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

Aryono Priyambudi[1], Beny Firman[2], Samuel Kristiyana[3]
[1,2,3]Teknik Elektro, Institut Sains & Teknologi AKPRIND Yogyakarta
Email: [1]priyambudi.ary@gmail.com, [2]benyfirman@akprind.ac.id, [3]yanaista@akprind.ac.id

Masuk: 12 Januari 2018, Revisi masuk: 30 Januari 2018, Diterima: 01 Februari 2018

ABSTRACT

In this research has been developed a system to conrol robot which have four wheel as a base for wheeled robot contest. The type of wheel is omni-wheel, they are shapeed with same distance each other. This system comonly have no stright 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 strightly 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 paramater 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: *Manual tuning, Omni-wheel, PID, Wheeled robot.*

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 pegendalian untai terbuka. Sistem penngendalian pergerakan robot pada penelitian ini menggunakan metode PID untuk mengendalikan kecepatan putran 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 sebessar 0.66% dari hasil pengujian pergerakan maju dan mundur robot dengan arah hadap yang sama.

Kata-kata kunci: PID, robot beroda, roda *omni*, Tuning manual.

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 peggerak 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*) (Wahyuono, 2015). PID adalah salah satu contoh metode untuk mengatur pergerakan robot. PID merupakan kontrol yang menggabungkan aksi kontrol propoesional, 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.

Metodologi

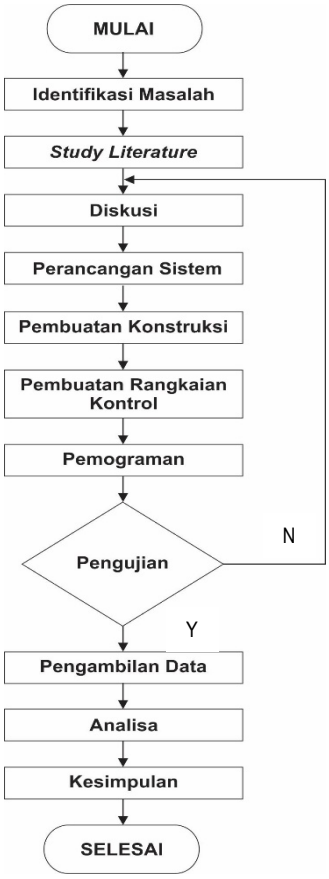
Alat dan spesifikasi yang dibutuhkan pada penelitian adalah sebagai berikut:

1. *Notebook* TOSHIBA *Satellite* C840 *Series* dengan spesifikasi processor Intel Core i3, RAM 4096 MB, VGA Card ATI Radeon, *Hard Disk Drive* 350 GB
2. Arduino IDE 16.9
3. *Software* Eagle

Bahan yang digunakan adalah sebagai berikut:

1. *Omni wheel* diameter 127 mm 4 unit
2. Motor DC dengan *encoder* 45mm, *planetary reducer* 1:19.2, 4 unit
3. Mikrokontroler Arduino Mega 2560 1 unit
4. *Driver* motor *Embedded Module Series* (EMS) 30 A H-Bridge, 4 unit
5. Regulator tegangan UBEC 5A, SBEC 15A, 1 unit
6. *Tachometer*, 1 unit
7. Krisbow *Tachometer* (KW06-563)

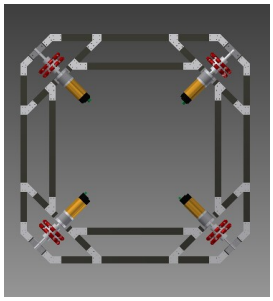
Langkah-langkah penelitian dalam pembuatan robot dengan 4 roda *omni* ini ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Metode Penelitian

1. Perancangan Sistem Mekanik

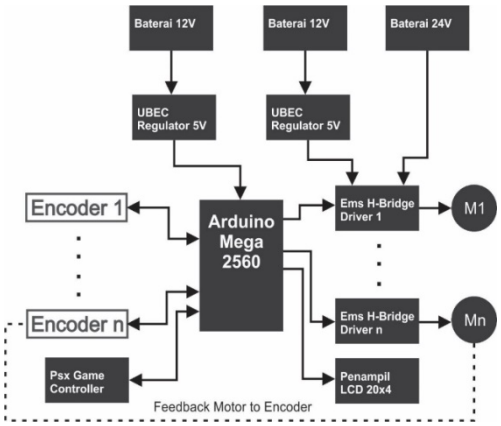
Perancangan mekanik dilakukan dengan menggambar *sketch* pada *software* perancangan mekanik yang kemudian dilakukan pembuatan tiap bagian yang telah dibuat. Dalam penelitian ini dibuat *base form* persegi dengan 4 roda *omni*. Kemudian dilakukan perakitan bagian tersebut menjadi satu kesatuan. Gambar 1 menampilkan rancangan robot dengan empat roda *omni*.



Gambar 2. 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 dipasang 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 3 menampilkan blog diagram rangkaian kendali kecepatan motor metode PID.



Gambar 3. Blog Diagram Rangkaian Kendali Kecepatan Motor Metode PID

3. Perancangan Perangkat Lunak

Pertama yang dilakukan adalah perancangan program perhitungan kecepatan roda *omni*. Motor DC yang digunakan dalam percobaan ini mempunyai *pulse per revolution (ppr)* sebanyak 7 ppr. Dengan rasio *gearbox* perbandingan 1:19 dengan poros roda memiliki 133 ppr, menunjukkan resolusi *encoder* tiap detak inpuls sebesar 2.7°.

$$ppr\ roda = 19 \times 7$$
$$ppr\ roda = 133$$

Dalam pemrograman dilakukan *external interrupt sampling* dalam setiap 0,03 detik (0,03 detik dibuat dalam internal *interrupt*), menghasilkan kecepatan putaran per 0,03 detik.

$$revolusi\ per\ 0{,}03 = \frac{count\ external\ interrupt}{133} \quad (1)$$

Satuan umum kecepatan putaran adalah revolusi per menit, dilakukan pengubahan satuan dari revolusi per 0,03 detik ke konversi revolusi per menit (rpm). Revolusi konversi detik dengan persamaan (1) dikalikan 33.33, angka tersebut didapat dari 1 dibagi dengan 0,03. Dalam satu detik terdapat sebanyak 33.333 kali 0,03 detik. Dalam persamaan perhitungan kecepatan motor *cei* adalah*count external interrupt*.

$$revolusi\ per\ detik = \frac{cei}{133} \times 33.33 \quad (2)$$

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

$$revolusi\ per\ menit = \frac{cei}{133} \times 33.33 \times 60 \quad (3)$$

$$revolusi\ per\ menit = \frac{cei}{133} \times 2.000 \quad (4)$$

Dari perhitungan itulah didapat persamaan perhitungan rpm seperti yang ditunjukkan pada persamaan (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 (5).

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

Persamaan PID yang digunakan ditunjukkan pada persamaan (6).

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

dengan:

- K_p adalah konstanta proporsonal
- K_i adalah konstanta integral
- K_d adalah konstanta diferensial
- *Err* adalah nilai kesalahan
- *Last*$_{err}$ adalah nilai kesalahan sebelumnya
- Ts adalah *sampling time* (waktu *sampling*)

PEMBAHASAN

Pengujian dilakukan dengan membandingkan pergerakan perpindahan menggunakan sistem kontrol PID dan tanpa menggunakan sistem 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 1. 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 digunakan *Tachometer* Krisbow tipe KW06-583.

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

Tabel 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 tampak 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 meguji pergereakan robot dengan kontrol *open loop* dilakukan dengan mejalankan 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 4 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 5 (a) menunjukkan pergerakan robot *open loop* dari titik A ke titik B, sedangkan (b) *error* pergerakan robot *open loop* dari titik A ke titik B.

Gambar 4. Titik Awal Pengujian Pergerakan Robot *Open Loop* (Titik A)

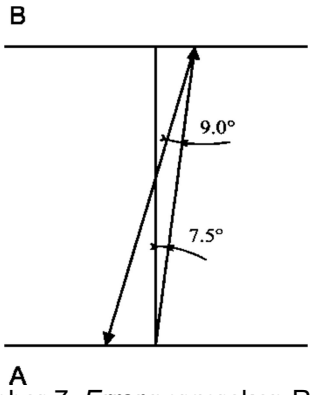
Gambar 5 a). Pengujian Pergerakan Robot *Open Loop* dari Titik A ke Titik B b). *Error* Pergerakan Robot *Open Loop* dari Titik A ke Titik B

Error yang didapat dari pergerakan maju dari titik A ke titik B adalah 26 cm. Nilai tersebut didapat pada saat robot mencapai titik B, seperti yang ditunjukkan pada Gambar . 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.

Pada pengujian gerak mundur didapat *error* atau robot bergeser ke kiri dari sumbu awal sebesar 33 cm. Pergeseran itu dilihat pada Gambar 6 di mana pada Gambar a) robot bergeser dari titik awal percobaan (titik A). Pada pengujian gerak mundur didapat *error* atau robot bergeser ke kiri dari sumbu awal sebesar 33 cm. Pergeseran itu dilihat pada Gambar 6 di mana pada gambar a) robot bergeser dari titik awal percobaan (titik A).

Gambar 6a). Pengujian Pergerakan Robot *Open Loop* dari Titik B ke Titik A b). *Error* Pergerakan Robot *Open Loop* dari Titik B ke Titik A

Pada Gambar 7 dapat 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. *Error* yang terjadi pada saat maju dan mundur sebesar 4.58%.



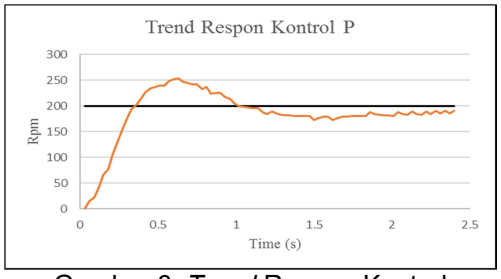
Gambar 7. *Error* pergerakan Robot dengan Kontrol *Open Loop*

D. *Tuning PID*

Tuning dilakukan dengan cara *manual tuning*.

1. *Tuning* P

Pada percobaan ini dilakukan tuning paramater P dan didapat nilai P = 0.04. Nilai ini didapat dengan menganti nilai P dengan nilai tertentu dan kemudian ditentukan nilainya dengan pertimbangan *overshoot* dan *steady state error* tidak terlalu besar, karena pada saat panambahan paramater I akan membuat *overshoot* akan bertambah besar dan parameter I juga dapat mengurangi *steady state error*. Gambar 8 menampilkan *trend* respon kontrol proporsional.



Gambar 8. *Trend* Respon Kontrol Proporsional

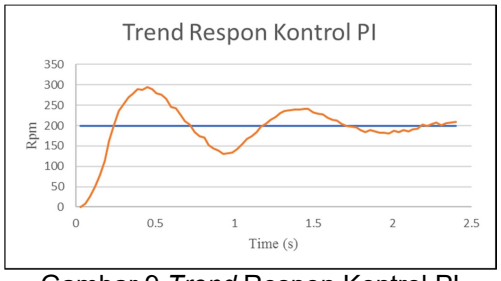
Dari *trend* respon kontrol proporsional yang ditunjukkan pada Gambar didapat parameter kualitatifnya yang ditunjukkan pada Tabel 4. Tabel parameter kualitatif kontrol P menunjukkan bahwa *overshoot* masih cukup besar yaitu pada nilai 53,65, *rise time* juga masih pada angka 0,36 detik dan begitu juga dengan *settling time* maupun *steady steade error* masih menunjukkan angka yang cukup tinggi. Untuk itu dibutuhkan kontrol tambahan untuk mendapatkan respon kontrol yang lebih baik.

Tabel 4. 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 9 menampilkan *trend* respon kontrol PI.



Gambar 9 *Trend* Respon Kontrol PI

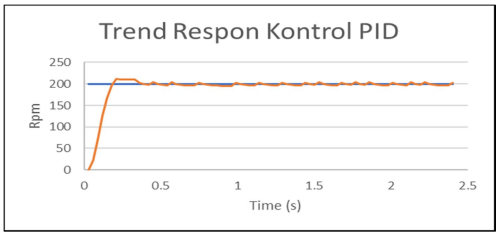
Dari *trend* respon kontrol proporsional dan integral didapat parameter kualitatifnya yang ditunjukkan pada Tabel 5. Dari tabel tersebut ditunjukkan nilai *rise time* yang menurun tetapi di sisi lain *overshoot* menjadi sangat tinggi. Untuk meredam *overshoot* tersebut dibutuhkan variabel derivatif yang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

Tabel 5. 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 Kd=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 10 menampilkan *trend* respon kontrol PID.



Gambar 10. *Trend* Respon Kontrol PID

Pada Tabel dapat dilihat data parameter kualititif yang telah ditamahkan 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 6. 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. PadaTabel 6 dapat dilihat *error steady state*, 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 ini 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 adalah 2,81%.

Tabel 6 Pengujian Motor 1, 2, 3, dan 4

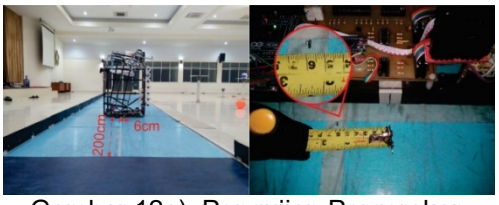
Time	*Set Point*	RPM Motor			
		M1	M2	M3	M4
0.3	200	0	0	0	0
0.33	200	7.52	7.52	15.04	15.04
0.36	200	48.87	48.87	60.15	52.63
0.39	200	92.11	99.62	105.26	86.47
0.42	200	136.28	140.04	142.86	125.94
0.45	200	173.4	175.28	176.69	168.23
2.28	200	209.58	197.91	202.75	208.63
2.31	200	195.01	204.22	206.64	209.58
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 pergerakan robot dengan kontrol *open loop*. Pertama robot diletakkan pada titik awal (titik A) kemudian dijalankan sejauh 2 meter ke depan (arah hadap robot) atau ke titik B, kemudian di ukur sudut yang terbentuk dengan sumbu lurus gerak robot, begitu juga diuji gerak mundur dan di ukur sudut yang terbentuk dengan garis pertama. Gambar 11 menampilkan titik awal pengujian pergerakan robot kontrol PID (Titik A).

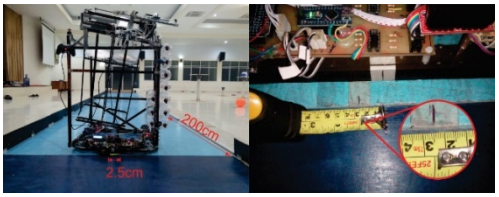
Gambar 11. Titik Awal Pengujian Pergerakan Robot Kontrol PID (Titik A)

Pada Gambar 12 e*rror* 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 arrah kanan di mana titik berat robot berada di bagian kanan belakang dari *base* robot. Selanjutnya robot diuji pergerakan mundurnya, yaitu dari titik B ke titik A tempat posisi awal robot.



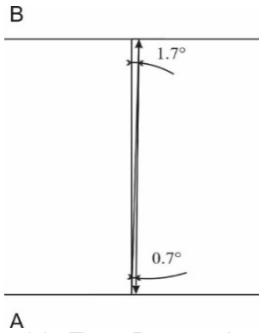
Gambar 12a). Pengujian Pergerakan Robot kontrol PID dari Titik A ke Titik B b) *Error* Pergerakan Robot kontrol PID dari Titik A ke Titik B

Gambar 3 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 13a). 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* 2.5 cm didapat *error* arah hadap robot sebesar 0,7°, seperti tampak pada Gambar 14.



Gambar 14. *Error* Pergerakan Robot dengan Kontrol PID

Pada Gambar 114 tampak sudut yang dibentuk dari pergerkan 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.

KESIMPULAN

Berdasarkan hasil penelitian dapat ditarik kesimpulan sebagai berikut:

1. Dengan metode PID untuk mengontrol masing masing roda *omni,* pergerakan perpindahan robot menjasi 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%.

Kontrol robot dalam penelitian ini sudah mampu untuk bergerak dengan *error* hadap yang relatif kecil. Untuk meminimalisir nilai *error* dibutuhkan perhitungan *center of gravity* dari robot, agar beban yang diterima masing-masing roda sama, sehingga membuat pergerakan robot lebih baik. Untuk pengembangan lebih lanjut, dibutuhkan kontrol yang dapat membaca dan menggerakan robot dalam koordinat. Sehingga robot dapat bergerak kemanapun dengan diketahui posisinya.

DAFTAR PUSTAKA

Amaldi, W., 2016, Ilmu Program, https://ilmuprogram.com/2016/12/17/arduino-penjelasan-dan-macamma camnya/, diakses 27 September 2017.

Arduino, 2017, Arduino Mega, https://www.arduino.cc/en/Main/arduino BoardMega, diakses 26 Juli 2017.

Collins, D., 2015, FAQ: How do magnetic encoders work?, http://www.motioncontroltips.com/faq-how-do-magnetic-encoders-work/, diakses 27 Juli 2017.

Dynpar, 2017, Magnetic Encoders, https://www.dynapar.com/Technology/Encoder_Basics/Magnetic_Encoder/, diakses 27 Juli 2017.

Hutahaean, R. Y., 2006, Mekanisme dan Dinamika Mesin, Yogyakarta: Andi.

Kho, D., 2017, Pengertian Optocoupler dan Prinsip Kerjanya, http://teknikelektronika.com/pengertian-optocoupler-fungsi-prinsip-kerja-optocoupler/, diakses 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.

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, hal. 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*, Vol. 3, No. 2, hal. 81-89.

Sumanto, M., 1991, Mesin Arus Searah, Edisi ke-2, Yogyakarta: Andi Offset.

Wahyuono, T. A., 2015, Kendali Robot Manual 4WD Mecanum Wheel Berbasis PID dengan Menggunakan ARM-CORTEX M4, *Journal of Control and Network Systems*, Vol. 4, No.1, hal. 39-45.

BIODATA PENULIS

Aryono Priyambudi, S.T., lahir di Bangka pada tanggal 10 Januari 1995, menyelesaikan pendidikan S1 bidang Teknik Elektro dari IST AKPRIND Yogyakarta tahun 2018.

Beny Firman, S.T., M.Eng., lahir di Pangkalpinang pada tanggal 5 Juli 1986, menyelesaikan pendidikan D3 dan S1 bidang Teknik Elektro dari IST AKPRIND Yogyakarta dan S2 bidang Teknik Elektro dan Teknologi Informasi dari UGM tahun 2012.

Dr. Samuel Kristiyana, S.T., M.T., lahir di Bantul pada tanggal 6 Desember 1970, menyelesaikan pendidikan S1 bidang Teknik Elektro dari IST AKPRIND Yogyakarta, S2 bidang Teknik Elektro dan Teknologi Informasi dari UGM tahun 2005, dan menyelesaikan S3 bidang Teknik Elektro dan Teknologi Informasi dari UGM tahun 2017.