

Desain dan analisis pengaruh parameter PI dan variasi kecepatan pada respon sistem kontrol arah hadap *prototype* kendaraan listrik

Afif Caesar Distara^{1*}, Fatkhur Rohman²

Program Studi Teknik Otomotif Elektronik, Jurusan Teknik Mesin, Politeknik Negeri Malang^{1,2}

Jl. Soekarno Hatta no. 09, Malang 64141, Indonesia^{1,2}

Email: afifcaesar1@gmail.com¹, fatkhur_rohman@polinema.ac.id²

Abstract

Electric vehicles are alternative vehicles that carry energy efficient. At this time the dominant vehicle uses ordinary wheels so that it will become an obstacle in the maneuver function that requires movement in various directions. With mehanum wheels the vehicle can move in various directions by adjusting the direction of rotation of each wheel. The problem is choosing the right control system for the control system needed by the vehicle. The purpose of this study is to determine and analyze the effect of variations in the value of PI (Proportional Integral) and speed of the vehicle to the stability response of the system to control the direction of prototype electric vehicles. This study method is an experiment that is by giving a treatment, then evaluating the effects caused by the research object. The results of this study can be concluded that the variation of PI constant values and speed variations have an effect on the stability parameters of the system, namely rise time, settling time, overshoot, and steady state error. To get the best system stability response results can use the constant value PI $K_p = 2$; and $K_i = 17$; where the stability response of the system for direction control at each speed condition has a fairly good value with a fast rise time, fast settling time, small overshoot and a small error steady state compared to other PI constant values in this study.

Keywords: *mechanum wheel, PI control, direction, prototype, system stability*

Abstrak

Kendaraan listrik merupakan kendaraan alternatif yang mengusung hemat energi. Pada saat ini kendaraan dominan menggunakan roda biasa sehingga akan menjadi kendala dalam fungsi manuver yang membutuhkan pergerakan ke berbagai arah. Dengan roda mekanum kendaraan dapat bergerak ke berbagai arah dengan mengatur arah putaran setiap roda. Permasalahannya adalah pemilihan sistem kendali yang tepat karena harus mengatur empat roda sekaligus. Tujuan dari penelitian ini adalah mengetahui dan menganalisa pengaruh variasi nilai PI (*Proportional Integral*) dan kecepatan laju kendaraan terhadap respon kestabilan sistem pada kontrol arah hadap *prototype* kendaraan listrik. Metode penelitian ini adalah eksperimen yaitu dengan cara memberi suatu perlakuan, kemudian melakukan evaluasi terhadap pengaruh yang ditimbulkan pada objek penelitian. Hasil dari penelitian ini dapat disimpulkan bahwa variasi nilai konstanta PI dan variasi kecepatan mempunyai pengaruh terhadap parameter kestabilan sistem yaitu *rise time*, *settling time*, *overshoot*, dan *error steady state*. Untuk mendapatkan hasil respon kestabilan sistem terbaik dapat menggunakan nilai konstanta PI $K_p = 2$; dan $K_i = 17$; dimana respon kestabilan sistem untuk kontrol arah hadap pada setiap kondisi kecepatan memiliki nilai yang cukup baik dengan *rise time* yang cepat, *settling time* yang cepat, *overshoot* yang kecil dan *error steady state* yang kecil dibandingkan dengan nilai konstanta PI yang lainnya dalam penelitian ini.

Kata kunci: roda mekanum, kontrol PI, arah hadap, *prototype*, kestabilan sistem

Pendahuluan

Kendaraan listrik merupakan kendaraan alternatif yang mengusung hemat

energi dan sedang digencarkan saat ini untuk menggantikan kendaraan berbahan bakar fosil. Banyak dari universitas dan

perusahaan berlomba-lomba dalam membuat kendaraan listrik. Banyak *prototype* kendaraan listrik yang telah diciptakan dengan berbagai macam bentuk maupun kegunaan dengan roda biasa. Pada *prototype* kendaraan listrik menggunakan roda biasa, pergerakan arah *prototype* sangat terbatas sehingga menjadi kendala dalam fungsi navigasi yang membutuhkan manuver ke berbagai arah. Dengan menggunakan roda mekanum maka arah pergerakan *prototype* kendaraan listrik dapat lebih banyak [1].

Dengan roda mekanum maka *prototype* kendaraan listrik dapat dikendalikan ke berbagai arah termasuk arah hadap kendaraan. Permasalahannya adalah pemilihan sistem yang tidak diketahui serta ketidaktepatan metode kendali sistemnya. Untuk itu diperlukan metode kendali yang tepat guna mendapatkan performa yang baik dan mendapatkan respon sistem yang baik [2]. Metode sistem kontrol menggunakan algoritma PID (*Proportional-Integral-Derivative*) bertujuan untuk memperbaiki kinerja sistem karena PID digunakan dalam sebuah sistem *close loop* yang melibatkan *feedback* dari *output* sistem untuk mencapai respon yang diinginkan [3].

Penelitian ini akan mendesain dan menganalisis pengaruh variasi nilai konstanta K_p , K_i , dan variasi kecepatan laju kendaraan *prototype* kendaraan listrik terhadap kestabilan sistem. Sistem yang dibuat adalah sistem kontrol arah hadap *prototype* kendaraan listrik dengan variasi kecepatan menggunakan kendali PI. Diharapkan dengan menggunakan sistem kendali PI adanya pengaruh dan mengurangi eror yang terjadi saat sistem bekerja sehingga dapat memberikan kinerja yang baik pada sistem kontrol arah hadap *prototype* kendaraan listrik ini.

Tinjauan Teoritis

1. Motor BLDC

Motor BLDC (*Brushless Direct Current*) adalah tipe motor dengan cepat mendapatkan popularitas. Sesuai namanya motor BLDC tidak menggunakan sikat

untuk komutasi, sebaliknya motor BLDC secara elektronik untuk komutasinya. Kelebihan motor BLDC dibanding dengan motor DC *brushed* dan motor induksi adalah karakteristik kecepatan dan torsi yang lebih baik, respon dinamis tinggi, efisiensi tinggi, masa pemakaian yang panjang, tanpa suara, rentang kecepatan yang lebih tinggi [4].



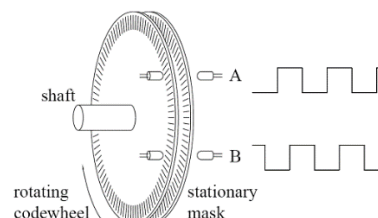
Gambar 1. Motor BLDC

2. Sensor kompas magnetometer (kompas digital)

Sensor kompas digital merupakan modul sensor magnetometer dengan keluaran berupa sudut yang menyatakan arah hadap. Sedangkan sensor magnetometer keluarannya berupa besar medan magnet bumi yang diukur dalam tiga sumbu yang dapat digunakan untuk menentukan sudut arah hadap dengan rumusan tertentu [5].

3. Rotary encoder

Rotary encoder adalah perangkat yang dapat merubah gerakan menjadi urutan sinyal digital. *Rotary encoder* dapat digunakan sebagai sensor posisi, arah, dan kecepatan. *Rotary encoder* terdiri dari kaca atau piringan plastik. Cara kerjanya adalah ketika garis-garis lubang disetiap trek piringan menyela sinar antara pasangan *photoemitter-detector* maka sinyal digital akan dihasilkan [6].



Gambar 2. Sensor *rotary encoder*

4. Sistem Kontrol PID

Sistem kontrol PID (*Proportional-Integral-Derivative*) adalah sistem pengontrolan melalui kombinasi dari tiga

macam kendali, yakni kendali proportional, kendali integral, dan kendali derivative [7]. Kontrol PID ini banyak diterapkan untuk mengatur kestabilan sistem seperti kestabilan suhu, pengaturan intensitas cahaya lampu, dan kecepatan putaran motor [8].

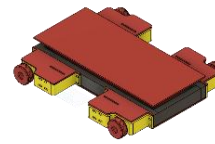
Metode Penelitian

Pada penelitian ini data yang diambil akan dilakukan tanpa beban pada prototype kendaraan listrik. Pada derajat arah hadap kendaraan listrik ini akan bernilai sama pada setiap variabel konstanta PI dan variabel kecepatan. Kecepatan laju kendaraan pada penelitian ini akan divariasikan menjadi (0,5m/s, 1m/s, 1,5m/s, 2m/s, 2,5m/s dan 3m/s). Selanjutnya setiap kecepatan laju kendaraan akan diberi 5 variasi nilai konstanta PI (K_p dan K_i) dengan nilai yang sama antara laju kendaraan satu dengan laju kendaraan yang lain. Untuk pencarian nilai Konstanta PI terbaik akan menggunakan metode *trial and error*. Dalam metode ini akan lebih banyak melakukan eksperimen terhadap penentuan nilai K_p dan K_i . Keunggulan cara ini adalah tidak perlunya untuk mengidentifikasi *plant*, membuat model matematis *plant*, menentukan parameter *plant* dengan grafis ataupun analitis. Caranya adalah cukup dengan cara memberikan variasi nilai konstanta PI hingga di peroleh hasil yang di inginkan [9]. Dengan variasi nilai konstanta PI tersebut akan menghasilkan grafik hubungan antara arah hadap (derajat) kendaraan terhadap waktu pada perangkat komputer.

Alat dan bahan yang digunakan:

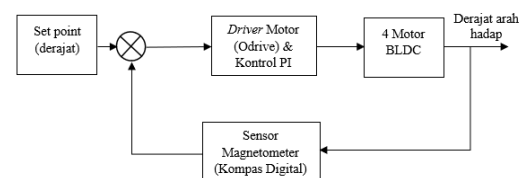
1. Multimeter
2. Komputer/pc
3. Kompas Digital
4. Interface Cable
5. Roda mekanum
6. Odrive (driver motor)
7. Besi Hollow
8. Motor BLDC 1400 KV
9. Baterai Li-po 3500 mAH
10. Charger Baterai
11. Sensor Encoder

Desain *prototype* kendaraan listrik pada penelitian ini:



Gambar 3. Desain *prototype*

1. Kerangka konsep penelitian



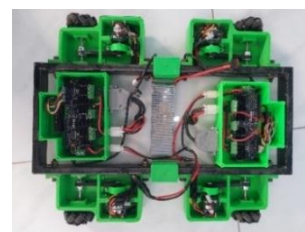
Gambar 4. Kerangka konsep penelitian

Kerangka konsep penelitian yang digunakan merupakan sistem *close loop* yang memiliki beberapa komponen utama seperti Driver motor Odrive sebagai kontrol penggerak motor BLDC dan juga sebagai Kontroler PI, sensor encoder sebagai pendeteksi kecepatan motor, motor BLDC 1400 KV sebagai penggerak kendaraan, kompas digital yang berfungsi untuk membaca derajat kompas dan kendali penggerak kendaraan ke arah yang dibutuhkan, dan laptop sebagai perangkat untuk memonitoring respon sistem secara *real time*.

2. Skema penelitian

1. Menyiapkan Alat

Pada tahap ini menyiapkan kendaraan dengan melakukan pengisian daya terlebih dahulu sampai 12,5 volt, selanjutnya melakukan pengecekan pada konektor elektronik, ketegangan belt, roda mekanum dan kompas digital.

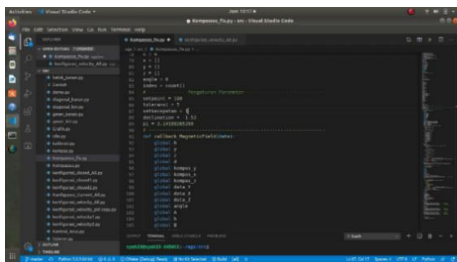


Gambar 5. MGV

2. Penanaman Program

Pada tahap ini melakukan penanaman skrip dengan menggunakan terminal (cmd) pada os ubuntu. Penanaman

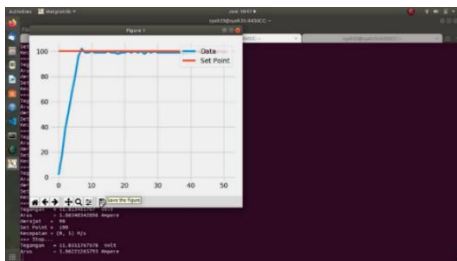
skrip program ditanam pada driver motor Odrive yang merupakan salah satu kontroler motor BLDC [10]. Pertama menjalankan program ROS (*Robot Operating System*) untuk menghubungkan perangkat keras (sensor kompas digital dan driver odrive) [11], dengan sistem operasi komputer. Program yang dijalankan yaitu 'roscore' dan 'rosetopic'. Setelah itu menjalankan program pengaturan Konstanta PI dan pengaturan lain pada kendaraan seperti (pembatasan arus, dan pembatasan kecepatan kendaraan). Setelah itu jalankan skrip pengambilan data dan memasukkan setpoint yang dibutuhkan. Untuk setiap variasi yang digunakan akan diubah dan ditanam ulang program secara berkala dengan menggunakan aplikasi *Visual Studio Code*.



Gambar 6. Interface Visual Studio Code

3. Pengambilan Data

Setelah kendaraan bergerak maka akan muncul grafik parameter kestabilan sistem yaitu derajat terhadap waktu (1/10 detik). Kemudian grafik tersebut yang akan diambil dan dihitung oleh peneliti. Data yang diambil adalah variasi konstanta PI dan kecepatan kendaraan 0,5 m/s sampai 3 m/s secara bergantian.



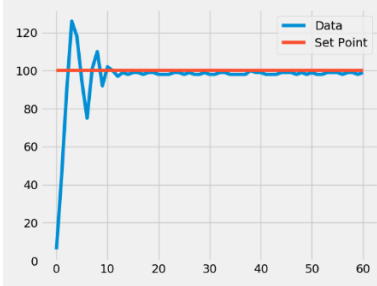
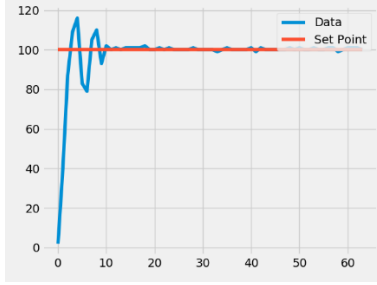
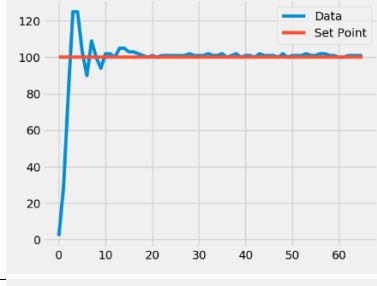
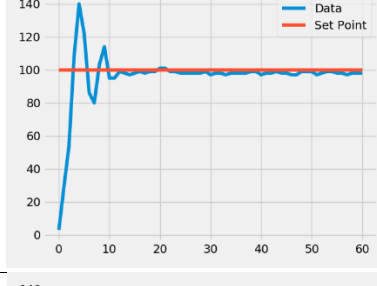
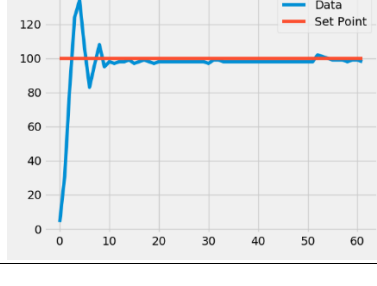
Gambar 7. Teminal (cmd) OS Ubuntu

Hasil dan Pembahasan

Berikut adalah grafik respon arah hadap dengan beberapa variasi nilai

konstanta PI pada kecepatan 3 m/s dengan *setpoint* 100 derajat. Waktu yang ditampilkan grafik dalam respon kestabilan sistem adalah 1/10 detik.

Tabel 1. Grafik Respon Kestabilan Sistem Dengan Kecepatan 3 m/s

Konstanta PI	Grafik
$K_p = 1$ $K_i = 13$	
$K_p = 2$ $K_i = 17$	
$K_p = 1.25$ $K_i = 10$	
$K_p = 1.7$ $K_i = 15$	
$K_p = 2.4$ $K_i = 7$	

Setelah mengambil data penelitian berupa data grafik respon kestabilan sistem maka setelah itu adalah mengubah kedalam data numerik untuk memudahkan data analisis. Data numerik tersebut meliputi nilai *rise time* dalam bentuk satuan detik, *settling time* dalam bentuk satuan detik, *overshot* dalam bentuk presentase dan *error steady state* dalam bentuk presentase.

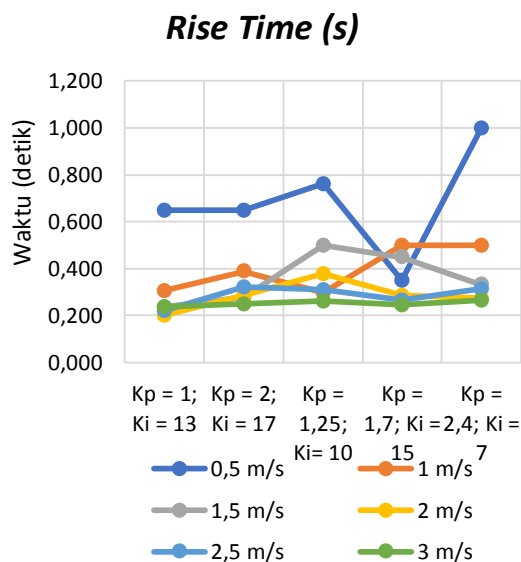
1. Rise Time

Pada pengolahan data *rise time* maka nilai dari setiap *rise time* akan diambil dari setiap nilai konstanta PI dan Kecepatan.

Tabel 2. Nilai Rise Time

Konstanta PI	Rise Time (s)					
	0,5 m/s	1 m/s	1,5 m/s	2 m/s	2,5 m/s	3 m/s
Kp = 1; Ki = 13	0,6	0,3	0,2	0,2	0,2	0,2
Kp = 2; Ki = 17	0,6	0,3	0,2	0,2	0,3	0,2
Kp=1,25; Ki= 10	0,7	0,3	0,5	0,3	0,3	0,2
Kp = 1,7; Ki = 15	0,3	0,5	0,4	0,2	0,2	0,2
Kp = 2,4; Ki = 7	1,0	0,5	0,3	0,2	0,3	0,2

Selanjutnya dibuat grafik hubungan antara konstanta PI dengan kecepatan pada kestabilan sistem *rise time*.



Gambar 8. *rise time* (s)

Pada Gambar 8, diketahui respon kestabilan sistem berupa *rise time* tercepat pada kecepatan 2 m/s dengan nilai sebesar

0,200 detik yaitu pada saat menggunakan konstanta PI dengan Kp = 1 dan Ki = 13. Nilai *rise time* tercepat untuk kecepatan 1 m/s, 1,5 m/s, 2,5 m/s, dan 3 m/s dengan nilai 0,306 detik, 0,238 detik, 0,222 detik dan 0,238 detik yaitu dengan menggunakan nilai konstanta PI yang sama pada nilai *rise time* tercepat yaitu Kp = 1 dan Ki = 13. Pada kecepatan 0,5 m/s *rise time* tercepat dengan nilai 0,350 detik dengan menggunakan konstanta PI yaitu Kp = 1,7 dan Ki = 15.

Sedangkan untuk kestabilan sistem dengan nilai *rise time* terlambat pada kecepatan 0,5; 1; 1,5; 2; 2,5; 3 m/s secara berturut – turut adalah 1,000 detik, 0,500 detik, 0,333 detik, 0,275 detik, 0,314 detik dan 0,264 detik. Pada nilai dengan *rise time* terlambat ini menggunakan konstanta PI dengan nilai Kp = 2,4 dan Ki = 7. Dari hasil tersebut dapat diketahui nilai konstanta PI yang besar dapat menghasilkan nilai *rise time* yang cepat. Sedangkan nilai *rise time* yang lambat adalah saat menggunakan konstanta PI yang kecil. Untuk variasi kecepatan pada salah satu konstanta PI dapat dilihat pada tabel bahwa variasi kecepatan mempengaruhi nilai *rise time* yaitu makin cepat kecepatan yang digunakan maka dapat membuat kecepatan *rise time* makin meningkat, tetapi semua itu juga tergantung dari konstanta PI mana yang digunakan pada sistem.

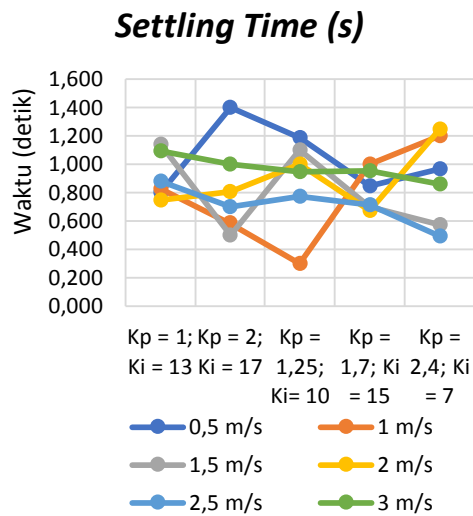
2. Settling Time

Pada pengolahan data *settling time* maka nilai dari setiap *settling time* akan diambil dari setiap nilai konstanta PI dan kecepatan.

Tabel 3. Nilai Settling Time

Konstanta PI	Settling Time (s)					
	0,5 m/s	1 m/s	1,5 m/s	2 m/s	2,5 m/s	3 m/s
Kp = 1; Ki = 13	0,8	0,8	1,1	0,7	0,8	1,0
Kp = 2; Ki = 17	1,4	0,5	0,4	0,8	0,7	1,0
Kp= 1,25; Ki= 10	1,1	0,3	1,1	1,0	0,7	0,9
Kp = 1,7; Ki = 15	0,8	1,0	0,7	0,6	0,7	0,9
Kp = 2,4; Ki = 7	0,9	1,2	0,5	1,2	0,4	0,8

Selanjutnya dibuat grafik hubungan antara konstanta PI dengan kecepatan pada kestabilan sistem *settling time*.



Gambar 9. *Settling Time (s)*

Pada Gambar 9 diketahui respon kestabilan sistem berupa *settling time* tercepat pada kecepatan 1 m/s dengan nilai sebesar 0,300 detik yaitu pada saat menggunakan konstanta PI dengan $K_p = 1,25$ dan $K_i = 10$. Nilai *settling time* untuk kecepatan yang lain berbeda – beda dalam penggunaan konstanta PI yaitu untuk kecepatan 0,5 m/s dengan nilai sebesar 0,800 detik menggunakan konstanta PI dengan $K_p = 1$ dan $K_i = 13$. Setelah itu untuk kecepatan 1,5 m/s dengan nilai sebesar 0,499 detik menggunakan konstanta PI dengan $K_p = 2$ dan $K_i = 17$. Untuk kecepatan 2 m/s dengan nilai sebesar 0,675 detik menggunakan konstanta PI dengan $K_p = 1,7$ dan $K_i = 15$. Untuk kecepatan 2,5 m/s dengan nilai sebesar 0,495 detik menggunakan konstanta PI dengan $K_p = 2,4$ dan $K_i = 7$. Dan untuk kecepatan 3 m/s dengan nilai sebesar 0,864 detik menggunakan konstanta PI dengan $K_p = 2,4$ dan $K_i = 17$.

Sedangkan untuk kestabilan sistem dengan nilai *settling time* terlambat pada kecepatan 0,5 m/s dengan nilai 1,400 detik dimana pada kecepatan ini menggunakan konstanta PI dengan $K_p = 2$ dan $K_i = 17$. Jadi pada saat menggunakan kendaraan pemilihan konstanta PI dan kecepatan berpengaruh pada kestabilan sistem yaitu

settling time dimana cepat tidaknya kendaraan menuju kestabilan setpoint yang sudah ditentukan.

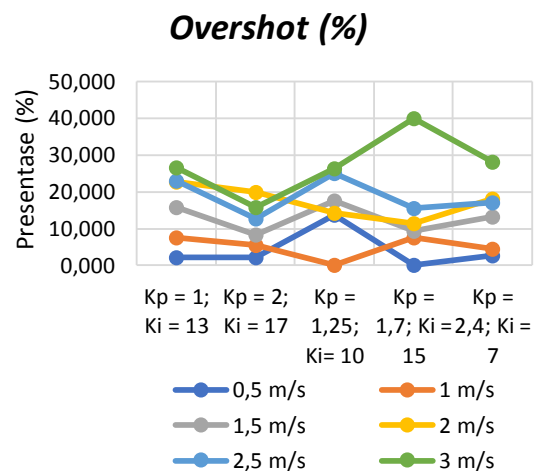
3. Overshot

Pada pengolahan data *overshot* maka nilai dari setiap *overshot* akan diambil dari setiap nilai konstanta PI dan kecepatan.

Tabel 4. Nilai *Overshot*

		<i>Overshot (%)</i>					
Konstanta PI	0,5 m/s	1 m/s	1,5 m/s	2 m/s	2,5 m/s	3 m/s	
$K_p = 1;$ $K_i = 13$	2,12	7,4	15,8	22,7	23,0	26,6	
$K_p = 2;$ $K_i = 17$	2,27	5,5	8,42	20,0	12,7	15,8	
$K_p = 1,2;$ $K_i = 10$	13,7	0,0	17,5	14,2	25,0	26,3	
$K_p = 1,7;$ $K_i = 15$	0,00	7,4	9,30	11,3	15,5	40,0	
$K_p = 2,4;$ $K_i = 7$	2,73	4,5	13,2	18,1	17,0	28,1	

Selanjutnya dibuat grafik hubungan antara konstanta PI dengan kecepatan pada kestabilan sistem *overshot*.



Gambar 10. *Overshot (%)*

Pada Gambar 10, diketahui respon kestabilan sistem berupa *overshot* tertinggi pada kecepatan 3 m/s dengan nilai sebesar 40,000% yaitu pada saat menggunakan konstanta PI dengan $K_p = 1,7$ dan $K_i = 15$. Selanjutnya pada kecepatan 2,5 m/s dan 3 m/s berturut - turut dengan nilai sebesar 25,000% dan 26,315% pada saat menggunakan konstanta PI dengan $K_p = 1,25$ dan $K_i = 10$.

Sedangkan untuk kestabilan sistem dengan nilai *overshot* terendah pada kecepatan 0,5 m/s dengan nilai 0,000%

menggunakan konstanta PI dengan $K_p = 1,7$ dan $K_i = 15$ dan dengan kecepatan 1 m/s dengan nilai *overshot* sebesar 0,000% menggunakan konstanta PI dengan $K_p = 1,25$ dan $K_i = 10$. Pada urutan ketiga nilai *overshot* terendah yaitu kecepatan 0,5 m/s dengan nilai 2,127% menggunakan konstanta PI dengan $K_p = 1$ dan $K_i = 13$. Dari hasil tersebut dapat diketahui nilai konstanta PI yang kecil dapat menghasilkan nilai *overshot* yang kecil. Sedangkan nilai *overshot* yang tinggi adalah saat menggunakan konstanta PI yang besar. Tetapi pemilihan kecepatan kendaraan juga berpengaruh dalam penggunaan konstanta PI dalam sistem. Terdapat beberapa kecepatan yang bernilai tinggi dengan kecepatan dan konstanta PI yang dipilih contohnya dengan $K_p = 1,7$ dan $K_i = 15$ pada kecepatan 3 m/s mendapatkan nilai *overshot* yang tinggi sedangkan jika menggunakan kecepatan yang rendah seperti 0,5 m/s mendapatkan nilai *overshot* yang rendah.

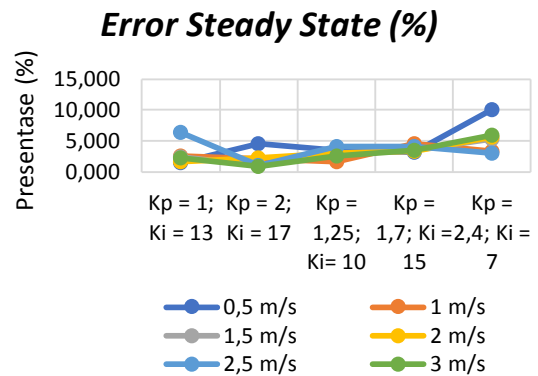
4. Error Steady State

Pada pengolahan data *error steady state* maka nilai dari setiap *error steady state* akan diambil dari setiap nilai konstanta PI dan kecepatan.

Tabel 5. Nilai *Error Steady State*

Konstanta PI	<i>Error Steady State</i> (%)					
	0,5 m/s	1 m/s	1,5 m/s	2 m/s	2,5 m/s	3 m/s
$K_p = 1;$ $K_i = 13$	1,59	2,6	2,4	1,7	6,4	2,3
$K_p = 2;$ $K_i = 17$	4,54	2,2	1,1	2,2	1,1	0,9
$K_p = 1,25;$ $K_i = 10$	3,48	1,6	3,0	3,0	4,2	2,6
$K_p = 1,7;$ $K_i = 15$	3,29	4,5	3,4	3,3	4,0	3,5
$K_p = 2,4;$ $K_i = 7$	10,0	3,4	5,3	5,6	3,0	5,9

Selanjutnya dibuat grafik hubungan antara konstanta PI dengan kecepatan pada kestabilan sistem *error steady state*.



Gambar 11. *Error Steady State* (%)

Pada Gambar 11, diketahui respon kestabilan sistem berupa *error steady state* tertinggi pada kecepatan 0,5 m/s dengan nilai sebesar 10,044% yaitu pada saat menggunakan konstanta PI dengan $K_p = 2,4$ dan $K_i = 7$. Selanjutnya pada kecepatan 2,5 m/s dan 3 m/s berturut – turut dengan nilai sebesar 6,410% dan 5,914% pada saat menggunakan konstanta PI dengan $K_p = 1$ dan $K_i = 13$ untuk 2,5 m/s, sedangkan untuk 3 m/s menggunakan $K_p = 2,4$ dan $K_i = 7$.

Sedangkan untuk kestabilan sistem dengan nilai *error steady state* terendah berturut – turut pada kecepatan 3 m/s, 2,5 m/s dan 1,5 m/s yaitu dengan nilai sebesar 0,975%, 1,162%, dan 1,123%. Kestabilan sistem pada *error steady state* dengan nilai terendah ini menggunakan konstanta PI dengan $K_p = 2$ dan $K_i = 17$. Dari hasil tersebut dapat diketahui nilai konstanta PI yang kecil dapat menghasilkan nilai *error steady state* yang besar. Sebaliknya pada nilai *error steady state* yang rendah adalah saat menggunakan konstanta PI yang besar.

Kesimpulan

Dari Penelitian yang dilakukan dapat disimpulkan bahwa variasi nilai konstanta PI dan variasi kecepatan mempunyai pengaruh terhadap parameter kestabilan sistem yaitu *rise time*, *settling time*, *overshot*, dan *error steady state*. Untuk mendapatkan hasil respon kestabilan sistem terbaik dapat menggunakan nilai konstanta PI $K_p = 2$; dan $K_i = 17$; dimana respon kestabilan sistem untuk kontrol arah hadap

pada setiap kondisi kecepatan memiliki nilai yang cukup baik dengan *rise time* yang cepat, *settling time* yang cepat, *overshot* yang kecil, dan *error steady state* yang kecil dibandingkan dengan nilai konstanta PI yang lainnya dalam penelitian ini.

Referensi

- [1] Priyambudi, A., Firman, B., & Kristiyana, S. (2018). Kendali Kecepatan Motor Pada Empat Roda Omni Menggunakan PID. *Jurnal Teknologi Technosciantia*. 10(2) 209-217.
- [2] Wahyuono, T.A., Widyantara, H., & Wibowo, M.C. (2015). Kendali Robot 4WD Mecanum Wheel Berbasis PID Dengan Menggunakan Arm-Cortex M4. *Journal of Control and Network System*. 4(1) 39-45.
- [3] Diana, D., Amperawan, & Rasyid, J.A. (2017). Implementasi Sensor Kompas HMC5883L Terhadap Gerak Robot Micromouse Dengan Menggunakan Algoritma PID. *Jurnal Teknik Elektro ITP*. 6(2) 120-124.
- [4] Yadamale, P. (2003). Brushless DC (BLDC). Motor Fundamentals. U.S.A. Microchip Technology Inc.
- [5] Noor, E.S., & Rifan, M. (2013). Implementasi Sensor Magnetometer dan Akselerometer Untuk Memonitor Arah Muatan Roket. *Jurusan Teknik Elektro*. 1-6.
- [6] NXR.Northwestern.edu (2012). *Rotary Encoder*.
- [7] Ardiansyah, M.D., dan Rohman, F. (2019). Implementasi dan Analisis Kendali Kecepatan Motor BLDC 1 KW Menggunakan Algoritma PID. *Jurnal ELTEK*. 17(02) 81-93.
- [8] Huda, M.A.N., dan Rohman, F. (2019). Implementasi dan Analisis Kendali Kecepatan Pada Motor BLDC 1 KW Tanpa Beban Menggunakan Algoritma PID. *Jurnal ELTEK*. 17(02) 94-106.
- [9] Huda, M.A.N. (2019). *Implementasi dan Analisis Kendali Kecepatan Pada Motor BLDC 1 KW Tanpa Beban Menggunakan Algoritma PID*. Skripsi. Jurusan Teknik Mesin. Politeknik Negeri Malang : Malang.
- [10] Odriverobotics.com. (n.d.). *ODriver High Performance Motor Control*.
- [11] Jalil, A. (2018). Robot Operating System (ROS) dan Gazebo Sebagai Media Pembelajaran Robot Interaktif. *Jurnal Ilmiah*. 10(3) 284-289.