

Perancangan dan Implementasi Kontroler PID untuk Pengaturan *Autonomous Car-Following Car*

Andreas Parluhutan Bonor Sinaga dan Josaphat Pramudijanto

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

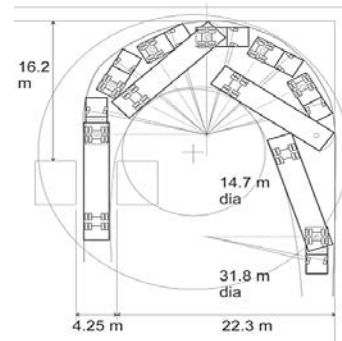
E-mail: bonor09@mhs.ee.its.ac.id , jos@ee.its.ac.id

Abstrak—Pengiriman logistik tentunya harus dilakukan secara merata ke setiap daerah, umumnya pengiriman logistik menggunakan kendaraan truk yang berbasis gandeng dalam pengirimannya. Beberapa permasalahan yang muncul, yaitu bagaimana mengontrol performansi truk gandeng dalam melakukan manuver di jalanan-jalanan yang relatif sempit dan berkelok-kelok. Untuk membantu pengemudi truk dalam melakukan manuver pada keadaan medan jalan yang seperti di atas, dilakukan suatu inovasi yaitu perancangan sistem *Autonomous Car-Following Car* di mana, pada aplikasinya gerakan kargo akan otomatis mengikuti *leadernya*. Penerapan kontroler PID digunakan untuk menjaga performansi *Autonomous Car* ketika melakukan manuver, pada implementasi didapatkan penurunan *time settling* menjadi 2,7 Detik dan peningkatan *error steady state* sebesar 2,44 %. Pada pengujian *following* terhadap *leader car*, kondisi *autonomous car* masih mampu mempertahankan range keadaan ideal pada setpoint 20 cm.

Kata Kunci— *Autonomous Car*, *Following Car*, Kontroler PID, *time settling*, *error steady state*

I. PENDAHULUAN

KONDISI medan jalan di Indonesia relative tidak merata bagusnya seperti di kota-kota besar pada umumnya, banyak daerah-daerah terpencil dan rawan bencana di pelosok Negara yang kondisi medan jalannya sangat buruk, belum lagi bila jalan-jalan yang dilalui sangat sempit dan berkelok-kelok, hal ini menyebabkan sulitnya menjangkau dan melakukan kegiatan logistik di daerah tersebut, belum lagi pada umumnya kendaraan-kendaraan logistik berupa truk gandeng yang mempunyai konstur yang sangat panjang dan sulit untuk melakukan manuver di kondisi jalan yang sempit bahkan berkelok-kelok, hal ini dapat diamati di Gambar 1. Terlihat bahwa untuk sebuah truk gandeng memerlukan sebuah ruang yang sangat besar untuk melakukan sebuah manuver, sehingga sangat tidak efektif untuk dikemudi di daerah-daerah yang mempunyai konstur medan jalan yang sempit dan berkelok-kelok. Perkembangan inovasi teknologi menjawab permasalahan tersebut dengan mengusung tema berkendara dengan prinsip *driverless*, di mana pengemudi truk gandeng tidak harus melakukan manuver yang sulit untuk melewati daerah-daerah tersebut, maka terbentuklah konsep *Autonomous Car-Following Car* yang dapat memberi kemudahan bagi pengemudi truk gandeng dalam melakukan kegiatan ekspedisi ke segala bentuk kondisi jalan. Sistem ini



Gambar 1. Sudut Belok terhadap Pergerakan Truk Gandeng

akan direalisasikan dengan bentuk *cargo-following truck*. Di mana sebuah kargo yang pada umumnya digandeng pada sebuah truk akan diletakkan terpisah dengan truk, sehingga kargo akan bergerak mengikuti pergerakan truk.

Penelitian ini dilakukan dengan mengambil sampel dua buah mobil RC (*Remote Control*) yang dimodifikasi, RC-1 bertindak sebagai *Autonomous Car* dan RC-2 bertindak sebagai *Leader Car* yang melakukan navigasi secara manual. Perancangan dari *Autonomous Car-Following Car*, adalah salah satu aplikasi di mana *Autonomous Car* dapat mengikuti navigasi dari *Leader Car*.

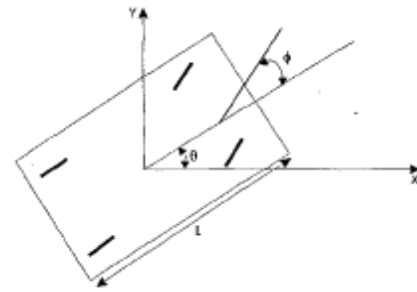
Sistem kontroler PID merupakan sebuah teknik kontrol klasik yang sangat banyak diimplementasikan dalam berbagai *plant*, termasuk *plant* yang memiliki sistem *nonlinear* yang memiliki berbagai jenis ketidakpastian. Kontroler PID ini ditujukan untuk mengontrol performansi dari *Autonomous Car*, serta untuk mempertahankan kestabilan ketika mengikuti pergerakan *Leader Car*.

II. DASAR TEORI

A. *Autonomous Car* [1]

Dikenal sebagai mobil mandiri atau dapat diinformasikan sebagai *driverless vehicle*, merupakan sebuah kendaraan beroda yang mampu bergerak secara mandiri karena dalam pergerakannya dilengkapi dengan penggerak (aktuator) yang dikendalikan oleh komputer yang terpasang di dalamnya. *Autonomous Car* sendiri banyak dipakai di dalam berbagai bidang, sehingga dalam penerapannya kendaraan ini dibagi menjadi 3 sifat utama, diantaranya:

- **Line Following**
Sebuah Autonomous Car dirancang dapat berjalan sendiri dengan mengikuti jalur-jalur yang sudah ditentukan, biasanya digunakan *Global Positioning Service* (GPS) dalam pengaplikasiannya.
- **Object Following**
Sebuah Autonomous Car dirancang dapat berjalan sendiri dengan mengikuti suatu objek yang bertindak sebagai *master* atau *leader*, umumnya biasanya digunakan sensor posisi atau jarak dalam pengaplikasiannya.
- **Manuver**
Sebuah Autonomous Car dirancang agar dapat melakukan manuver-manuver tertentu, seperti parkir paralel, *sliding*, dsb.



Gambar 2. Sudut Pusat dan Roda Depan Mobil [2]

B. Model Matematis Autonomous Car [2]

Model *Autonomous Car* yang digunakan pada Tugas Akhir ini adalah model Ackerman dengan penggerak utama terdapat pada roda belakang (*Rear Wheel Shaft Drive*) yang pergerakannya meliputi pergerakan maju dan mundur, sedangkan kemudi diletakkan pada posisi roda depan. Untuk permodelan dari mobil ini terdiri dari dinamika (permodelan fisik dari kendaraan) dan kinematika (permodelan dari pergerakan kendaraan).

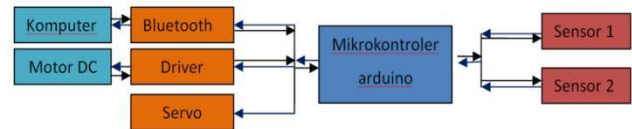
Permodelan dengan menggunakan metode ini memiliki empat buah titik acuan : (x,y) sebagai posisi dari *autonomous car*, ϕ sebagai sudut pusat posisi mobil, dan θ sebagai posisi sudut kemudi mobil. Untuk permodelan ini juga memiliki dua buah masukan, yaitu u_1 sebagai kecepatan putar roda belakang (kecepatan mengemudi) dan u_2 sebagai kecepatan *steering*. Dimensi fisik dari robot mobil ini ditunjukkan pada Gambar 2.

Persamaan dinamika mobil dapat ditulis.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{\tan \theta}{L} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (1)$$

Proses pengujian *following* dilakukan dengan kecepatan relatif lambat, dalam hal ini pengaruh *shockbreaker* dari mobil diabaikan. Sehingga persamaan posisi diperoleh dengan mengintegrasikan persamaan dinamika sebelumnya, seperti yang ditunjukkan pada Persamaan 2.

$$\begin{aligned} x &= \int_{t_0}^t \cos \theta \cdot u_1 dt \\ y &= \int_{t_0}^t \sin \theta \cdot u_1 dt \\ \theta &= \int_{t_0}^t \frac{\tan \phi}{L} \cdot u_1 dt \\ \phi &= \int_{t_0}^t u_2 dt \end{aligned} \quad (2)$$



Gambar 3. Sudut Pusat dan Roda Depan Mobil

C. Kontroler PID [3]

Kontroler proporsional ditambah integral ditambah differensial (PID) merupakan kontroler yang aksi kontrolnya merupakan kombinasi dari aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol differensial.

Untuk kontroler PID, sinyal kesalahan $e(t)$ merupakan masukan kontroler sedangkan keluaran kontroler adalah sinyal kontrol $u(t)$. fungsi alih kontroler PID dapat dilihat pada Persamaan 3.

$$\frac{U(s)}{E(s)} = K_p \left\{ 1 + \frac{1}{\tau_i s} + \tau_d \right\} \quad (3)$$

dimana K_p adalah penguatan proporsional, τ_i adalah waktu integral dan τ_d adalah waktu differensial.

III. PERANCANGAN SISTEM

A. Perancangan Sistem Autonomous Car

Sistem kerja *autonomous car* yang akan dirancang untuk bisa melakukan gerakan *autonomous* (otomatis) dapat dilihat pada Gambar 3.

Mengacu pada sistem yang dideskripsikan pada Gambar 3., mikrokontroler digunakan sebagai otak dari sistem yang akan dibuat, pemrograman akan dilakukan melalui komputer yang terintegrasi dengan modul perangkat lunak arduino, hasil dari pemrograman tersebut akan terintegrasi menuju perangkat kabel konektor yang terhubung dengan mikrokontroler arduino, hasil dari pemrograman tersebut akan langsung dihubungkan menuju pin-pin mikrokontroler yang terintegrasi menuju perangkat *driver* motor DC, servo, dan sensor. Sensor yang telah menerima masukan perintah dari arduino akan langsung terintegrasi dengan plant *autonomous car*, di mana plant tersebut akan bekerja sesuai dengan pengukuran jarak *autonomous car* dengan *leader*, hasil dari jarak pengukuran akan kembali dikirim ulang menuju mikrokontroler untuk dilakukan penganalisis data terhadap perilaku yang

ditimbulkan oleh motor DC dalam beberapa kondisi yang ditimbulkan (meliputi gerak lurus ataupun berbelok), hasil dari respons motor DC yang ditimbulkan akan dikirim kembali menuju mikrokontroler dan dianalisis di komputer untuk melakukan step pengontrolan PID-nya.

B. Perancangan Model Matematis

Kecepatan yang dihasilkan pada Persamaan 2., sehingga posisi mobil pada bidang x dan y serta sudut terhadap mobil dan sudut belok ban diperoleh dengan mengintegrasikan Persamaan 2. sebagai berikut :

$$\begin{aligned}
 x &= \int_{t_0}^t \cos \theta . u_1 dt \\
 y &= \int_{t_0}^t \sin \theta . u_1 dt \\
 \theta &= \int_{t_0}^t \frac{\tan \phi}{0.26} . u_1 dt \\
 \phi &= \int_{t_0}^t u_2 dt
 \end{aligned}
 \tag{3}$$

Karena hasil output dari *plant autonomous car* berupa posisi terhadap waktu, maka untuk mendapatkan persamaan posisi berlaku Persamaan 4 :

$$\text{Posisi} = \sqrt{x^2 + y^2}
 \tag{4}$$

C. Perancangan Mekanika Plant

Dengan asumsi bahwa jangkauan servo terhadap mobil RC hanya sebesar $\pm 45^\circ$ untuk belok kiri dan kanan, sehingga sudut jangkauan servo tidak digunakan seluruhnya (jangkauan servo sebesar 90° untuk belok kiri dan kanan). Untuk lebih jelasnya dapat dilihat pada Gambar 4.

D. Perancangan Pembebanan

Tahap pembebanan sangat penting digunakan untuk pengujian performansi dari motor, pembebanan harus dilakukan dengan gaya lawan yang relatif konstan. Pembebanan sendiri dibagi menjadi dua bagian yaitu kondisi berbeban nominal dan kondisi berbeban maksimal.

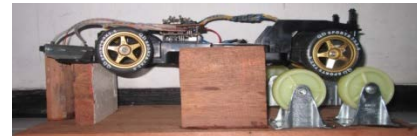
Seperti pada Gambar 5. pengujian pembebanan nominal dilakukan dengan hanya menggunakan roda sebagai asumsi pengganti gaya gesek di bidang datar, dan pada Gambar 6. pengujian maksimal dengan menggunakan roda ditambah lempengan kayu untuk menambah gaya gesek.

E. Model Matematika Plant [4]

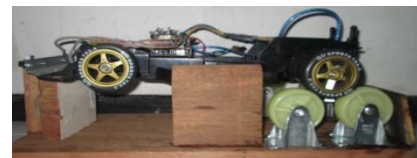
Sebelum melakukan pemodelan sistem, saya membandingkan antara 6 metode identifikasi. Setelah didapat fungsi penghantar dari masing-masing metode, langkah selanjutnya adalah menentukan mana metode yang terbaik melalui metode validasi *Integral Square Error (ISE)*. Dan akhirnya dapat disimpulkan bahwa identifikasi *viteckhova 2nd order* merupakan identifikasi yang paling baik.



Gambar 4. Jangkauan Servo terhadap Mobil RC



Gambar 5. Pengujian dengan Pembebanan Nominal



Gambar 6. Pengujian dengan Pembebanan Maksimal

Tabel 1. Model Plant dan nilai RMSE

Beban	Persamaan Model Plant	RMSE
Nom	$G_t(s) = \frac{1,009}{0,626s^2 + 1,582s + 1}$	72,88
Mak	$G_t(s) = \frac{1,004}{0,508s^2 + 1,426s + 1}$	102,9

Dipilih kecepatan antara 833-2059 rpm untuk beban nominal dan kecepatan 549-1864 rpm untuk beban maksimal karena pada kecepatan itu terletak pada kurva linier.

Didapat fungsi penghantar dari masing-masing kondisi beban seperti yang ditunjukkan pada Tabel 1.

F. Desain Kontroler [3]

Kontroler PID pada pengerjaan Tugas Akhir ini digunakan untuk mengendalikan kecepatan *autonomous car* dalam hal ini penulis menginginkan penurunan waktu tunak sebesar 5% dari keadaan awal, serta mengurangi *error steady stated* dengan batas toleransi diasumsikan sebesar 1%. Kontroler PID digunakan karena kesederhanaannya, mudah diimplementasikan, dan tentunya sangat familiar di teknologi industri.

Langkah pertama dalam merancang kontroler PID, adalah menemukan nilai dari parameter-parameternya. Tuning dilakukan secara analitik.

Parameter yang harus dicari adalah K_p, τ_i, τ_d pada kondisi beban nominal dan beban maksimal.

1. Parameter PID untuk beban nominal

- $K = 1,009$
- $\tau = 0$ (Tidak ada *delay*)
- $\omega_n = 1,263$
- $\zeta = 1$
- $\tau_d = \frac{1}{2\zeta\omega_n} = \frac{1}{2(1)(1,582)} = 0,3160$

$$\bullet \tau_i = \frac{2\zeta}{\omega_n} = \frac{2(1)}{1,5182} = 1,264$$

$$\bullet K_P = \frac{2\zeta}{\tau^* \omega_n K} \text{ bila } t_s(5\%) = 4,8 \text{ detik}$$

Maka

$$\tau^* = \frac{4,8}{3} = 1,6$$

$$\bullet K_P = \frac{2\zeta}{\tau^* \omega_n K} = \frac{2(1)}{(1,6)(1,263)(1,009)} = 0,9808$$

$$\bullet K_i = \frac{0,9808}{1,264} = 0,776$$

$$\bullet K_d = 0,9808 * 0,3160 = 0,3099$$

2. Parameter PID untuk beban maksimal

$$\bullet K = 1,004$$

$$\bullet \tau = 0 \text{ (Tidak ada delay)}$$

$$\bullet \omega_n = 1,403$$

$$\bullet \zeta = 1$$

$$\bullet \tau_d = \frac{1}{2\zeta\omega_n} = \frac{1}{2(1)(1,403)} = 0,3563$$

$$\bullet \tau_i = \frac{2\zeta}{\omega_n} = \frac{2(1)}{1,403} = 1,4255$$

$$\bullet K_P = \frac{2\zeta}{\tau^* \omega_n K} \text{ bila } t_s(5\%) = 4,5 \text{ detik}$$

Maka

$$\tau^* = \frac{3,7}{3} = 1,5$$

$$\bullet K_P = \frac{2\zeta}{\tau^* \omega_n K} = \frac{2(1)}{(1,5)(1,403)(1,004)} = 0,946$$

$$\bullet K_i = \frac{0,946}{1,4255} = 0,664$$

$$\bullet K_d = 0,946 * 0,3563 = 0,337$$

IV. PENGUJIAN DAN ANALISIS

Pada pengerjaan Tugas Akhir ini pengujian sistem dilakukan melalui dua cara yaitu simulasi dan implementasi. Tujuan dilakukan pengujian untuk mengetahui performansi dari sistem. Dalam pengujian secara simulasi terdapat beberapa aspek pengujian yang diberikan terhadap sistem, yaitu:

Beberapa hal yang penting diperhatikan oleh penulis:

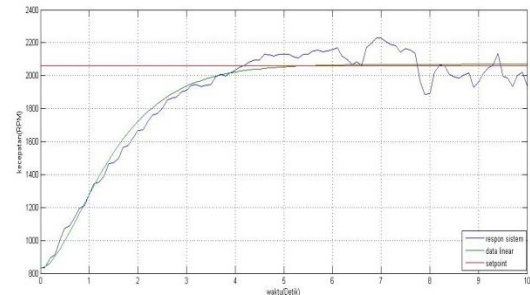
- 1) Pengujian simulasi performansi *autonomous car* dengan memberikan beban berupa gaya gesekan nominal terhadap *set point* berupa sinyal step.
- 2) Pengujian simulasi performansi *autonomous car* dengan memberikan beban berupa gaya gesekan maksimal terhadap *set point* berupa sinyal step.
- 3) Pengujian implementasi performansi *autonomous car* – *following car* pada bidang nominal dengan memberikan *set point* berupa jarak.

A. Perancangan Sistem *Autonomous Car*

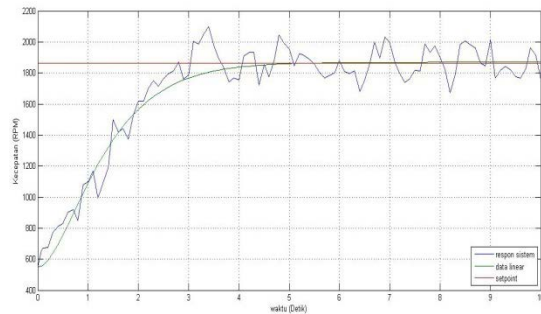
Dari respon pemodelan dalam semua kondisi pembebanan didapatkan spesifikasi respon yang ditunjukkan pada Tabel 2. dan Gambar 7 .

Tabel 2.
Spesifikasi Hasil Pemodelan

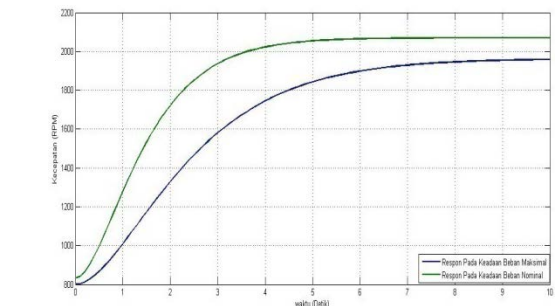
Beban	t_r (detik)	t_d (detik)	$t_s(5\%)$ (detik)	$Ess(\%)$
Nominal	4,540	1,068	4,8	0,486
Maksimal	4,463	1,050	4,6	0,29



(a) Beban Nominal



(b) Beban Maksimal



(c) Perbandingan Beban Nominal dengan Beban Maksimal

Gambar 7. Pengujian dengan Pembebanan Maksimal

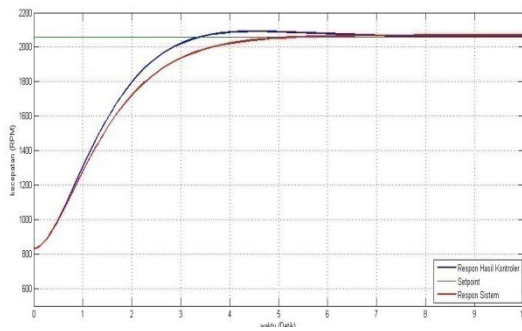
Tabel 3.
Perbandingan Hasil Kontroler PID Beban Nominal

Beban Nominal	t_r (detik)	t_d (detik)	$t_s(5\%)$ (detik)	$Ess(\%)$
Tanpa Kontroler	4,540	1,068	4,8	0,486
Dengan Kontroler	4,478	1,054	4,5	0,61

B. Simulasi Performansi *Autonomous Car* Beban Nominal dengan Menggunakan Kontroler PID

Pengujian ini bertujuan untuk mengetahui performansi kecepatan putar sistem *autonomous car* dengan menggunakan kontroler PID. Pada simulasi ini dilakukan pengujian terhadap performansi motor DC terhadap *set point* berupa sinyal step..

Pada Gambar 8. menunjukkan data hasil simulasi kontroler PID, dimana spesifikasi sistem menjadi berupa, untuk perbandingannya dapat dilihat pada Tabel 3.



Gambar 8. Perbandingan Hasil Kontroler PID Beban Nominal

Tabel 4. Perbandingan Hasil Kontroler PID Beban Maksimal

Beban Maksimal	t_r (detik)	t_d (detik)	t_s (5%) (detik)	Ess (%)
Tanpa Kontroler	4,463	1,050	4,6	0,29
Dengan Kontroler	4,063	0,956	4	0,03

Dari data yang diperoleh, dapat diambil kesimpulan bahwa kontroler PID yang terdapat pada sistem sangat bagus untuk memperbaiki nilai *settling time* serta sedikit *rise time* dan *time delay*, walaupun nilai *Error Steady State* yang diperoleh sedikit lebih besar, namun masih diambang batas toleransi.

C. Simulasi Performansi Autonomous Car Beban Maksimal dengan Menggunakan Kontroler PID

Pengujian ini bertujuan untuk mengetahui performansi kecepatan putar sistem *autonomous car* dengan menggunakan kontroler PID dalam kondisi beban maksimal. Pada simulasi ini dilakukan pengujian terhadap performansi motor DC terhadap *set point* berupa sinyal step.

Pada Gambar 9. menunjukkan data hasil simulasi kontroler PID, dimana spesifikasi sistem menjadi berupa, untuk perbandingannya dapat dilihat pada Tabel 4.

Dari data yang diperoleh, dapat diambil kesimpulan bahwa kontroler PID yang dapat memperbaiki nilai *Error Steady State* sehingga direduksi menjadi 0,03%.

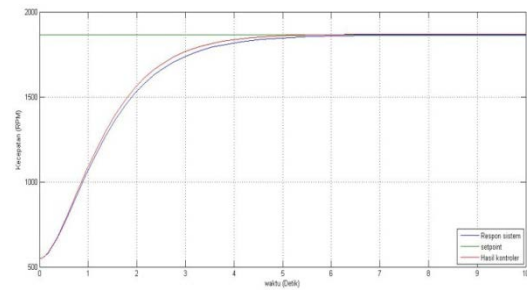
D. Implementasi Performansi Autonomous Car Beban Nominal dengan Menggunakan Kontroler PID

Pengujian ini bertujuan untuk mengetahui performansi kecepatan putar sistem *autonomous car* dengan menggunakan kontroler PID secara nyata. Pada pengujian ini dilakukan pengujian terhadap performansi motor DC terhadap *set point* berupa sinyal step.

Hasil diperlihatkan pada Tabel 5. dan hasil pengujian implementasi diperlihatkan pada Gambar 10.

E. Implementasi Performansi Autonomous Car Beban Maksimal dengan Menggunakan Kontroler PID

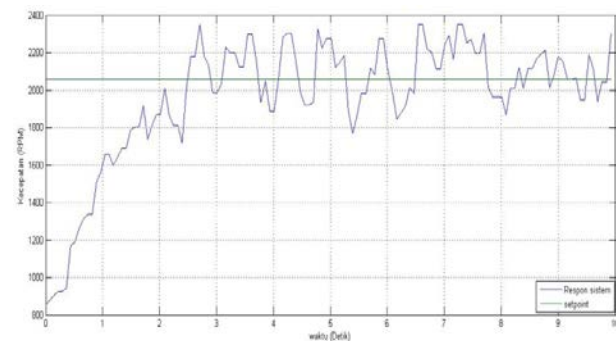
Pengujian ini bertujuan untuk mengetahui performansi kecepatan putar sistem *autonomous car* dengan menggunakan kontroler PID secara nyata. Pada pengujian ini dilakukan pengujian terhadap performansi motor DC terhadap *set point* berupa sinyal step.



Gambar 9. Perbandingan Hasil Kontroler PID Beban Maksimal

Tabel 5. Hasil implementasi PID Beban Nominal

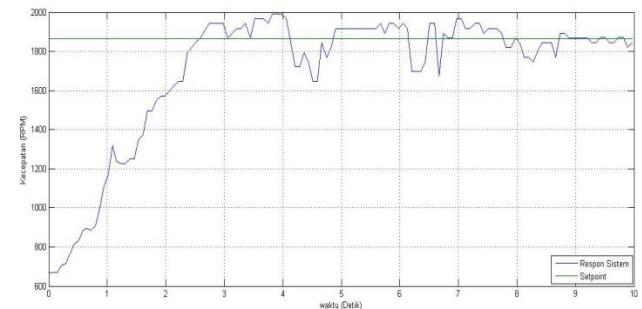
Beban Nominal	t_r (detik)	t_d (detik)	t_s (5%) (detik)	Ess (%)
	2,447	0,564	2,7	2,44



Gambar 10. Respon Implementasi PID pada Beban Nominal

Tabel 6. Hasil implementasi PID Beban Maksimal

Beban Maksimal	t_r (detik)	t_d (detik)	t_s (5%) (detik)	Ess (%)
	3,162	0,73	2,612	0,398



Gambar 11. Respon Implementasi PID pada Beban Maksimal

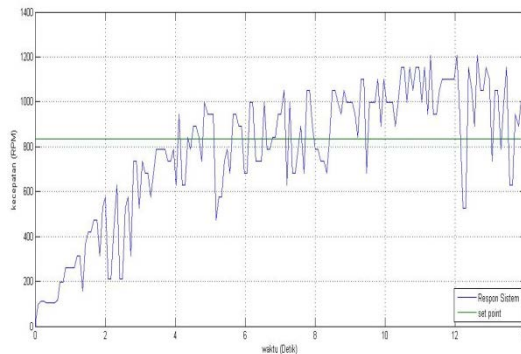
Hasil diperlihatkan pada Tabel 6. dan hasil pengujian implementasi diperlihatkan pada Gambar 11.

F. Implementasi Performansi Autonomous Car Menggunakan Kontroler PID Dengan Setpoint Posisi

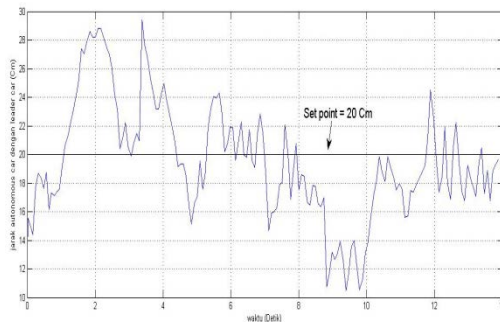
Implementasi kontroler sendiri menggunakan *software* pemrograman Arduino 1.0.5 yang dapat dilihat pada Gambar 10. , di mana blok kontroler PID yang digunakan diasumsikan sama dalam keadaan pembebanan nominal.



Gambar 10. Konsep Implementasi *Autonomous Car-Following Car*



Gambar 11. Respon Kecepatan Secara Acak



Gambar 12. Respon Implementasi dengan Kontroler PID

Pengujian ini akan memperlihatkan, bagaimana kondisi performansi *autonomous car* ketika melakukan *following*, ketika diberikan *set point* untuk jarak sebesar 20 cm dengan kondisi pergerakan secara lurus dan diberikan kecepatan untuk *leader* secara acak, untuk grafik kecepatan dapat dilihat pada Gambar 11.

Dari data implementasi juga dapat dilihat, bagaimana performansi *Autonomous Car* dalam mempertahankan jarak idealnya ketika melakukan *following*, seperti yang diperlihatkan pada Gambar 12. dari hasil yang diperoleh dapat diambil kesimpulan bahwa kondisi *autonomous car* masih mampu mengikuti pergerakan *leader* dengan *set point* yang ditentukan, walaupun masih sering terjadi penyimpangan posisi di daerah *set point*.

V. KESIMPULAN/SARAN

A. Kesimpulan

Berdasarkan hasil pengujian dan analisis, dapat diperoleh kesimpulan sebagai berikut:

1. Pada simulasi kontroler PID untuk pengaturan performansi *autonomous car-following car* didapatkan respon dengan spesifikasi yang diinginkan

yaitu penurunan *time settling* sebesar 0,3 Detik untuk beban nominal, namun terjadi peningkatan *error steady state* sebesar 0,124% dan pada beban maksimal terjadi penurunan *time settling* sebesar 0,6) dan pengurangan *error steady state* sebesar 0,26%

2. Pada implementasi kontroler PID untuk pengaturan performansi *autonomous car*, untuk beban nominal terjadi penurunan waktu dalam mencapai keadaan tunak menjadi 2,7 Detik dan terjadi peningkatan *error steady state* sebesar 2,44% untuk beban maksimal terjadi penurunan waktu dalam mencapai keadaan tunak menjadi 2,6 Detik dan menghasilkan *error steady state* sebesar 0,398%
3. kontroler PID hanya dapat dilakukan dalam kecepatan lambat dan didapatkan respon *autonomous car* masih dapat mampu mempertahankan jarak idealnya sebesar 20 cm, walau masih banyak terjadi osilasi di sekitar daerah *set point*-nya

B. Saran

Dari hasil penelitian yang dilakukan, untuk pengembangan berikutnya, disarankan beberapa hal berikut ini:

1. Diperlukannya bentuk pemodelan dan pemahaman tentang *autonomous car* yang akan membantu perancangan sistem yang lebih baik dan mendapatkan hasil implementasi yang lebih akurat.
2. Perancangan mekanik yang lebih akurat tanpa menggunakan metode modifikasi alat akan menghasilkan data yang lebih akurat.
3. Diperlukannya penguasaan metode programming arduino dan elektronika terutama untuk membuat *driver*.

DAFTAR PUSTAKA

- [1] Tom Vanderbilt, "Autonomous Cars Through the Ages", *AUTOPIA*, 2012.
- [2] A. Benalia, M. Djemal and J-P. Barbot, "Control of The Kinematic Car Using Trajectory Generation and The High Order Sliding Mode Control" *IEEE International Conference*, 2003.
- [3] Nurlita Gamayanti, *Diktat Kuliah Dasar Sistem Pengaturan*, Teknik Elektro Institut Teknologi Sepuluh November, Surabaya.
- [4] Jakoubek Pavel, "Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods", *Konference Studentske tvurci cinnosti*, 2009.