

KENDALI KECEPATAN MOTOR PADA ROBOT DENGAN EMPAT RODA *OMNI* MENGGUNAKAN METODE PID

Aryono Priyambudi¹, Beny Firman², dan Samuel Kristiyana³
^{1,2,3}Jurusan Teknik Elektro, Fakultas Teknologi Industri
Institut Sains & Teknologi AKPRIND Yogyakarta
Jl. Kalisahak No. 28 Komplek Balapan, Yogyakarta, Indonesia
priyambudi.ary@gmail.com¹, benyfirman@akprind.ac.id², yanaista@akprind.ac.id³

ABSTRACT

In this research has been developed a system to control robot which have four wheel as a base for wheeled robot contest. The type of wheel is omni-wheel, they are shaped with same distance each other. This system commonly have no straight movement and have no consistent aim-face of the robot in open loop control. The control system in this research use PID method to control the speed of each wheel. It use to make robot can move straightly and consistent of aim-face of the robot. The test on this system show that rotation speed of each wheel are equal to set point which set to PID control. With tune the parameter of PID with manual tuning method the system get 2.81% motor speed error. And get 0.66% error of aim-face of robot from forward and backward movement.

Keywords *wheeled robot, omni-wheel, PID, manual tuning*

INTISARI

Pada penelitian ini telah dikembangkan sistem penggerak robot beroda empat sebagai basis robot kontes kategori beroda. Jenis roda yang digunakan adalah roda *omni* dengan mekanisme empat roda dengan jarak yang sama. Sistem pergerakan robot dengan jenis roda ini umumnya menghasilkan arah gerak yang tidak lurus dan arah hadap yang tidak konsisten melalui pengendalian untai terbuka. Sistem pengendalian pergerakan robot pada penelitian ini menggunakan metode PID untuk mengendalikan kecepatan putaran masing-masing roda. Pengendalian ini bertujuan agar pergerakan robot lurus dengan arah hadap robot yang konsisten. Dari hasil pengujian didapatkan nilai putaran masing-masing roda setara dengan nilai set point yang diberikan pada kendali PID. Dengan parameter PID yang ditentukan melalui metode *manual tuning* didapat nilai galat kecepatan motor sebesar 2.81%. Dan galat simpangan yang dihasilkan dari pergerakan robot tersebut sebesar 0.66% dari hasil pengujian pergerakan maju dan mundur robot dengan arah hadap yang sama.

Kata Kunci: robot beroda, roda *omni*, PID, *manual tuning*

I. PENDAHULUAN

Seiring dengan kemajuan teknologi, dunia robotika juga terus berkembang pesat. Robot juga telah banyak membantu pekerjaan dan aktifitas manusia sehingga dapat lebih mudah, cepat dan efisien. Banyak robot yang telah diciptakan dengan berbagai macam bentuk maupun fungsi, dan juga dengan penggerak berupa lengan ataupun roda. Pada robot beroda dengan menggunakan roda biasa, pergerakan arah robot sangat terbatas sehingga menjadi kendala dalam fungsi robot yang dibutuhkan manuver ke berbagai arah. Dengan menggunakan roda *omni* arah pergerakan robot dapat lebih banyak.

Robot dengan menggunakan empat roda *omni* dapat bergerak lebih dari 8 arah (maju, mundur, kiri, kanan, serong kiri atas, serong kanan atas, serong kiri bawah, serong kanan bawah) tanpa merubah arah hadapnya. Dengan menggunakan empat roda *omni* maka kontrolnya juga lebih sulit dari kontrol dua roda biasa yang dipasang satu sumbu. Karakteristik tiap motor yang berbeda juga dapat mempengaruhi kecepatan tiap roda sehingga dapat mempengaruhi pergerakan robot. Dengan menggunakan kontrol biasa (open loop) pada masing-masing roda, perpindahan robot menjadi tidak lurus. Agar robot dapat bergerak lurus, robot dengan empat roda *omni* membutuhkan

sebuah kontrol yang dapat mengatur kecepatan dan arah tiap roda.

Pemilihan sistem yang kurang sesuai terhadap plant yang tidak diketahui serta ketidaktepatan serta pemodelan sistemnya. Untuk itu diperlukan metode kendali yang tepat untuk memenuhi kondisi guna mendapatkan performa yang baik dengan menambahkan PID (Proportional-Integral-Derivative) (Toto Alfian Wahyuono, 2015). PID adalah salah satu contoh metode untuk mengatur pergerakan robot. PID merupakan kontrol yang menggabungkan aksi kontrol proporsional, aksi kontrol turunan, dan aksi kontrol integral. Gabungan aksi ini mempunyai keunggulan dibandingkan dengan masing-masing dari aksi kontrol tersebut. Dengan kontrol PID ini, robot diharapkan dapat bergerak lurus sesuai dengan set point yang diberikan.

II. METODOLOGI

Dalam penelitian ini terdapat alat dan bahan serta langkah penelitian untuk membuat kontrol pada robot dengan 4 roda *omni*, diantaranya:

A. Alat dan Bahan

Adapun alat beserta spesifikasinya yang dibutuhkan pada penelitian ini ditunjukkan sebagai berikut :

1.	<i>Notebook TOSHIBA Satellite C840 Series</i> Spesifikasi : <i>Processor : Intel Core i3</i> <i>RAM : 4096 MB</i> <i>VGA Card : ATI Radeon</i> <i>Hard Disk Drive : 350 GB</i>	Menulis Program Menjalankan Program Menguji coba sistem Membuat Laporan
2.	Arduino IDE 16.9	Menulis Program
3.	Software Eagle	Membuat rangkaian PCB

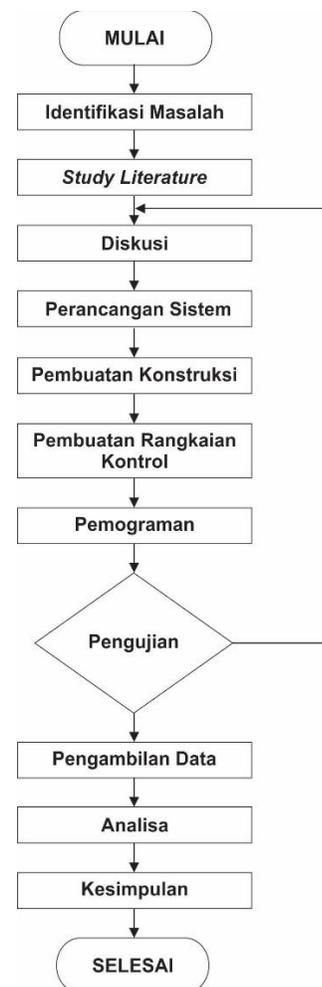
Dan bahan yang digunakan dalam penelitian ini adalah:

1.	<i>Omni Wheel</i>	Diameter 127 mm	4
2.	Motor DC	<i>DC Gear motor with encoder 45mm Planetary</i>	4

		<i>reducer, 1:19.2</i>	
3.	Mikrokontroler	Arduino Mega 2560	1
4.	<i>Driver Motor</i>	Embedded Module Series (EMS) 30 A H-Bridge	4
5.	Regulator Tegangan	UBEC 5A SBEC 15A	1 1
6.	<i>Tachometer</i>	Krisbow <i>Tachometer (KW06-563)</i>	1

B. Langkah Penelitian

Langkah-langkah penelitian dalam pembuatan robot dengan 4 roda *omni* ini ditunjukkan pada gambar Gambar 2. 1.

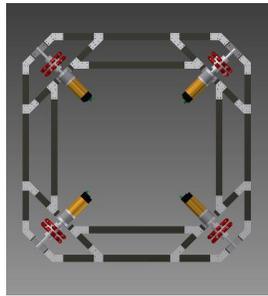


Gambar 2. 1 Diagram Alir Metode Penelitian

C. Perancangan Sistem

1. Perancangan Mekanik

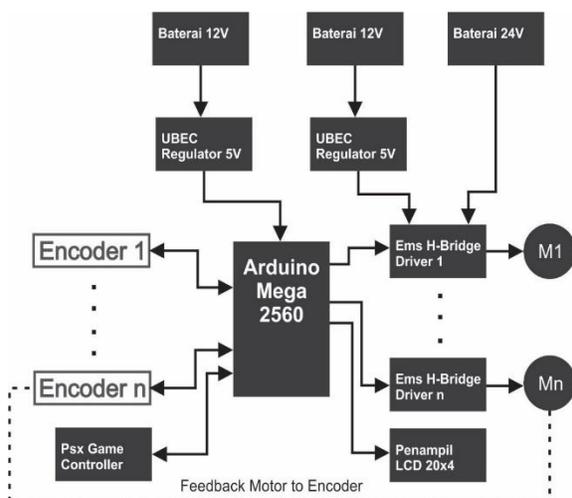
Perancangan mekanik dilakukan dengan menggambar *sketch* pada *software* perancangan mekanik yang kemudian dilakukan pembuatan tiap bagian yang telah dibuat. Dalam penelitian ini dibuat *base form* persegi dengan 4 roda omni. Kemudian dilakukan perakitan bagian-bagian tersebut menjadi satu kesatuan.



Gambar 2. 2 Gambar Rancangan Robot dengan Empat Roda Omni

2. Perancangan Rangkaian Kontrol

Pada penelitian ini dilakukan pembuatan suatu sistem kontrol yang mengatur kestabilan gerak robot. Robot menggunakan aktuator motor DC yang di pasang dengan roda omni. Sistem ini tersusun dari berbagai beberapa blok rangkaian, yaitu mikrokontroler (Arduino Mega 2560) sebagai pengendali, *encoder* sebagai sensor pendeteksi kecepatan dan *driver* motor *Embedded Module Series (EMS) 30 A H-Bridge* yang kemudian menggerakkan motor DC dan roda omni.



Gambar 2. 3 Blog Diagram Rangkaian Kendali Kecepatan Motor Metode PID

3. Perancangan Perangkat Lunak

Perancangan yang pertama yaitu perancangan program perhitungan kecepatan roda omni. Diketahui bahwa motor DC yang digunakan dalam percobaan ini mempunyai *pulse per revolution (ppr)* sebanyak 7 ppr. Dengan rasio *gearbox* perbandingan 1:19 maka poros roda memiliki 133 ppr. Dengan begitu resolusi *encoder* tiap detak impuls sebesar 2.7° .

$$ppr \text{ roda} = 19 \times 7$$

$$ppr \text{ roda} = 133$$

Dalam pemrograman dilakukan cuplik *external interrupt* dalam setiap 0.03 detik (0.03 detik dibuat dalam *internal interrupt*). Dari sana akan di dapat kecepatan putaran per 0.03 detik.

$$\text{revolusi per } 0.03 = \frac{\text{count external interrupt}}{133} \quad (2.1)$$

Karena satuan umum yang biasa dipakai untuk kecepatan putaran adalah revolusi per menit, maka perlu dilakukan konversi satuan dari revolusi per 0.03 detik ke satuan revolusi per menit (rpm). Pertama dilakukan konversi ke revolusi ke detik dengan persamaan yaitu dikalikan 33.333, angka tersebut di dapat dari 1 di bagi dengan 0,03. Yang mana dalam satu detik terdapat sebanyak 33.333 kali 0,03 detik. Dalam rumus perhitungan kecepatan motor *cei* adalah *count external interrupt*.

$$\text{revolusi per detik} = \frac{cei}{133} \times 33.33 \quad (2.2)$$

Kemudian untuk mendapatkan satuan revolusi per menit dikalikan 60 karena dalam satu menit sama dengan 60 detik.

$$\text{revolusi per menit} = \frac{cei}{133} \times 33.33 \times 60 \quad (2.3)$$

$$\text{revolusi per menit} = \frac{cei}{133} \times 2.000 \quad (2.4)$$

Dari perhitungan itulah di dapat rumus perhitungan rpm seperti yang ditunjukkan pada persamaan (2.4).

Perancangan yang ke dua yaitu perancangan kontrol PID. Kombinasi dari kontrol proporsional, aksi kontrol integral, dan aksi kontrol turunan disebut kontrol PID. Kombinasi ini mempunyai keuntungan dibanding masing-masing kontroler kekurangan dan kelebihan

kontroler saling menutupi. Persamaan dengan tiga kombinasi ditunjukkan pada persamaan (2. 5).

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2. 5)$$

Rumus PID yang digunakan pada program yaitu ditunjukkan pada persamaan (2. 6).

$$V_o = K_p \times err + K_i \times (err + last_{err}) \times T_s + \frac{K_d}{T_s} \times (err - last_{err}) \quad (2. 6)$$

Dimana:

- K_p adalah konstanta proporsional
- K_i adalah konstanta integral
- K_d adalah konstanta diferensial
- Err adalah nilai kesalahan
- $Last_{err}$ adalah nilai kesalahan sebelumnya
- T_s adalah *sampling time* (waktu cuplik)

III. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan membandingkan pergerakan perpindahan dengan tanpa dan dengan sistem dengan kontrol PID.

A. Pengujian Putaran Motor untuk Gerak Robot

Pengujian putaran motor untuk gerak perpindahan robot dilakukan untuk menguji arah putaran tiap roda *omni* dan gerakan perpindahan robot yang dihasilkan dengan dari putaran tersebut.

Tabel 3. 1 Tabel Konfigurasi Putaran Motor untuk Gerak Robot

Arah Robot	Motor 1	Motor 2	Motor 3	Motor 4
Maju	CCW	CCW	CW	CW
Mundur	CW	CW	CCW	CCW
Kiri	CW	CCW	CCW	CW
Kanan	CCW	CW	CW	CCW
Maju Kiri	-	CCW	-	CW
Maju Kanan	CCW	-	CW	-
Mundur kiri	CW	-	CCW	-
Mundur Kanan	-	CW	-	CCW

B. Kalibrasi RPM

Kalibrasi Rpm dilakukan untuk memastikan bahwa perhitungan Rpm yang dieksekusi oleh mikrokontroler telah sesuai. Kalibrasi dilakukan dengan membandingkan nilai rpm hasil

perhitungan mikrokontroler dengan rpm yang terbaca pada alat yang telah standar, dalam percobaan ini menggunakan *tachometer* Krisbow tipe KW06-583.

Tabel 3. 2 Hasil Kalibrasi Pembacaan Rpm Motor pada Arduino

No.	Rpm Terbaca Pada Tachometer	Rpm Hasil Perhitungan
1.	84.1	83.874
2.	259.7	261.014
3.	370.3	371.042
4.	427.1	433.668
5.	459	464.684
6.	472.9	478.212
7.	482.7	489.086
8.	489.6	495.512
9.	493.8	501.602
10	502.1	509.388
Rata-rata	404.13	408.8

Dari kalibrasi pembacaan Rpm pada kontroler, didapat nilai *error* sebesar 1.04%. dari nilai kesalahan yang kecil tersebut, dapat dikatakan pembacaan kecepatan motor pada kontroler sudah cukup baik.

$$Error = \frac{|404.13 - 408.8|}{404.13} \times 100\%$$

$$Error = 1.15\%$$

C. Pengujian Gerak Robot dengan Kontrol Open Loop

Pengujian ini dilakukan untuk melihat gerak robot dengan menggunakan *open loop control*. Kemudian hasil dari pengujian ini dibandingkan dengan hasil pengujian *close loop control PID*.

1. Pengujian Kecepatan Motor dengan Kontrol Open Loop

Dalam pengujian ini motor diberi masukan *open loop* dimana tidak ada umpan balik dari putaran motor untuk memperbaiki kecepatan

motor. Data pengujian kecepatan motor ditunjukkan pada Tabel 3. 3.

Tabel 3. 3 Data Pengujian Rpm Kontrol *Open Loop*

No.	Kecepatan yang diset	Motor 1 (E ₁)	Motor 2 (E ₂)	Motor 3 (E ₃)	Motor 4 (E ₄)
1	200	229.02	243.03	240.67	214.72
2	200	228.67	244.29	242.16	215.8
3	200	226.86	242.42	240.5	215.27
4	200	226.73	243.74	240.51	214.8
5	200	226.61	243.42	242.03	215.88
6	200	226.51	243.14	240.38	213.84
7	200	226.41	244.39	240.4	213.51
8	200	227.83	242.51	240.42	213.21
9	200	227.6	243.82	240.44	212.94
10	200	225.9	243.5	240.46	214.2
Rata-rata		227.21	243.38	240.79	214.41

Dari data pada Tabel 3. 3 dilihat bahwa masing-masing motor memiliki kecepatan yang berbeda. Hal itu dikarenakan kontrol yang digunakan yaitu *open loop* dimana tidak ada koreksi terhadap Rpm Motor. Nilai *error* rata-rata dari keempat motor yaitu 15.73%.

2. Pengujian Gerakan Robot dengan Kontrol *Open Loop*

Untuk menguji pergerakan robot dengan kontrol *open loop* dilakukan dengan menjalankan robot dari titik A ke titik B, kemudian diambil data berupa *error* sudut dari titik A ke titik B dan sebaliknya. Pada pengujian ini robot digerakkan ke arah depan (arah hadap robot) sejauh 2 meter, kemudian diukur *error* pergeseran robot dari sumbu arah gerak robot.



Gambar 3. 1 Titik Awal Pengujian Pergerakan Robot *Open Loop* (Titik A)

Gambar 3. 1 menunjukkan titik A atau titik awal pengujian pergerakan robot dengan menggunakan kontrol *open loop*. Pada pengujian pertama robot digerakkan ke titik B, kemudian diukur *error*-nya yaitu jarak robot dengan sumbu awal pergerakan robot.



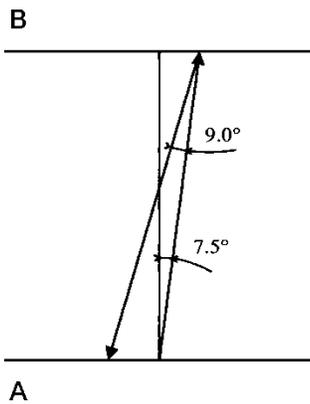
Gambar 3. 2 a) Pengujian Pergerakan Robot *Open Loop* dari Titik A ke Titik B b) *Error* Pergerakan Robot *Open Loop* dari Titik A ke Titik B

Error yang didapat dari pergerakan maju dari titik A ke titik B, yaitu 26cm. Nilai tersebut didapat pada saat robot mencapai titik B, seperti yang ditunjukkan pada Gambar 3. 2. Pada saat bergerak maju robot bergerak ke arah kanan hal itu dikarenakan titik berat robot berada pada bagian belakang kanan dari robot, sehingga beban pada motor 3 dan motor 4 lebih berat. Setelah pengujian maju, pada tahap ke dua robot digerakkan mundur dari titik B ke titik A dan kemudian di ambil data *error* nya.



Gambar 3. 3 a) Pengujian Pergerakan Robot *Open Loop* dari Titik B ke Titik A b) *Error* Pergerakan Robot *Open Loop* dari Titik B ke Titik A

Pada pengujian gerak mundur didapat *error* atau robot bergeser ke kiri dari sumbu awal sebesar 33 cm. Pergeseran itu dilihat pada Gambar 3. 3 di mana pada gambar a) robot bergeser dari titik awal percobaan (titik A).



Gambar 3. 4 *Error* pergerakan Robot dengan Kontrol *Open Loop*

Pada Gambar 3. 4 dilihat *error* pergerakan robot maju dan mundur cukup besar, hal itu dikarenakan kontrol yang digunakan pada roda robot *omni* masih menggunakan kontrol *open loop* dan juga penyebab lainnya yaitu titik beban dari robot yang juga tidak berada pada titik simetris dari dimensi *base* robot. Dan *error* yang terjadi pada saat maju dan mundur sebesar 4.58%.

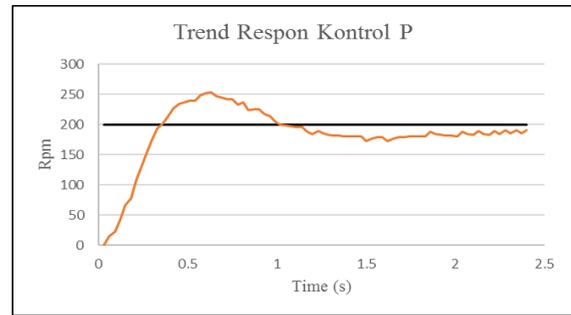
D. Tuning PID

Tuning dilakukan dengan cara *manual tuning*. Tahap *tuning* pada penelitian ini antara lain:

1. Tuning P

Pada percobaan ini dilakukan tuning parameter P dan didapat nilai $P = 0.04$. Nilai ini

didapat dengan mengganti nilai P dengan nilai tertentu dan kemudian ditentukan nilainya dengan pertimbangan *overshoot* dan *steady state error* tidak terlalu besar, karena pada saat penambahan parameter I akan membuat *overshoot* akan bertambah besar dan parameter I juga dapat mengurangi *steady state error*.



Gambar 3. 5 *Trend Respon* Kontrol Proporsional

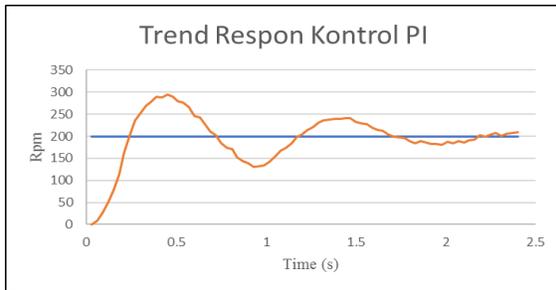
Dari *trend respon* kontrol proporsional yang ditunjukkan pada Gambar 3. 5 didapat parameter kualitatifnya yang ditunjukkan pada Tabel 3. 4. Tabel parameter kualitatif kontrol P menunjukkan bahwa *overshoot* masih cukup besar yaitu pada nilai 53.65 , *rise time* juga masih pada angka 0.36 detik dan begitu juga dengan *settling time* maupun *steady state error* masih menunjukkan angka yang cukup tinggi. Untuk itu dibutuhkan kontrol tambahan untuk mendapatkan respon kontrol yang lebih baik.

Tabel 3. 4 Tabel Paramter Kualitatif kontrol P

Parameter	Nilai
Set Point (rpm)	200
Overshoot (rpm)	53.65
Rise Time (s)	0.36
Settling Time (s)	1.86

2. Tuning PI

Pada *tuning* ini didapatkan nilai besar variabel integral (I)=5, nilai tersebut didapat dengan melihat grafik percobaan yang memperbaiki nilai *rise time* pada sistem. Di sisi lain turunnya nilai *rise time* oleh karena adanya variabel integral juga menaikkan nilai *overshoot* sehingga membuat sistem tidak stabil.



Gambar 3. 6 *Trend Respon Kontrol PI*

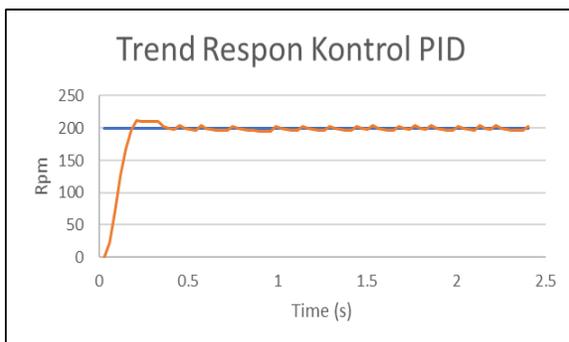
Dari *trend* respon kontrol proporsional dan integral yang ditunjukkan pada Gambar 3. 6 didapat parameter kualitatifnya yang ditunjukkan pada Tabel 3. 5. Dari tabel kualitatif tersebut ditunjukkan nilai *rise time* yang menurun tetapi di sisi lain *overshoot* menjadi sangat tinggi. Untuk meredam *overshoot* tersebut dibutuhkan variabel derivatif yang dapat mengoreksi sebelum terjadi kesalahan.

Tabel 3. 5 Tabel Paramter Kualitatif kontrol PI

Parameter	Nilai
Set Point (rpm)	200
Overshoot (rpm)	89.91
Rise Time (s)	0.24
Settling Time (s)	2.4
Ess (rpm)	9.4

3. Tuning PID

Nilai derivatif yang sesuai dengan sistem yaitu $K_d=10$. Nilai itu di *tuning* hingga didapatkan respon yang paling baik, yang dapat menghilangkan *overshoot*, mengurangi *rise time*, *settling time* dan *Error steady state* juga berkurang.



Gambar 3. 7 *Trend Respon Kontrol PID*

Dilihat pada Tabel 3. 6 dilihat data parameter kualitatif yang telah ditambahkan nilai derivatif, *overshoot* pada sistem berkurang dan tidak ada lagi isolasi yang terjadi. *Rise time* sistem juga semakin kecil, *settling time* juga lebih kecil dari kontrol P atau PI. *Settling time* yang kecil menandakan sistem mencapai kestabilan dengan cepat. *Error steady state* juga kecil, sehingga rpm motor mendekati dengan nilai *set point*.

Tabel 3. 6 Tabel Paramter Kualitatif kontrol PID

Parameter	Nilai
Set Point (rpm)	200
Overshoot (rpm)	10.88
Rise Time (s)	0.18
Settling Time (s)	0.42
Ess (rpm)	3.13

E. Pengujian Kestabilan Motor DC dan Gerak Robot dengan Kontrol PID

Pada pengujian ini dilakukan uji kestabilan rpm tiap motor pada roda *omni* dan pergerakan robot setelah menggunakan kontrol PID yang telah di *tuning* sebelumnya. Dan akan dilihat *error* dari pergerakan perpindahan dari robot. Kemudian akan dibandingkan dengan uji pergerakan pada kontrol *open loop* yang telah dilakukan sebelumnya.

1. Pengujian RPM Motor 1, 2, 3 dan 4

Untuk mengetahui kestabilan motor, dilakukan uji rpm masing-masing motor yang telah dikontrol dengan metode PID dan akan dilihat *error* dari tiap roda. Pada Tabel 3. 7 Dilihat *error steady state* diketahui rata-rata *error* masing-masing motor relatif kecil dan dapat dikatakan stabil.

Persentase *error* rata-rata pada kecepatan motor 1 sebesar 2.84%, motor 2 sebesar 1.32%, motor 3 sebesar 3.08%, dan motor 4 sebesar 3.99%. Nilai lebih baik dibandingkan dengan *error* ketika menggunakan kontrol *open loop* yang tidak mempunyai umpan balik. Kemudian setelah dilakukan uji rpm, dilakukan uji gerak robot. Nilai *error* dari keempat motor yaitu 2.81%.

Tabel 3. 7 Tabel Pegujian Motor 1, 2, 3 dan 4

Time	Set Point	RPM Motor			
		M1	M2	M3	M4
0.3	200	0	0	0	0
0.33	200	7.52	7.52	15.04	15.04
0.36	200	48.87	48.87	60.15	52.63
0.39	200	92.11	99.62	105.26	86.47
0.42	200	136.28	140.04	142.86	125.94
0.45	200	173.4	175.28	176.69	168.23
.
.
.
2.28	200	209.58	197.91	202.75	208.63
2.31	200	195.01	204.22	206.64	209.58
2.34	200	202.77	207.37	208.58	210.05
2.37	200	206.65	201.43	209.55	210.29
2.4	200	208.59	198.46	210.04	210.41
Rata Rata Ess		5.99	3.73	6.16	8.4

2. Pengujian Gerak Robot dengan Kontrol PID

Pengujian gerak robot dengan kontrol PID ini sama tekniknya dengan pengujian pada uji pergereakan robot dengan kontrol *open loop*. Yaitu pertama robot diletakan pada titik awal (titik A) kemudian dijalankan sejauh 2 meter ke arah depan (arah hadap robot) atau ke titik B, kemudian di ukur sudut yang terbentuk dengan sumbu lurus gerak robot, begitu juga di uji gerak mundur dan di ukur sudut yang terbentuk dengan garis pertama.



Gambar 3. 8 Titik Awal Pengujian Pergerakan Robot Kontrol PID (Titik A)

Pada Gambar 3. 9 *error* yang terukur setelah robot bergerak ke arah depan (dari titik A ke titik B) yaitu 5.8cm. Arah *error* dari robot sama dengan uji pada kontrol *open loop*, tetapi nilai *error* dari pergerakan jauh lebih kecil. Arah *error*

yaitu ke arah kanan di mana titik berat robot berada di bagian kanan belakang dari *base* robot. Kemudian selanjutnya robot di uji pergerakan mundurnya, yaitu dari titik B ke titik A tempat posisi awal robot.



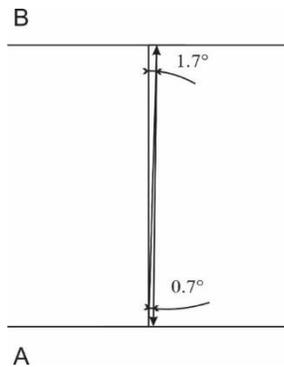
Gambar 3. 9 a) Pengujian Pergerakan Robot kontrol PID dari Titik A ke Titik B b) *Error* Pergerakan Robot kontrol PID dari Titik A ke Titik B

Gambar 3. 10 menunjukkan *error* gerak mundur robot, pada pengujian tersebut didapat angka *error* sebesar 2.5 cm dari titik awal *start*. Nilai itu juga jauh lebih kecil dibandingkan dengan *error* gerak mundur saat *open loop*.



Gambar 3. 10 a) Pengujian Pergerakan Robot kontrol PID dari Titik B ke Titik A b) *Error* Pergerakan Robot kontrol PID dari Titik B ke Titik A

Setelah pengujian maju dan mundur dan di dapat *error* masing-masing dilakukan, didapat *error* sudut arah hadap robot. Dari gerak maju *error* 5.8 cm didapat *error* arah hadap robot sebesar 1.7° dan dari gerak mundur dengan *error* 2.5cm didapat *error* arah hadap robot sebesar 0.7°.



Gambar 3. 11 *Error* Pergerakan Robot dengan Kontrol PID

Pada Gambar 3. 11 dilihat sudut yang dibentuk dari pergerakan robot maju dan mundur jauh lebih kecil dibandingkan dengan sudut *error* yang dibentuk dengan kontrol *open loop*. *Error* yang terjadi pada saat menggunakan kontrol *open loop* sebesar 0.66%. Dapat dikatakan kontrol PID yang dibuat dapat membuat gerak robot lebih lurus dan dapat meredam efek titik berat yang tidak berada pada tengah dari dimensi *base* robot.

IV. KESIMPULAN

A. Kesimpulan

Berdasarkan dari pengujian dan analisa dari penelitian yang berjudul “Kendali Kecepatan Motor Pada Robot Dengan Empat Roda *Omni* Menggunakan Metode PID” ini maka dapat ditarik kesimpulan sebagai berikut:

1. Dengan metode PID untuk mengontrol masing masing roda *omni*, pergerakan perpindahan robot menjadi lurus dan *error* arah hadap robot menjadi kecil. Dan dapat memperbaiki *error* arah hadap dari *open loop* sebesar 4.58% menjadi 0.66%.
2. Pegujian putaran roda dengan kontrol PID dapat membuat kecepatan putaran masing-masing roda mendekati sama dengan persentase *error* rata-rata pada kecepatan motor sebesar 2.81% dibandingkan dengan *error* rata-rata pada kecepatan motor dengan kontrol *open loop* sebesar 15.73%.

B. Saran

Kontrol robot dalam penelitian ini sudah mampu untuk bergerak dengan *error* hadap yang relatif kecil. Untuk lebih meminimalisir nilai *error*

dibutuhkan perhitungan *center of gravity* dari robot, agar beban yang diterima masing-masing roda sama, sehingga membuat pergerakan robot lebih baik.

Jika ingin mengembangkan penelitian ini lebih lanjut, dibutuhkan kontrol yang dapat membaca dan menggerakkan robot dalam koordinat. Sehingga robot dapat bergerak ke manapun dengan diketahui posisinya.

V. UCAPAN TERIMAKASIH

Pertama penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa yang memberikan karunia-Nya sehingga penulis dapat menyelesaikan penelitian ini. Kepada orang tua dan keluarga yang selalu memberikan kasih sayang, doa serta dukungan moril dan materil. Dan juga kepada berbagai pihak yang telah membantu di antaranya Bapak Beny Firman, S.T., M.Eng. sebagai pembimbing 1 yang telah membimbing dalam pengerjaan serta penulisan laporan. Bapak Dr. Samuel Kristiyana, S.T., M.T. sebagai pembimbing 2 yang telah meluangkan waktu untuk memberi masukan serta ide dalam pengerjaan maupun laporan serta jurnal. Kemudian teman-teman dan berbagai pihak yang telah membantu dan tidak dapat disebutkan satu persatu.

VI. DAFTAR PUSTAKA

- Amaldi, W., 2016. Ilmu Program. [Online] Available at: <https://ilmuprogram.com/2016/12/17/arduino-penjelasan-dan-macam-macamnya/>[Accessed 27 September 2017].
- Arduino, 2017. Arduino Mega. [Online] Available at: <https://www.arduino.cc/en/Main/arduinoBoardMega>[Accessed 26 Juli 2017].
- COLLINS, D., 2015. FAQ: How do magnetic encoders work?. [Online] Available at: <http://www.motioncontroltips.com/faq-how-do-magnetic-encoders-work/>[Accessed 27 Juli 2017].
- Dynpar, 2017. Magnetic Encoders. [Online] Available at: <https://www.dynpar.com>.

- com/Technology/Encoder_Basics/Magnetic_Encoder/[Accessed 27 Juli 2017].
- ElektronikaDasar, 2012. LCD (Liquid Cristal Display). [Online] Available at: <http://elektronika-dasar.web.id/lcd-liquid-cristal-display/>[Accessed 26 Juli 2017].
- Feri Yusivar, j., 2010. PERANCANGAN PENGENDALI PID PADA PROPORTIONAL.
- Hutahaean, R. Y., 2006. Mekanisme dan Dinamika Mesin. Yogyakarta: Andi.
- InovativeElectronic, 2007. EMS 30 A H-Bridge. [Online] Available at: http://www.innovativeelectronics.com/innovative_electronics/download_files/manual/EMS_30_A_HBridge_manual.pdf[Accessed 27 Juli 2017].
- Irfan, M., 2016. DESAIN DAN IMPLEMENTASI KENDALI KECEPATAN MOTOR PADA ROBOT DENGAN EMPAT RODA OMNI MENGGUNAKAN METODE LOGIKA FUZZY. Jurnal.
- Kho, D., 2017. Pengertian Optocoupler dan Prinsip Kerjanya. [Online] Available at: <http://teknikelektronika.com/pengetahuan-optocoupler-fungsi-prinsip-kerja-optocoupler/>[Accessed 27 Juli 2017].
- Ogata, K., 1996. Teknik Kontrol Automatik Jilid 1. Jakarta: Erlangga.
- Pambudi, W. S., 2011. RANCANG BANGUN 3 WHEELS OMNI-DIRECTIONAL MOBILE ROBOT MENGGUNAKAN SENSOR POSITION SENSITIVE DEVICE (PSD) SERTA SENSOR VISION DENGAN METODE KENDALI FUZZY LOGIC CONTROLLER (FLC) UNTUK MENGHINDARI HALANGAN. Sematik.
- Pitowarno, E., 2006. Desain, Kontrol, dan Kecerdasan Buatan. Yogyakarta: Andi.
- Rochamnto, R. A., 2014. IMPLEMENTASI ROBOT THREE OMNI-DIRECTIONAL MENGGUNAKAN KONTROLER PID PADA ROBOT KONTES ROBOT ABU INDONESIA (KRAI). Jurnal Universitas Brawijaya, pp. 1-6.
- Saputra, A. W., 2014. Kendali Kecepatan dan Posisi pada Mobile Robot yang Menggunakan Triangle Omni-Directional Wheels dengan Metode PID. Journal of Control and Network Systems, Volume 3, No. 2, pp. 81-89.
- Sumanto, M., 1991. Mesin Arus Searah. ke2 ed. Yogyakarta: Andi Offset.
- Toto Alfian Wahyuono, H. W. M. C. W., 2015. KENDALI ROBOT MANUAL 4WD MECANUM WHEEL BERBASIS PID DENGAN MENGGUNAKAN ARM – CORTEX M4.. Journal of Control and Network Systems, Volume 4, No.1, pp. 39-45.