

SISTEM KENDALI KECEPATAN PADA AUTONOMOUS ELECTRIC VEHICLE BERBASIS PENGENDALI PID DENGAN MASUKAN MENGGUNAKAN RFID

¹*Caroline, ¹M. Zaid Haritsyah, ¹M. Fauzan, ¹Sariman, ¹Hermawati, ¹Suci Dwijayanti, ¹Bhakti Yudho Suprpto, ¹Zaenal Husin

¹Teknik Elektro, Universitas Sriwijaya, Palembang

Corresponding author: caroline@ft.unsri.ac.id

Abstrak— *Autonomous electric vehicle* merupakan sebuah kendaraan cerdas yang menggunakan sistem *self-driving*. Salah satu hal penting dalam sistem *self-driving* adalah sistem *speed-control otomatis*, yang merupakan sebuah sistem yang bertugas untuk mengatur kecepatan kendaraan secara otomatis saat melakukan deselerasi maupun akselerasi. Penelitian ini bertujuan untuk mengatur sistem *speed-control* agar berjalan dengan baik, untuk itu diperlukan sebuah inputan untuk mengatur *setpoint* kecepatan yang diinginkan dan juga sebuah pengendali untuk mempertahankan kecepatan tersebut. Pada penelitian ini digunakan RFID (*Radio Frequency Identification*) sebagai input *setpoint* dengan menetapkan nomor tag pada kecepatan tertentu dan metode kontrol PID akan mempertahankan kecepatan sesuai dengan *setpoint* yang ada pada RFID. Metode kontrol PID (*Proportional Integral Derivative*) dipilih karena metode ini dapat menjaga kestabilan kecepatan kendaraan listrik. Pada penentuan-parameter gain pada PID dilakukan dengan metode *trial and error* yang dilakukan dengan 4 kali pengujian. Pengujian-pengujian ini dilakukan di 7 titik dengan 13 tag RFID untuk kondisi lurus dan berbelok. Setiap RFID memiliki inputan *setpoint* yang berbeda, yaitu 60, 70, 80, dan 120 rpm. Hasil pengujian terbaik adalah dengan nilai $k_p = 1$, $k_i = 0.1$ dan $k_d = 2$ dinilai menjadi yang paling stabil dalam mengatur kecepatan kendaraan hal ini dikarenakan pengujian dengan nilai $k_p = 1$, $k_i = 0.1$ dan $k_d = 2$ memiliki *rise time*, *delay time* dan *settling time* yang cukup cepat serta *overshoot* yang cukup kecil sehingga memiliki kestabilan terbaik dibanding pengujian yang lainnya.

Kata Kunci: *Autonomous Electric Vehicle*, RFID, Kendali Kecepatan, PID

Abstract - *Autonomous electric vehicle* is an intelligent vehicle that uses a *self-driving* system. An important thing in *self-driving* system is the automatic *speed-control* system, which is a system assigned to adjust the speed of the vehicle automatically when decelerating or accelerating. This research aims to adjust the *speed-control* system to run properly, for that we need an input to set the desired speed *setpoint* and also a controller to maintain that speed. On our research, RFID (*Radio Frequency Identification*) is used as the *setpoint's* input by setting the tag number at a certain speed and the PID control method will maintain speed according to the *setpoint* on RFID. PID (*Proportional Integral Derivative*) control method was chosen because this method can maintain a stable speed of electric vehicle. The determination of the gain parameter on the PID is carried out by the *trial and error* method which is carried out with 4 tests. These tests were carried out at 7 points with 13 RFID tags for straight and turn conditions. Each RFID has a different *setpoint* input, namely 60, 70, 80, and 120 RPM. The best result is the value of $K_p = 1$, $K_i = 0.1$ and $K_d = 2$ which is considered to be the most stable in regulating vehicle speed, this is because tests with values of $K_p = 1$, $K_i = 0.1$ and $K_d = 2$ have *rise time*, *delay time* and *The settling time* is quite fast and the *overshoot* is small enough so that it has the best stability compared to other tests.

Keywords: *Autonomous Electric Vehicle*, RFID, Speed Control, PID

PENDAHULUAN

Autonomous electric vehicle merupakan sebuah kendaraan cerdas yang menggunakan sistem *self-driving* [1]. *Self-driving* merupakan salah satu jenis kontrol *autonomous car* yang berfungsi memudahkan manusia dalam berkendara secara otomatis tanpa campur tangan pengemudi. *Self-driving* berguna ketika pengemudi

mengalami kondisi-kondisi tertentu yang harus melepaskan kemudi seperti mengangkat telpon, mengambil sesuatu di dashboard, dan lain sebagainya [2]. Salah satu hal penting dalam sistem *self-driving* adalah mengenai sistem *speed-control* otomatis.

Sistem *speed-control* otomatis sendiri merupakan sebuah sistem yang bertugas untuk mengatur laju atau kecepatan kendaraan secara otomatis. Sistem ini

bermanfaat dalam penerapatan *autonomous electric vehicle* untuk mengatur kecepatan kendaraan baik itu saat melakukan deselerasi yaitu pengurangan kecepatan kendaraan maupun saat melakukan akselerasi yaitu penambahan kecepatan kendaraan. Penelitian mengenai sistem *speed-control* telah banyak dilakukan salah satunya adalah menggunakan RFID [3][4].

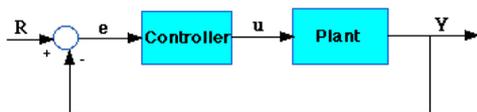
Radio Frequency Identification atau biasa dikenal dengan sebutan RFID merupakan sebuah metode pengidentifikasian otomatis dengan sistem frekuensi radio yang menggunakan sarana yang disebut label RFID atau transponder (tag) untuk menyimpan dan mengambil data jarak jauh [5]. RFID sendiri memiliki beberapa keunggulan diantaranya harganya yang murah dan mudah diimplementasikan ke berbagai sistem kontrol termasuk dalam mengontrol kecepatan kendaraan *autonomous electric vehicle*.

Dalam penelitian terhadap *speed-control* pada *autonomous electric vehicle* telah dilakukan dalam berbagai metode diantaranya adalah menggunakan metode Fuzzy[4] dan metode PID[2][6][7].

Berdasarkan penelitian-penelitian sebelumnya, pada tugas akhir ini akan dilakukan penelitian menggunakan metode *proportional integral derivative* (PID). Metode ini dipilih karena dapat menjaga kestabilan kecepatan kendaraan listrik [6] serta memiliki struktur yang lebih sederhana dan menghasilkan kinerja yang baik [7].

PROPORTIONAL INTEGRAL DERIVATIVE (PID)

Proportional Integral Derivative (PID) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Pengendali ini merupakan gabungan dari pengendali *proportional* (P), *integral* (I), dan *derivative* (D). Berikut ini merupakan blok diagram dari sistem pengendali dengan untai tertutup (*closed loop*) dapat dilihat pada gambar 1.



Gambar 1. Diagram blok closed loop[8]

Dimana plant merupakan suatu sistem yang akan dikendalikan, kontroler merupakan pengendali yang memberikan respon untuk memperbaiki respon dan e adalah *error* dari pengukuran sensor. Pada setiap parameter PID memiliki konstanta sendiri yaitu Kp (konstanta *Proportional*), Ki (Konstanta *Integral*) dan Kd

(Konstanta *Derivative*). Secara umum persamaannya dapat ditulis sebagai berikut

$$U(t) = K_p(e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{d e(t)}{dt})(1)$$

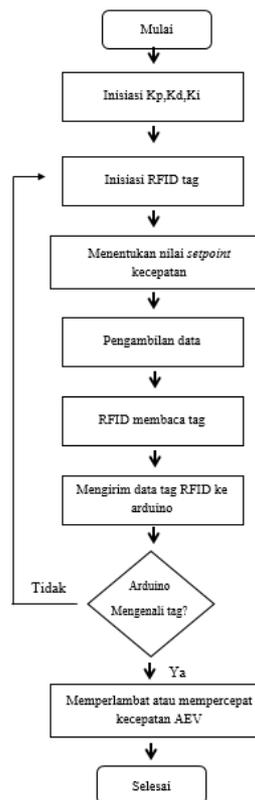
TABLE I. KARAKTERISTIK MASING-MASING PENGENDALI

CL Response	Rise Time	Overshoot	Settling Time	S-S Error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	Small Change

METODE PENELITIAN

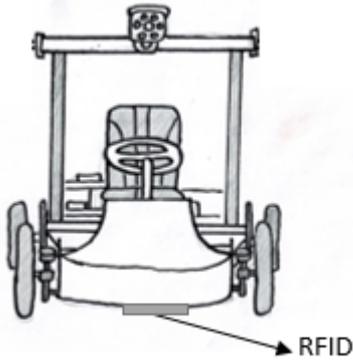
A. Perancangan Sistem

Dalam perancangan sistem pada penelitian ini dibagi menjadi dua, meliputi perancangan *software* dan perancangan *hardware*. Sistem yang akan dibuat dapat dilihat pada gambar 2 berikut.



Gambar 2. Flowchart penelitian

Dalam perancangan hardware dalam penelitian ini ada beberapa perangkat keras yang digunakan untuk mengatur kecepatan, pada *autonomous electric vehicle* yang dapat dilihat pada gambar 3.



Gambar 3. Rancangan posisi hardware yang akan digunakan

B. Perancangan Sistem

Sebelum dilakukan pengujian sistem akan dilakukan pengambilan data tersebut dahulu dengan melihat grafik kecepatan *autonomous electric vehicle* sebelum dan sesudah RFID Reader membaca tag. Tag RFID akan disebar diberbagai lokasi yang telah ditentukan di lingkungan Universitas Sriwijaya kampus Indralaya.

Selanjutnya dilakukan pengujian sistem yang bertujuan untuk mengetahui keberhasilan maupun kesalahan yang ada terhadap sistem yang telah dirancang. Dalam proses uji *autonomous electric vehicle* akan melewati tag RFID yang telah ditentukan sehingga akan terjadi perlambatan dan percepatan pada *autonomous electric vehicle*. Dalam proses pengujian ini juga akan diketahui seberapa besar *overshoot, error, steady state* dan *rise time* yang menjadi parameter sistem yang dirancang telah bekerja dengan baik atau tidak.

HASIL DAN PEMBAHASAN

A. Penentuan Titik Tag RFID dan *Setpoint* Kecepatan Kendaraan

Percobaan ini akan dilakukan dengan 7 titik dengan 13 tag RFID yang berbeda, dimana kendaraan *autonomous electric vehicle* akan melewati ketujuh titik lalu saat kendaraan mendeteksi tag maka kendaraan akan mempercepat atau mengurangi kecepatannya sesuai dengan *setpoint* yang ada pada masing-masing tag tersebut. Masing-masing titik posisi tag RFID dapat dilihat pada gambar 4 dan *setpoint* kecepataannya saat kendaraan mendeteksi tag dapat dilihat pada Tabel II.



Gambar 4. Peta Map Titik Pemasangan Tag RFID

TABLE II. PENENTUAN TITIK POSISI TAG RFID DAN *SETPOINT* KECEPATAN KENDARAAN

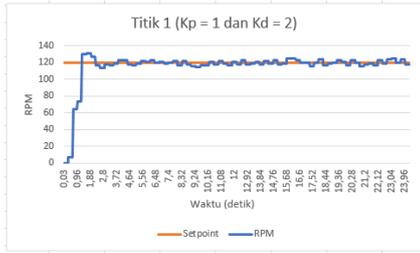
No	Titik Posisi Tag RFID	Tag RFID	<i>Setpoint</i> Kecepatan
1	Titik 1	Tag 1	120 RPM
2	Titik 2	Tag 2	80 RPM
		Tag 3	120 RPM
3	Titik 3	Tag 4	70 RPM
		Tag 5	120 RPM
4	Titik 4	Tag 6	80 RPM
		Tag 7	120 RPM
5	Titik 5	Tag 8	70 RPM
		Tag 9	120 RPM
6	Titik 6	Tag 10	60 RPM
		Tag 11	120 RPM
7	Titik 7	Tag 12	70 RPM
		Tag 13	120 RPM

B. Pengujian Sistem dengan Menggunakan Pengendali PID

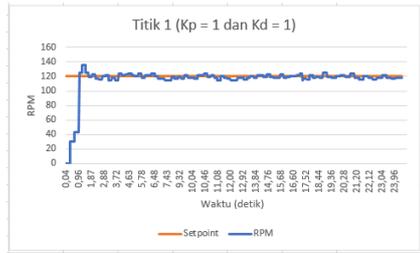
Pengujian sistem dilakukan dengan menggunakan kontroler PID, dengan menentukan parameter K_p , K_i , dan K_d berdasarkan metode *trial and error*. Pengujian dilakukan sebanyak 4 kali, pengujian pertama menggunakan nilai $K_p = 1$ dan $K_d = 2$, pengujian kedua dengan nilai $K_p = 1$ dan $K_d = 1$, pengujian ketiga dengan nilai $K_p = 2$ dan pengujian 4 dengan nilai $K_p = 1$, $K_i = 0.1$ dan $K_d = 2$. Pengujian-pengujian ini dilakukan di 7 titik berbeda yang terbagi menjadi 13 tag RFID, dimana masing-masing tag memiliki *setpoint* kecepatannya sendiri sehingga sistem akan menyesuaikan kecepatan kendaraan dengan *setpoint* tersebut. Adapun parameter-parameter yang digunakan untuk menentukan performansi sistem dalam domain waktu meliputi *delay time, rise time, peak time, overshoot, dan settling time* [9].

C. Pengujian Sistem di Titik 1

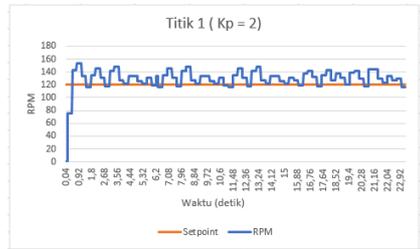
Pada pengujian di titik 1, kendaraan akan mendeteksi tag 1 yang memiliki *setpoint* kecepatan 120 RPM sehingga saat kendaraan mendeteksi tag tersebut maka kendaraan akan menyesuaikan kecepatan sesuai dengan *setpoint* yang ada. Hasil dari pengujian tersebut dapat dilihat pada gambar 5, gambar 6, gambar 7 dan gambar 8 serta Tabel III.



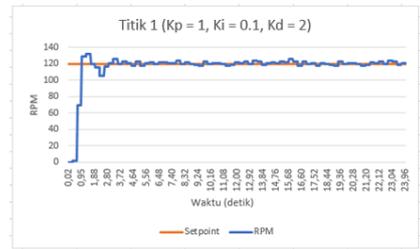
Gambar 5. Grafik Pengujian Pertama di Titik 1 dengan $K_p = 1$ dan $K_d = 2$



Gambar 6. Grafik Pengujian kedua di Titik 1 dengan $K_p = 1$ dan $K_d = 1$



Gambar 7. Grafik Pengujian Ketiga di Titik 1 dengan $K_p = 2$



Gambar 8. Grafik Pengujian Keempat di Titik 1 dengan $K_p = 1$, $K_i = 0.1$ dan $K_d = 2$

TABLE III. PERFORMANSI SISTEM DI TITIK 1

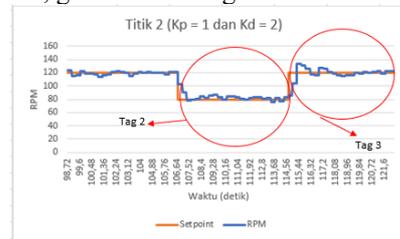
Parameter	Tag 1			
	Pengujian 1 $K_p = 1$ dan $K_d = 2$	Pengujian 2 $K_p = 1$ dan $K_d = 1$	Pengujian 3 $K_p = 2$	Pengujian 4 $K_p = 1$, $K_i =$ 0.1 dan $K_d = 2$
Delay Time	1,01 detik	0,96 detik	0,36 detik	0,66 detik
Rise Time	1,28 detik	1 detik	0,64 detik	0,95 detik
Peak Time	1,64 detik	1,23 detik	0,8 detik	1,32 detik
Overshoot	9,44%	12,77%	27,77%	10%
Settling time	2,92 detik	3,88 detik	-	3,56 detik

Berdasarkan keempat grafik yang ditunjukkan oleh gambar 5, gambar 6, gambar 7 dan gambar 8 serta Tabel III pengujian 4 memiliki *overshoot* dan *settling time* yang cukup kecil yaitu *overshoot* bernilai 10% dan *settling time* bernilai 3,56 detik selain hal itu juga memiliki *delay time*,

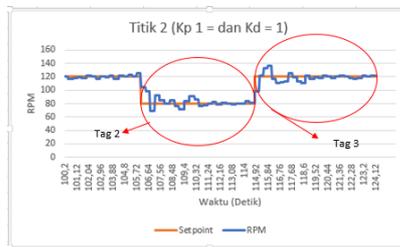
rise time dan *peak time* yang juga cukup kecil yaitu *delay time* bernilai 0,66 detik, *rise time* bernilai 0,95 detik dan pada *peak time* bernilai 1,32 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada titik 1

D. Pengujian Sistem di Titik 2

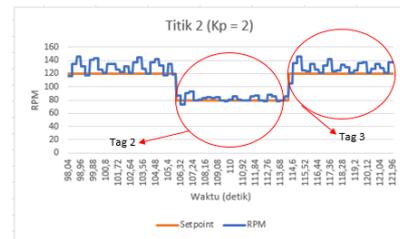
Pada pengujian di titik kedua, kendaraan akan mendeteksi tag 2 yang memiliki *setpoint* kecepatan 80 RPM sehingga kendaraan yang tadinya memiliki kecepatan 120 RPM sesuai dengan tag 1 di titik 1 akan mengalami deselerasi hingga menyesuaikan dengan *setpoint* di tag 2 selanjutnya kendaraan akan mendeteksi tag 3 yang memiliki *setpoint* 120 RPM sehingga kendaraan akan mengalami akselerasi dan akan menyesuaikan kecepatan sesuai dengan *setpoint* tag 3. Hasil dari pengujian di titik 2 dapat dilihat pada gambar 9, gambar 10, gambar 11 dan gambar 12 serta Tabel IV.



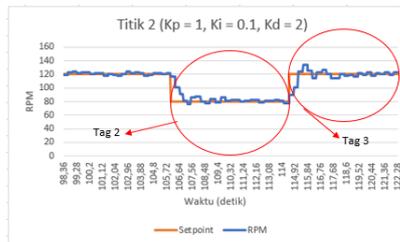
Gambar 9. Grafik Pengujian Pertama di Titik 2 dengan $K_p = 1$ dan $K_d = 2$



Gambar 10. Grafik Pengujian kedua di Titik 2 dengan $K_p = 1$ dan $K_d = 1$



Gambar 11. Grafik Pengujian ketiga di Titik 2 dengan $K_p = 2$



Gambar 12. Grafik Pengujian keempat di Titik 2 dengan $K_p = 1$, $K_i = 0.1$ dan $K_d = 2$

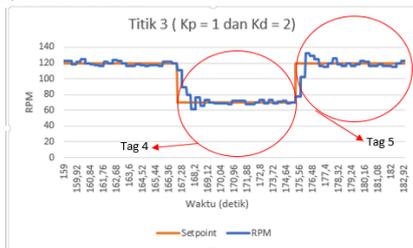
TABLE IV. PERFORMANSI SISTEM DI TITIK 2

Parameter	Tag 2				Tag 3			
	Pengujian 1 Kp = 1 dan Kd = 2	Pengujian 2 Kp = 1 Dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0.1 dan Kd = 2	Pengujian 1 Kp = 1 dan Kd = 2	Pengujian 2 Kp = 1 Dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0.1 dan Kd = 2
Delay Time	106,73 detik	106,4 detik	106,04 detik	106,36 detik	115 detik	115,16 detik	114,32 detik	114,84 detik
Rise Time	107 detik	106,76 detik	106,32 detik	106,92 detik	115,28 detik	115,2 detik	114,6 detik	115,16 detik
Peak Time	108,32 detik	107,16 detik	107 detik	107,8 detik	115,32 detik	113,54 detik	114,96 detik	115,52 detik
Overshoot	8,75%	10%	15,8%	9,58%	11,6%	13,89%	21,94%	12,22%
Settling time	109,9 detik	110,32 detik	-	109,96 detik	117,56 detik	119,04 detik	-	118,16 detik

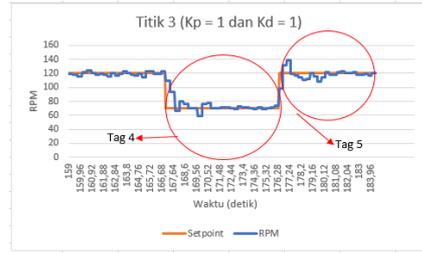
Berdasarkan keempat grafik yang ditunjukkan oleh gambar 9, gambar 10, gambar 11 dan gambar 12 serta Tabel IV pergerakan kecepatan kendaraan saat mendeteksi tag 2 akan mengalami deselerasi menjadi disekitar 80 RPM sesuai dengan *setpoint* yang ada. Dalam performansi saat kendaraan mengalami deselerasi pengujian 4 memiliki *delay time*, *rise time*, *peak time* dan *settling time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu *delay time* bernilai 106,36 detik, *rise time* bernilai 106,32 detik, *peak time* bernilai 107 detik, *overshoot* bernilai 9,58% dan *settling time* bernilai 109,96 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 2. Lalu saat kendaraan mendeteksi tag 3 maka kecepatan kendaraan akan mengalami akselerasi hingga mencapai disekitar 120 RPM sesuai *setpoint* pada tag 3. Dalam performansi saat kendaraan mengalami deselerasi pengujian 4 memiliki *delay time*, *rise time* dan *settling time* yang cukup cepat serta *overshoot* yang cukup kecil yaitu *delay time* bernilai 114,84 detik, *rise time* bernilai 115,16 detik, *overshoot* bernilai 12,22% dan *settling time* bernilai 118,16 detik Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 3.

E. Pengujian Sistem di Titik 3

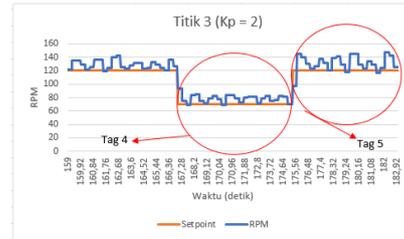
Pada pengujian di titik 3, kendaraan akan mendeteksi tag 4 yang memiliki *setpoint* kecepatan 70 RPM sehingga kendaraan yang tadinya memiliki kecepatan 120 RPM sesuai dengan tag 3 di titik 2 akan mengalami deselerasi hingga menyesuaikan dengan *setpoint* di tag 4 selanjutnya kendaraan akan mendeteksi tag 5 yang memiliki *setpoint* 120 RPM sehingga kendaraan akan mengalami akselerasi dan akan menyesuaikan kecepatan sesuai dengan *setpoint* tag 5. Hasil dari pengujian di titik 3 dapat dilihat pada gambar 13, gambar 14, gambar 15 dan gambar 16 serta Tabel V.



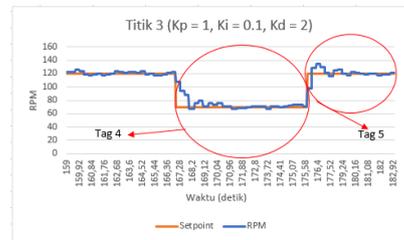
Gambar 13. Grafik Pengujian Pertama di Titik 3 dengan Kp = 1 dan Kd = 2



Gambar 14. Grafik Pengujian Kedua di Titik 3 dengan Kp = 1 dan Kd = 1



Gambar 15. Grafik Pengujian Ketiga di Titik 3 dengan Kp = 2



Gambar 16. Grafik Pengujian Keempat di Titik 3 dengan Kp = 1, Ki = 0.1 dan Kd = 2

TABLE V. PERFORMANSI SISTEM DI TITIK 3

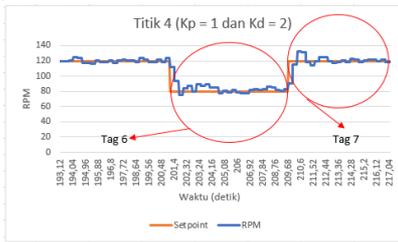
Parameter	Tag 4				Tag 5			
	Pengujian 1 Kp = 1 dan Kd = 2	Pengujian 2 Kp = 1 dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0.1 dan Kd = 2	Pengujian 1 Kp = 1 dan Kd = 2	Pengujian 2 Kp = 1 dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0.1 dan Kd = 2
Delay Time	167,28 detik	167,36 detik	167 detik	167,2 detik	175,68 detik	176,4 detik	175,36 detik	175,64 detik
Rise Time	167,92 detik	167,8 detik	167,28 detik	167,63 detik	175,96 detik	176,68 detik	175,64 detik	175,9 detik
Peak Time	168,32 detik	168,2 detik	168,28 detik	168,68 detik	176 detik	177,04 detik	175,68 detik	176,28 detik
Overshoot	9,04%	14,28%	21,9%	12,85%	10,27%	15,55%	21,1%	11,94%
Settling time	168,92 detik	170,76 detik	-	171,08	178,24 detik	180,56 detik	-	179,2 detik

Berdasarkan keempat grafik yang ditunjukkan oleh gambar 13, gambar 14, gambar 15 dan gambar 16 serta Tabel V pergerakan kecepatan kendaraan saat mendeteksi tag 4 akan mengalami deselerasi menjadi disekitar 70 RPM sesuai dengan *setpoint* yang ada. Dalam performansi saat kendaraan mengalami deselerasi pengujian 1 memiliki *overshoot* dan *settling time* yang lebih kecil dibanding 3 performansi lainnya yaitu untuk *overshoot* 9,04% dan *settling time* 168,92 detik. Hal ini menunjukkan pengujian 1 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 4. Lalu saat kendaraan mendeteksi tag 5 maka kecepatan kendaraan akan mengalami akselerasi hingga mencapai disekitar 120 RPM sesuai *setpoint* pada tag 5. Dalam performansi saat kendaraan mengalami akselerasi pengujian 4 memiliki *delay time*, *rise time*, *settling time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 175,64 detik, *rise time* bernilai 175,9 detik, *overshoot* bernilai 11,94% dan *settling time* bernilai 179,2

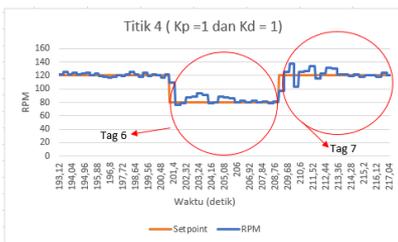
detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 5.

F. Pengujian Sistem di Titik 4

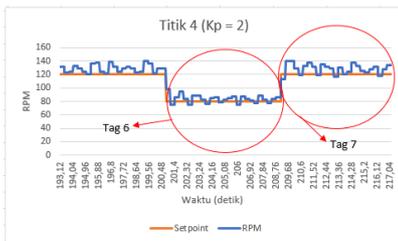
Pada pengujian di titik 4, kendaraan akan mendeteksi tag 6 yang memiliki *setpoint* kecepatan 80 RPM sehingga kendaraan yang tadinya memiliki kecepatan 120 RPM sesuai dengan tag 5 di titik 3 akan mengalami deselerasi hingga menyesuaikan dengan *setpoint* di tag 4 selanjutnya kendaraan akan mendeteksi tag 7 yang memiliki *setpoint* 120 RPM sehingga kendaraan akan mengalami akselerasi dan akan menyesuaikan kecepatan sesuai dengan *setpoint* tag 7. Hasil dari pengujian di titik 4 dapat dilihat pada gambar 17, gambar 18, gambar 19 dan gambar 20 serta Table VI.



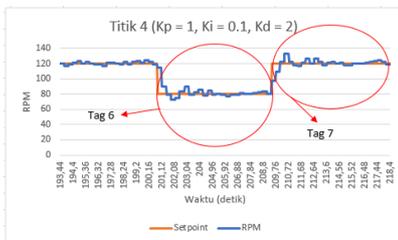
Gambar 17. Grafik Pengujian Pertama di Titik 4 dengan $K_p = 1$ dan $K_d = 2$



Gambar 18. Grafik Pengujian Kedua di Titik 4 dengan $K_p = 1$ dan $K_d = 1$



Gambar 19. Grafik Pengujian Ketiga di Titik 4 dengan $K_p = 2$



Gambar 20. Grafik Pengujian Keempat di Titik 4 dengan $K_p = 1$, $K_i = 0.1$ dan $K_d = 2$

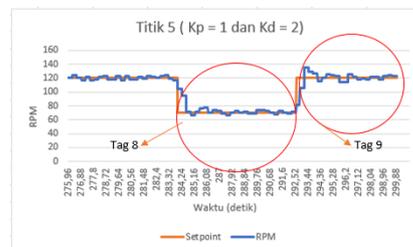
TABLE VI. PERFORMANSI SISTEM DI TITIK 4

Parameter	Tag 6				Tag 7			
	Pengujian 1 $K_p = 1$ dan $K_d = 2$	Pengujian 2 $K_p = 1$ dan $K_d = 1$	Pengujian 3 $K_p = 2$	Pengujian 4 $K_p = 1, K_i = 0,01$ dan $K_d = 2$	Pengujian 1 $K_p = 1$ dan $K_d = 1$	Pengujian 2 $K_p = 1$ dan $K_d = 1$	Pengujian 3 $K_p = 2$	Pengujian 4 $K_p = 1, K_i = 0,01$ dan $K_d = 2$
Delay Time	201,44 detik	201,48 detik	200,08 detik	201,08 detik	210 detik	209,4 detik	209,16 detik	208,72 detik
Rise Time	201,76 detik	201,52 detik	201,12 detik	201,52 detik	210,32 detik	209,44 detik	209,44 detik	210,08 detik
Peak Time	203,68 detik	203,12 detik	201,76 detik	202,88 detik	210,36 detik	209,84 detik	209,48 detik	210,44 detik
Overshoot	11,66%	17,5%	18,75%	12,5%	10,55%	14,72%	16,94%	10,83%
Settling time	204,64 detik	205,84 detik	-	204,96 detik	211,64 detik	211,28 detik	-	213 detik

Berdasarkan keempat grafik yang ditunjukkan oleh gambar 17, gambar 18, gambar 19 dan gambar 20 serta Table VI pergerakan kecepatan kendaraan saat mendeteksi tag 6 akan mengalami deselerasi menjadi disekitar 80 RPM sesuai dengan *setpoint* yang ada. Dalam performansi saat kendaraan mengalami deselerasi pengujian 4 memiliki *delay time*, *rise time* dan *peak time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 201,08 detik, *rise time* bernilai 201,52 detik, *peak time* bernilai 202,88 detik, *overshoot* bernilai 12,5% dan *settling time* bernilai 204,96 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 6. Lalu saat kendaraan mendeteksi tag 7 maka kecepatan kendaraan akan mengalami akselerasi hingga mencapai disekitar 120 RPM sesuai *setpoint* pada tag 7. Dalam performansi saat kendaraan mengalami akselerasi pengujian 4 memiliki *delay time*, *rise time*, *settling time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 209,72 detik, *rise time* bernilai 210,08 detik, *overshoot* bernilai 10,83% dan *settling time* bernilai 213 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 7

G. Pengujian Sistem di Titik 5

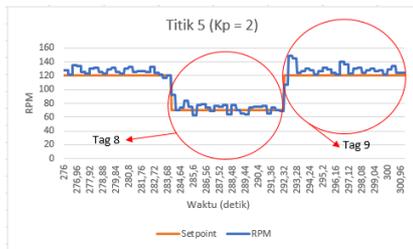
Pada pengujian di titik 5, kendaraan akan mendeteksi tag 8 yang memiliki *setpoint* kecepatan 70 RPM sehingga kendaraan yang tadinya memiliki kecepatan 120 RPM sesuai dengan tag 7 di titik 4 akan mengalami deselerasi hingga menyesuaikan dengan *setpoint* di tag 8 selanjutnya kendaraan akan mendeteksi tag 9 yang memiliki *setpoint* 120 RPM sehingga kendaraan akan mengalami akselerasi dan akan menyesuaikan kecepatan sesuai dengan *setpoint* tag 9. Hasil dari pengujian di titik 5 dapat dilihat pada gambar 21, gambar 22, gambar 23 dan gambar 24 serta Table VII.



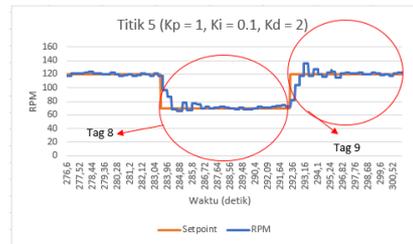
Gambar 21. Grafik Pengujian Pertama di Titik 5 dengan $K_p = 1$ dan $K_d = 2$



Gambar 22. Grafik Pengujian Kedua di Titik 5 dengan $K_p = 1$ dan $K_d = 1$



Gambar 23. Grafik Pengujian Ketiga di Titik 5 dengan $K_p = 2$



Gambar 24. Grafik Pengujian Keempat di Titik 5 dengan $K_p = 1$, $K_i = 0.1$, $K_d = 2$

TABLE VII. PERFORMANSI SISTEM DI TITIK 5

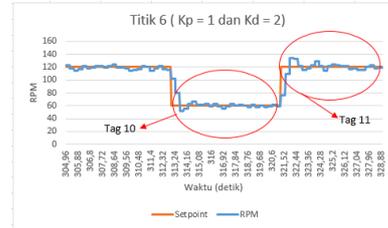
Parameter	Tag 8				Tag 9			
	Pengujian 1 $K_p = 1$ dan $K_d = 2$	Pengujian 2 $K_p = 1$ dan $K_d = 1$	Pengujian 3 $K_p = 2$	Pengujian 4 $K_p = 1, K_i = 0.1$ dan $K_d = 2$	Pengujian 1 $K_p = 1$ dan $K_d = 2$	Pengujian 2 $K_p = 1$ dan $K_d = 1$	Pengujian 3 $K_p = 2$	Pengujian 4 $K_p = 1, K_i = 0.1$ dan $K_d = 2$
Delay Time	284,28 detik	283,96 detik	283,96 detik	283,92 detik	292,33 detik	293,64 detik	292,33 detik	292,63 detik
Rise Time	284,6 detik	284,52 detik	284,31 detik	285,05 detik	293,16 detik	293,68 detik	292,6 detik	293,04 detik
Peak Time	285,88 detik	284,76 detik	284,92 detik	285,08 detik	293,2 detik	294,08 detik	292,64 detik	293,09 detik
Overshoot	11,4%	12,38%	19,04%	11,42%	12,7%	16,6%	23,6%	13,33%
Settling time	286,2 detik	287,16 detik	-	286,44 detik	296,4 detik	296,96 detik	-	296,55 detik

Berdasarkan keempat grafik yang ditunjukkan oleh gambar 21, gambar 22, gambar 23 dan gambar 24 serta Table VII pergerakan kecepatan kendaraan saat mendeteksi tag 8 akan mengalami deselerasi menjadi disekitar 70 RPM sesuai dengan *setpoint* yang ada. Dalam performansi saat kendaraan mengalami deselerasi pengujian 4 memiliki *delay time*, *rise time*, *peak time* dan *settling time* yang cukup cepat serta *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 283,92 detik, *rise time* bernilai 284,27 detik, *peak time* bernilai 285,05, *overshoot* bernilai 11,42% dan *settling time* bernilai 286,44 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 8. Lalu saat kendaraan mendeteksi tag 9 maka kecepatan kendaraan akan mengalami akselerasi hingga mencapai disekitar 120 RPM sesuai *setpoint* pada tag 9. Dalam performansi saat kendaraan mengalami akselerasi pengujian 4 memiliki *delay time*, *rise time*, *peak time*,

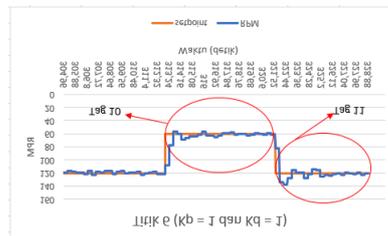
settling time yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 292,63 detik, *rise time* bernilai 293,04 detik, *peak time* bernilai 293,0, *overshoot* bernilai 13,33% dan *settling time* bernilai 296,72 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 9.

H. Pengujian Sistem di Titik 6

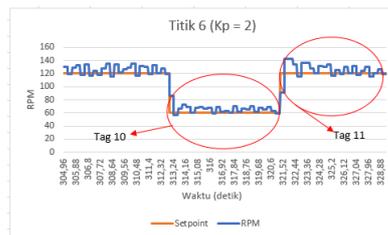
Pada pengujian di titik 6, kendaraan akan mendeteksi tag 10 yang memiliki *setpoint* kecepatan 60 RPM sehingga kendaraan yang tadinya memiliki kecepatan 120 RPM sesuai dengan tag 9 di titik 5 akan mengalami deselerasi hingga menyesuaikan dengan *setpoint* di tag 10 selanjutnya kendaraan akan mendeteksi tag 11 yang memiliki *setpoint* 120 RPM sehingga kendaraan akan mengalami akselerasi dan akan menyesuaikan kecepatan sesuai dengan *setpoint* tag 11. Hasil dari pengujian di titik 6 dapat dilihat pada gambar 25, gambar 26, gambar 27 dan gambar 28 serta Table VIII.



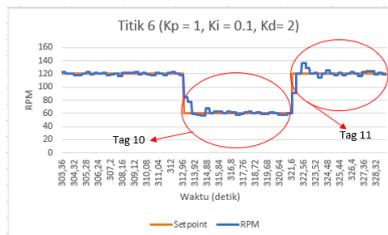
Gambar 25. Grafik Pengujian Pertama di Titik 6 dengan $K_p = 1$ dan $K_d = 2$



Gambar 26. Grafik Pengujian Kedua di Titik 6 dengan $K_p = 1$ dan $K_d = 1$



Gambar 27. Grafik Pengujian Ketiga di Titik 6 dengan $K_p = 2$



Gambar 28. Grafik Pengujian Keempat di Titik 6 dengan $K_p = 1$, $K_i = 0.1$ dan $K_d = 2$

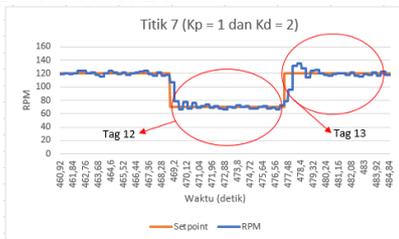
TABLE VIII. PERFORMANSI SISTEM DI TITIK 6

Parameter	Tag 10				Tag 11			
	Pengujian 1 Kp = 1 dan Kd = 2	Pengujian 2 Kp = 1 dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0,1 dan Kd = 2	Pengujian 1 Kp = 1 Kd = 1	Pengujian 2 Kp = 1 dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0,1 dan Kd = 2
Delay Time	313,24 detik	313,24 detik	312,96 detik	313,04 detik	321,6 detik	321,96 detik	321,32 detik	321,63 detik
Rise Time	313,56 detik	313,6 detik	313,28 detik	313,36 detik	321,92 detik	321,92 detik	321,6 detik	321,85 detik
Peak Time	314,56 detik	314,24 detik	314,24 detik	314,8 detik	321,96 detik	322,32 detik	321,96 detik	322,4 detik
Overshoot	10%	13,88%	21,66%	13,33%	11,6%	15%	19,16%	13,33%
Settling time	314,88 detik	316,16 detik	-	315,12 detik	324,84 detik	325,16 detik	-	324,68 detik

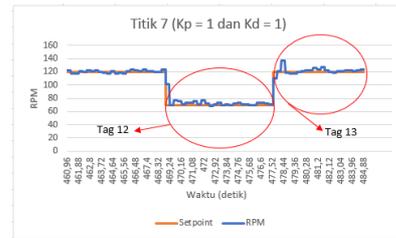
Berdasarkan keempat grafik yang ditunjukkan oleh gambar 25, gambar 26, gambar 27 dan gambar 28 serta Tabel VIII pergerakan kecepatan kendaraan saat mendeteksi tag 10 akan mengalami deselerasi menjadi disekitar 60 RPM sesuai dengan *setpoint* yang ada. Dalam performansi saat kendaraan mengalami deselerasi pengujian 4 memiliki *delay time*, *rise time* dan *settling time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 313,04 detik, *rise time* bernilai 313,36 detik, *overshoot* bernilai 13,33% dan *settling time* bernilai 315,12 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 10. Lalu saat kendaraan mendeteksi tag 11 maka kecepatan kendaraan akan mengalami akselerasi hingga mencapai disekitar 120 RPM sesuai *setpoint* pada tag 11. Dalam performansi saat kendaraan mengalami akselerasi pengujian 4 memiliki *rise time*, *settling time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu *rise time* bernilai 321,85 detik, *settling time* bernilai 324,68 dan *overshoot* bernilai 13,33%. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 11.

I. Pengujian Sistem di Titik 7

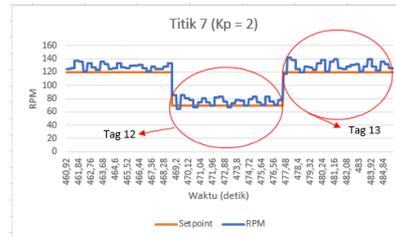
Pada pengujian di titik 7, kendaraan akan mendeteksi tag 12 yang memiliki *setpoint* kecepatan 70 RPM sehingga kendaraan yang tadinya memiliki kecepatan 120 RPM sesuai dengan tag 11 di titik 6 akan mengalami deselerasi hingga menyesuaikan dengan *setpoint* di tag 12 selanjutnya kendaraan akan mendeteksi tag 13 yang memiliki *setpoint* 120 RPM sehingga kendaraan akan mengalami akselerasi dan akan menyesuaikan kecepatan sesuai dengan *setpoint* tag 13. Hasil dari pengujian di titik 7 dapat dilihat pada gambar 29, gambar 30, gambar 31 dan gambar 32 serta Tabel IX.



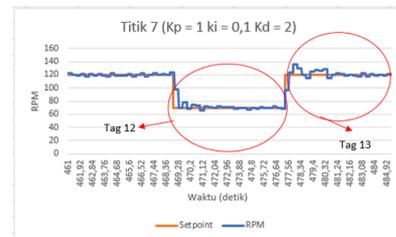
Gambar 29. Grafik Pengujian Pertama di Titik 7 dengan Kp = 1 dan Kd = 2



Gambar 30. Grafik Pengujian Kedua di Titik 7 dengan Kp = 1 dan Kd = 1



Gambar 31. Grafik Pengujian Ketiga di Titik 7 dengan Kp = 2



Gambar 32. Grafik Pengujian Ketiga di Titik 7 dengan Kp = 1, Ki = 0,1 dan Kd = 2

TABLE IX. PERFORMANSI SISTEM DI TITIK 7

Parameter	Tag 12				Tag 13			
	Pengujian 1 Kp = 1 dan Kd = 2	Pengujian 2 Kp = 1 dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0,1 dan Kd = 2	Pengujian 1 Kp = 1 Kd = 1	Pengujian 2 Kp = 1 dan Kd = 1	Pengujian 3 Kp = 2	Pengujian 4 Kp = 1, Ki = 0,1 dan Kd = 2
Delay Time	469,2 detik	469,2 detik	468,92 detik	469 detik	477,48 detik	477,6 detik	477,24 detik	477,32 detik
Rise Time	469,52 detik	469,24 detik	469,2 detik	469,32 detik	477,8 detik	477,92 detik	477,56 detik	477,6 detik
Peak Time	469,88 detik	469,56 detik	469,56 detik	469,64 detik	478,16 detik	478,24 detik	477,6 detik	477,92 detik
Overshoot	10,95%	18,1%	21,9%	12,38%	13%	14,72%	18,6%	13,33%
Settling time	470,84 detik	472,12 detik	-	471,24 detik	480,72 detik	481,76 detik	-	480,8 detik

Berdasarkan keempat grafik yang ditunjukkan oleh gambar 4.27, gambar 4.28, gambar 4.29 dan gambar 4.30 serta Tabel IX pergerakan kecepatan kendaraan saat mendeteksi tag 12 akan mengalami deselerasi menjadi disekitar 70 RPM sesuai dengan *setpoint* yang ada. Dalam performansi saat kendaraan mengalami deselerasi pengujian 4 memiliki *delay time*, *rise time*, *peak time* dan *settling time* yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 469 detik, *rise time* bernilai 469,2 detik, *peak time* bernilai 469,56, *overshoot* bernilai 12,38% dan *settling time* bernilai 471,24 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 12. Lalu saat kendaraan mendeteksi tag 13 maka kecepatan kendaraan akan mengalami akselerasi hingga mencapai disekitar 120 RPM sesuai *setpoint* pada tag 13. Dalam performansi saat kendaraan mengalami akselerasi pengujian 4 memiliki *delay time*, *rise time*, *peak time*,

settling time yang cukup cepat dan *overshoot* yang cukup kecil yaitu memiliki *delay time* bernilai 477,32 detik, *rise time* bernilai 477,6 detik, *peak time* bernilai 477,92, *overshoot* bernilai 13,33% dan *settling time* bernilai 480,8 detik. Hal ini menunjukkan pengujian 4 memiliki tingkat kestabilan kecepatan kendaraan terbaik pada tag 12.

KESIMPULAN

Sistem kendali PID pada *autonomous electric vehicle* untuk mengontrol kecepatannya dapat diimplementasikan dan berkerja dengan baik dengan menggunakan $K_p = 1$, $K_i = 0,1$ $K_d = 2$ sesuai dengan pengujian 4 dikarenakan memiliki *overshoot* yang tidak terlalu besar dan *settling time* yang cukup cepat serta memiliki nilai *rise time*, *delay time* dan *peak time* yang cukup cepat sehingga memiliki kestabilan terbaik dibanding sistem dengan pengujian yang lainnya. Kemudian apabila nilai K_p yang digunakan besar dan tidak adanya pengendali K_i dan K_d maka kecepatan rata-rata akan berada diatas *setpoint*. Kondisi ini mengakibatkan terjadinya *overshoot* yang besar, serta osilasi yang besar pada sistem yang mengakibatkan sistem menjadi tidak stabil.

Saat ini penelitian hanya menggunakan metode PID sebagai parameter kecepatan yang dipakai, untuk penelitian selanjutnya metode PID dapat digabungkan dengan beberapa metode lainnya seperti metode Fuzzy ataupun Neural Network (NN) sehingga pergerakan *autonomous electric vehicle* dapat menjadi lebih baik lagi dan pada penelitian ini hanya menggunakan RFID sebagai inputan kecepatannya, untuk penelitian selanjutnya dapat juga dipakai sensor lainnya seperti radar ataupun kamera sehingga kecepatan kendaraan dapat diatur berdasarkan jenis lingkungannya.

DAFTAR PUSTAKA

- [1] Y. G. Choi, K. Il Lim, and J. H. Kim, "Lane change and path planning of autonomous vehicles using GIS," *2015 12th Int. Conf. Ubiquitous Robot. Ambient Intell. URAI 2015*, pp. 163–166, 2015, doi: 10.1109/URAI.2015.7358855.
- [2] M. Taufiqurrahman, S. Sumardi, and M. A. Riyadi, "Perancangan Self Driving Dengan Metode Kontrol Pd Pada Sistem Tracking Autonomous Car," *Transient J. Ilm. Tek. Elektro*, vol. 5, no. 2, pp. 173–179, 2016.
- [3] J. H. Teng, K. Y. Hsiao, S. W. Luan, R. C. Leou, and S. Y. Chan, "RFID-based autonomous mobile car," *IEEE Int. Conf. Ind. Informatics*, pp. 417–422, 2010, doi: 10.1109/INDIN.2010.5549708.
- [4] J. Pérez, F. Seco, V. Milanés, A. Jiménez, J. C. Díaz, and T. de Pedro, "An RFID-based intelligent vehicle speed controller using active traffic signals," *Sensors*, vol. 10, no. 6, pp. 5872–5887, 2010, doi: 10.3390/s100605872.
- [5] D. Saputra, D. Cahyadi, and A. Harsa Kridalaksana, "Sistem Otomasi Perpustakaan Dengan Menggunakan Radio Frequency Identification (RFID)," *J. Inform. Mulawarman*, vol. 5, no. 3, pp. 1–11, 2010.
- [6] Y. U. D. Sipul, D. A. Prasetya, and W. Dirgantara, "Control System Kendali Kecepatan Sepeda Motor Listrik Dengan Metode PID," vol. 1, no. 1, pp. 45–49, 2019.
- [7] A. P. B. J. P. Sinaga, "Perancangan dan Implementasi Kontroler PID untuk Pengaturan Autonomous Car-Following Car," *Tek. Pomits*, vol. 3, no. 1, pp. 1–6, 2014.
- [8] R. S. R. Simbolon, "Analisis dan Simulasi Sistem Kontrol PI dan PID Menggunakan XCOS SCILAB," pp. 1–114, 2020.
- [9] H. D. Laksono, *Sistem Kendali Dengan PID dan Kompensator (Pendekatan Tempat Kedudukan Akar)*. 2019.