of Science, Engineering and Technology*, 2008.