

IMPLEMENTASI KONTROLER PID PADA SISTEM KONTROL KECEPATAN PUTAR MOTOR DC UNTUK MOBIL ANAK BERBASIS ANDROID

PID CONTROLLER IMPLEMENTATION IN DC MOTOR ANGULAR SPEED CONTROL SYSTEM FOR CHILD CAR BASED ON ANDROID

Havan Arsyah Rahardjo¹, Ir. Porman Pangaribuan, M.T.², & Agung Surya Wibowo, S.T.,M.T.³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik, Elektro Universitas Telkom

havanarsya@students.telkomuniversity.ac.id¹, porman@telkomuniversity.ac.id², agungsw@telkomuniversity.ac.id³

Abstrak

Mobil mainan anak otomatis akan berbahaya jika hanya dikemudikan oleh anak. Diperlukan peran orang tua untuk mengawasi anak yang sedang bermain mobil mainan anak otomatis. Berdasarkan masalah tersebut, akan dirancang dan diimplementasikan suatu sistem kontrol kecepatan putar pada mobil mainan anak yang dapat dioperasikan melalui aplikasi *smartphone* berbasis Android, sehingga orang tua dapat ikut mengontrol kecepatan putar mobil tersebut.

Pada tugas akhir ini pengontrolan *proportional integrative derivative* (PID) akan digunakan sebagai pengontrol kecepatan putar pada *child car wheel* yang kecepatan putarnya akan tetap konstan. Karena dengan metode kontrol PID, tingkat kestabilan dan waktu untuk mencapai kestabilan akan menjadi lebih baik. Pada sistem mobil anak ini, digunakan dua buah sensor untuk membaca kecepatan putar yaitu *line tracking sensor*, untuk mendapatkan pembacaan kecepatan putar (RPM). Setelah melakukan pembacaan, maka sistem akan mengatur kecepatan putar motor DC agar kecepatan putarnya tetap stabil. Terdapat dua fitur utama pada aplikasi Android yang akan digunakan pada perancangan dan implementasi tugas akhir ini yaitu dapat maju dan mundur, sementara untuk berbelok masih menggunakan lingkaran kemudi pada mobil anak tersebut.

Hasil yang telah didapatkan dari pengujian adalah nilai KP, KI dan KD yang paling baik yaitu $KP = 4.75$, $KP2 = 3.75$, $KI = 0.2$, $KI2 = 0.15$, $KD = 0.8$ dan $KD2 = 0.75$ dengan perbedaan beban 1 kg tidak terlalu berpengaruh pada sistem dan *overshoot*. Pada saat menggunakan beban tambahan dari 8 kg-15 kg untuk 25 RPM dan 50 RPM, *settling time* cukup terlihat perbedaan dan pengaruhnya dengan nilai maksimal yaitu ± 19 detik dan 31.5 detik, walaupun pada akhirnya nilai *setpoint* yang diinginkan dapat tercapai yaitu 25 RPM dan 50 RPM.

Kata Kunci: Proportional Integrative Derivative (PID), *Child Car Wheel*, RPM, Android.

Abstract

Automatic car toys will be dangerous if only driven by child. Parents required to supervise children while playing automatic car toys. Based on these issues, will be designed and implemented a angular speed control system on the car toy which can be operated via smartphone applications based on Android. So that parents can participate to control the angular speed of the car.

Control Proportional Integrative Derivative (PID) in this final project will be used as a controller of angular speed on the child car wheel the angular speed will remain constant. Due to the PID control method, the level of stability and time to achieve stability will be better. On this child car systems, will be used two angular speed measuring sensor that is a line tracking sensor, to get a reading of angular speed (RPM). After reading, then the system will regulate angular speed of motor DC in order the angular speed remained stable. There are two main features of the Android app that will be used in the design and implementation of this final project that it can go forward and backward, while to turn still using the steering wheel on the the child car.

The results that have been obtained from the experiments is the value of KP, KI and KD is best $KP = 4.75$, $KP2 = 3.75$, $KI = 0.2$, $ki2 = 0.15$, $KD = 0.8$ and $KD2 = 0.75$ with the difference in the load of 1 kg was not affecting the system and overshoot. When using the extra load of 8 kg-15 kg for 25 RPM and 50 RPM, settling time is quite enough difference and the maximum influence is ± 19 second and 31.5 second, although in the end the desired setpoint values can be achieved at 25 RPM and 50 RPM.

Keywords : Proportional Integrative Derivative (PID), *Child Car Wheel*, RPM, Android.

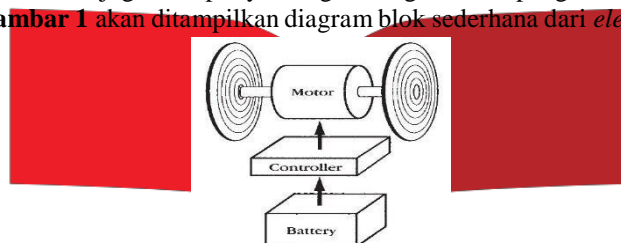
1. Pendahuluan

Kehadiran seorang anak dapat memberikan suasana yang berbeda dalam sebuah keluarga, tetapi mengurus seorang anak bukan persoalan yang mudah. Banyak anak yang kedua orang tuanya mencari nafkah yang

berakibat tidak memiliki banyak tenaga dan waktu luang untuk menemani sang anak bermain. Pada zaman yang modern saat ini, seharusnya orang tua meluangkan waktu dan tenaga untuk mengajak anak bermain supaya dapat mencegah anak dari ketergantungan gawai. Mobil anak yang bergerak otomatis tanpa harus didorong memang sudah terdapat di beberapa toko khusus anak kecil, tetapi masih jarang yang dapat bergerak secara konstan secara otomatis. Mobil mainan anak otomatis akan berbahaya jika hanya dikemudikan oleh anak. Diperlukan peran orang tua untuk mengawasi anak yang sedang bermain mobil mainan anak otomatis. Berdasarkan masalah tersebut, akan dirancang dan diimplementasikan suatu sistem kontrol kecepatan putar pada mobil mainan anak yang dapat dioperasikan melalui aplikasi *smartphone* berbasis Android, sehingga orang tua dapat ikut mengontrol kecepatan putar mobil anak tersebut. Untuk menentukan kecepatan putarnya, orang tua tidak perlu khawatir terhadap bahaya terlalu cepat, karena kecepatan putarnya sudah diatur saat pembuatan alat.

2. Prinsip Dasar Electric Vehicle

Electric vehicle atau sering disebut kendaraan listrik merupakan suatu alat transportasi yang mempunyai sumber energi dari baterai. Kendaraan listrik ini menggunakan sistem penggerak satu atau lebih motor listrik untuk menggerakkan roda. Lalu terdapat sebuah kontroler yang berfungsi sebagai penghubung antara motor listrik dan baterai, disamping itu kontroler juga mempunyai fungsi sebagai sistem pengatur aliran energi yang dibutuhkan oleh motor listrik. Pada **Gambar 1** akan ditampilkan diagram blok sederhana dari *electric vehicle*.



Gambar 1 Diagram Blok Sederhana Electric Vehicle

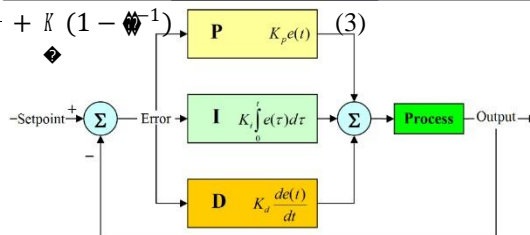
A. Metode Pengontrolan PID

Pada tugas akhir ini pengontrolan *Proportional Integrative Derivative* (PID) akan digunakan sebagai pengontrol kecepatan putar pada *child car wheel* yang kecepatan putarnya akan tetap konstan dan dapat diakses melalui Android. Karena dengan metode kontrol PID, tingkat kestabilan dan waktu untuk mencapai kestabilan akan menjadi lebih baik. Untuk persamaan PID waktu kontinu dan PID waktu diskrit akan dituliskan pada persamaan (1) dan (3). Sebelum mendapatkan PID waktu diskrit, PID waktu kontinu harus diubah terlebih dahulu menjadi PID transformasi Laplace yaitu pada persamaan (2).

$$U(s) = K_p E(s) + K_i \int_0^s E(s) ds + K_d \frac{dE(s)}{ds} \quad (1)$$

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2)$$

$$\frac{U(z)}{E(z)} = K_p + \frac{K_i}{(1-z^{-1})} + K_d (1-z^{-1}) \quad (3)$$



Gambar 2 Diagram Blok Kontroler PID

Kontrol proporsional memiliki keuntungan bahwa variabel kontrol yang kecil ketika kesalahan kontrol bernilai kecil dan untuk menghindari pengendalian yang berlebihan. Kelemahan utama menggunakan kontrol proporsional murni adalah menghasilkan *steady-state error*.

Fungsi utama dari kontrol integral adalah untuk memastikan bahwa *output* dari proses setuju dengan *setpoint* dalam kondisi mapan (*steady-state*). Kontrol proporsional biasanya terdapat kesalahan kontrol dalam kondisi mapan. Dengan kontrol integral, kesalahan kecil yang bernilai positif akan meningkatkan sinyal kontrol dan kesalahan kecil yang bernilai negatif akan menurunkan sinyal kontrol seberapa kecilpun kesalahannya.

Tujuan dari kontrol derivatif adalah untuk meningkatkan kestabilan sistem *loop* tertutup dengan syarat nilai konstanta derivatifnya kecil. Ketika proses dinamika, proses tersebut akan mengambil beberapa waktu sebelum perubahan variabel kontrol yang terlihat dalam proses *output*. Dengan demikian, sistem kontrol akan mengalami keterlambatan dalam mengoreksi kesalahan. Aksi kontroler dengan kontrol proporsional dan kontrol derivatif mungkin dapat diartikan kontrol dibuat sebanding dengan prediksi dari proses *output*.

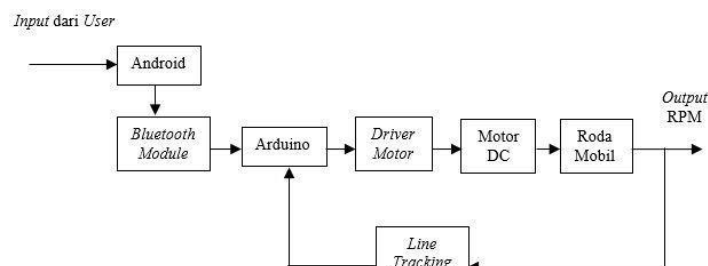
Untuk lebih jelas pengaruh dari kontrol proporsional, kontrol integral dan kontrol derivatif dapat dilihat pada **Tabel 1**.

Konstanta	Rise time (<i>tr</i>)	Overshoot	Settling time (<i>ts</i>)	Error Steady-state	Kestabilan
KP	<i>tr</i> berkurang	bertambah	-	berkurang	berkurang
KI	<i>tr</i> berkurang	bertambah	<i>ts</i> bertambah	nol	berkurang
KD	-	berkurang	<i>ts</i> berkurang	-	KD <<

Tabel 1 Table of Effect KP, KI and KD

B. Perancangan Sistem Secara Umum

Sistem yang akan direalisasikan dari perancangan sistem pada tugas akhir ini adalah sistem yang dapat mempertahankan kecepatan putar dengan stabil dengan beban yang berbeda-beda. Sistem dapat direalisasikan dengan cara membaca kecepatan putar dari roda mobil anak dan membandingkannya dengan *set-point* yang sudah ditetapkan. Setelah melakukan pembacaan dan perbandingan, maka akan dihasilkan *error* yang kemudian diproses menjadi data supaya kecepatan putar pada roda mobil anak tersebut dapat stabil seperti yang diharapkan.



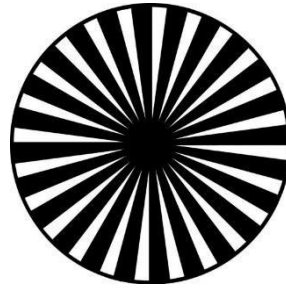
Gambar 4 Diagram Blok Kontrol Kecepatan Putar Pada Mobil Anak

Dari gambar diagram blok sistem secara umum pada **Gambar 4**, terlihat bahwa Android yang merupakan salah satu *operating system* (OS) *smartphone*, mempunyai peranan penting dalam sistem tugas akhir ini. Android akan dikendalikan oleh orang tua dari anak yang sedang menduduki mobil anak kecil, mobil anak tersebut akan bergerak jika sinyal bergerak dari Android sudah diterima oleh *Bluetooth Module HC-06*. Jika sinyal untuk bergerak sudah diterima oleh *Bluetooth Module HC-06*, maka Arduino akan memproses sinyal bergerak tersebut untuk menggerakkan roda mobil anak yang sudah dilengkapi dengan motor DC.

Setelah mobil anak yang sudah dilengkapi dengan motor DC pada rodanya bergerak, *line tracking sensor* akan membaca kecepatan putar pada roda mobil anak tersebut. Cara *line tracking sensor* membaca kecepatan putar dari roda mobil anak adalah dengan membaca perubahan warna hitam dan putih seperti pada **Gambar 5** dengan bentuk lingkaran yang sudah dipasang pada roda mobil anak. Hasil data yang sudah diproses dari pembacaan *line tracking sensor* akan mendapatkan kecepatan putar dalam satuan RPM. Terdapat 48 perubahan warna seperti pada **Gambar 5**. Dengan 48 perubahan warna tersebut maka kecepatan RPM dapat dihitung dengan persamaan seperti berikut :

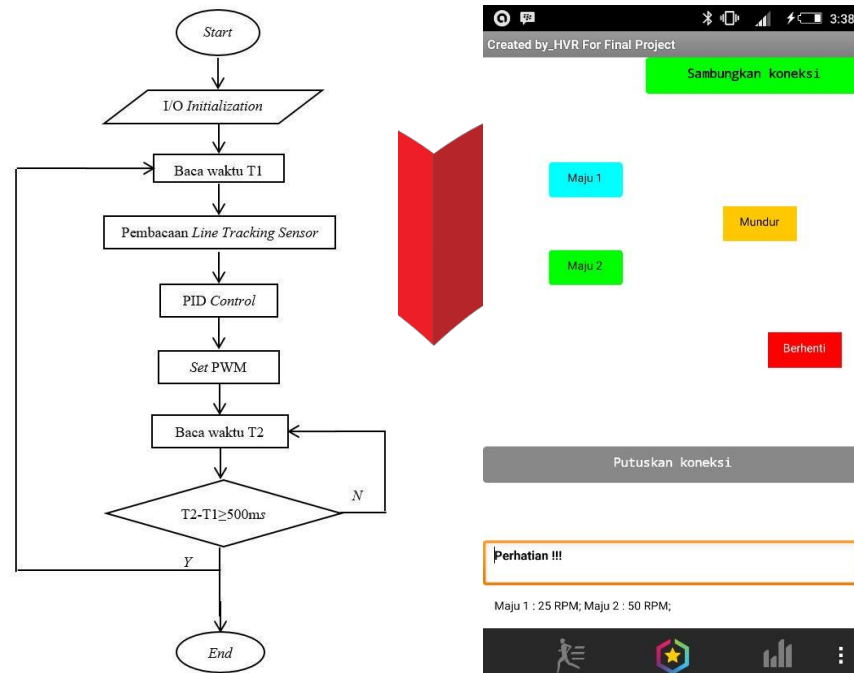
$$\text{RPM} = \left(\frac{D_s}{48}\right) 60 \quad (4)$$

Keterangan :
 RPM : *Rotation per minute*
 D_s : Data hasil pembacaan sensor
 48 : Perubahan (*chance*) yang terjadi dalam satu putaran
 60s : Dalam satu menit terdapat 60 detik



Gambar 5 Contoh 48 Segment Circle Black and White

C. Perancangan Sistem Perangkat Lunak



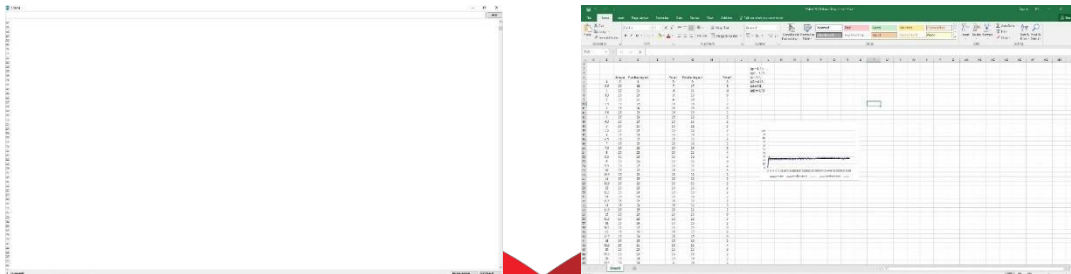
Gambar 6 Flowchart Perancangan Perangkat Lunak Dan User Interface Aplikasi Android

Pada perancangan perangkat lunak ini akan dijelaskan tahapan kerja dari sistem perangkat lunak yang dibuat dalam bahasa pemrograman C Arduino yang dibuat menggunakan *software* Arduino IDE dengan *flowchart* seperti pada **Gambar 3.4**. Pertama sistem harus terkoneksi dengan *smartphone* berbasis Android. Jika sudah terkoneksi, maka Android akan mengirimkan sinyal untuk bergerak yang dikendalikan oleh orang tua supaya aman. Sistem akan bergerak dengan RPM yang sudah di *set*. Sistem akan membaca waktu T1 yang akan digunakan sebagai perhitungan waktu *sampling* saat pertama kali bergerak. Setelah bergerak maka *line tracking sensor* akan membaca kecepatan putar dari roda mobil anak. Pembacaan *line tracking sensor* tersebut akan dibandingkan dengan RPM yang sudah di *set*. Dari perbandingan tersebut akan didapatkan *error* yang kemudian dikontrol dengan metode pengontrolan PID agar *error* tersebut dapat dikurangi. Langkah selanjutnya adalah *set* PWM untuk mengatur kecepatan putar motor DC. Lalu sistem akan membaca waktu T2 yang akan digunakan dalam perhitungan waktu *sampling* dengan rumus $T2-T1 \geq 500ms$. Waktu *sampling* (T_s) yang akan digunakan adalah 500ms atau $\frac{1}{2}$ detik.

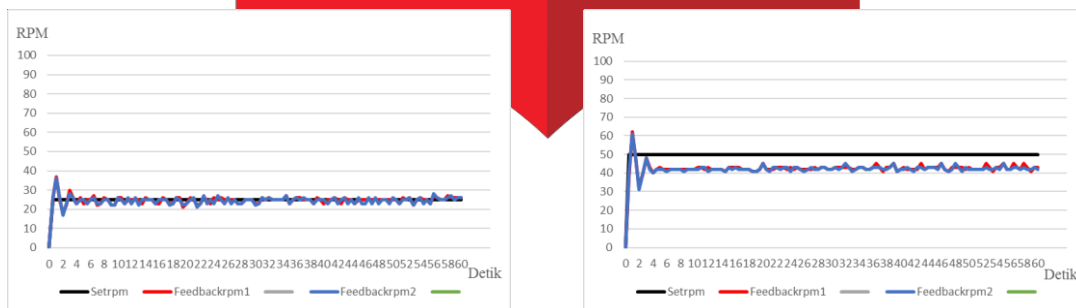
3. Pengujian dan Analisa

Setelah menyiapkan alat dan bahan seperti perangkat tugas akhir, laptop, *smartphone* Android dan beras sebagai beban, lalu akan dilakukan pengujian. Pertama kali sebelum pengujian adalah menyalakan saklar-saklar untuk sumber tegangan dari motor, kipas dan Arduino. Pengujian yang pertama ini alat akan diuji menggunakan kontrol P, letakkan laptop diatas bangku dan sambungkan kabel *peripheral* untuk membaca data dari Arduino, untuk nilai KI dan KD dibuat sama dengan 0, nilai KP, KI dan KD didapatkan dari hasil *trial and error*. Upload program yang sudah dibuat pada *software* Arduino lalu sambungkan koneksi antara Android dan Bluetooth yang sudah terhubung dengan Arduino dengan cara klik sambungkan koneksi dan pilih modul Bluetooth HC-O6 yang sudah di *pair*. *Pair* adalah menyambungkan dan verifikasi perangkat yang pertama kali dihubungkan pada pengaturan Bluetooth yang ada di Android. Setelah semua sudah siap buka *serial monitor* yang ada pada *software* Arduino lalu klik maju 1 untuk menguji dengan kecepatan putar 25 RPM, lalu hasil akan muncul di

serial monitor seperti pada **Gambar 7**, setelah jarak dan waktu yang ditempuh cukup lalu klik berhenti pada Android lalu salin dan simpan data yang sudah didapatkan di *serial monitor* kedalam *software notepad* dan tutup *serial monitor*, grafik dengan kontrol P 25 RPM akan ditampilkan pada **Gambar 8**. Setelah pengujian dengan 25 RPM lalu dilakukan pengujian dengan 50 RPM. Cara pengujian dengan 50 RPM hampir sama dengan 25 RPM, hanya saja tombol yang di klik pada Android yaitu maju 2, grafik dengan kontrol P 50 RPM akan ditampilkan pada **Gambar 8**. Setelah data-data yang diperlukan didapatkan lalu salin data dari *notepad* kedalam *Microsoft Exel* seperti pada **Gambar 7**, lalu tandai bagian yang ingin dibuat kedalam grafik dan klik *insert* pada menu yang berada pada bagian atas lalu pilih menu grafik dan pilih bentuk grafik sesuai dengan yang diinginkan.

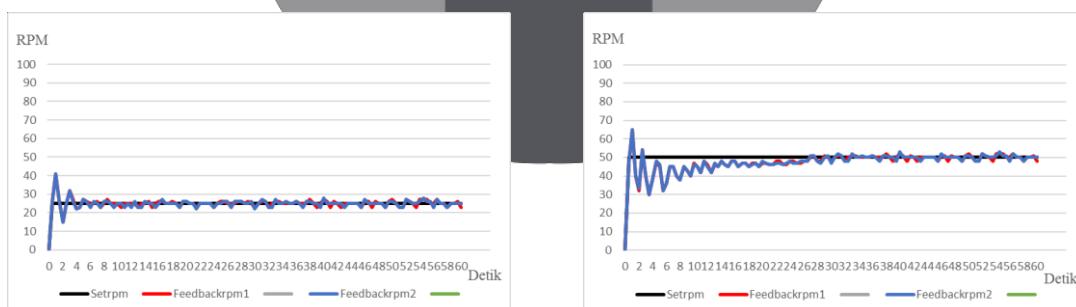


Gambar 7 Contoh Gambar Dari Serial Monitor Dan Microsoft Exel



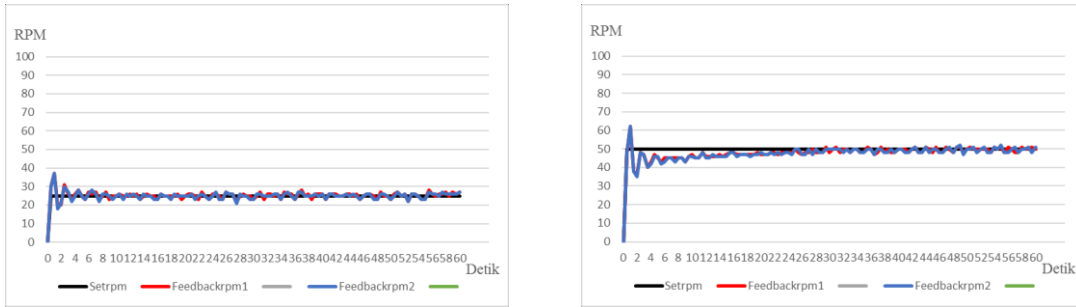
Gambar 8 Grafik Dengan Kontrol P 25 RPM Dan 50 RPM

Setelah pengujian dengan kontrol P, lalu dilakukan pengujian dengan kontrol PI, prosedurnya hampir sama dengan kontrol P, tetapi hanya nilai KD yang dibuat sama dengan 0. Masukkan nilai KI lalu *upload* program Arduino dan buka *serial monitor* lalu klik maju 1 untuk mendapatkan hasil dari 25 RPM. Setelah jarak dan waktu tertentu klik berhenti pada Android lalu salin dan simpan data dari *serial monitor* kedalam *notepad* lalu tutup *serial monitor* dan salin data kedalam *notepad* lalu tutup *serial monitor*. Setelah pengujian 25 RPM lalu pengujian dengan 50 RPM. Prosedur dan cara pengujian dengan 50 RPM hampir sama tetapi tombol yang di klik pada *software* Android adalah maju 2. Grafik dari 25 RPM dan 50 RPM dengan kontrol PI akan ditampilkan pada **Gambar 9**.



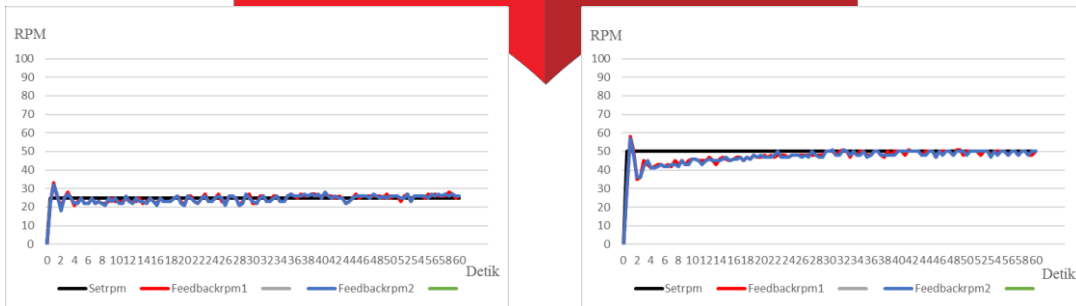
Gambar 9 Grafik Dengan Kontrol PI 25 RPM Dan 50 RPM

Setelah pengujian dengan kontrol P dan PI, lalu dilakukan pengujian dengan kontrol PID, prosedurnya hampir sama dengan kontrol P dan PI lalu nilai KD dimasukkan dan program di *upload* kedalam Arduino. Setelah program di *upload* lalu klik maju 1 untuk 25 RPM dan maju 2 untuk 50 RPM. Grafik dari 25 RPM dan 50 RPM dengan kontrol PID akan ditampilkan pada **Gambar 10**.

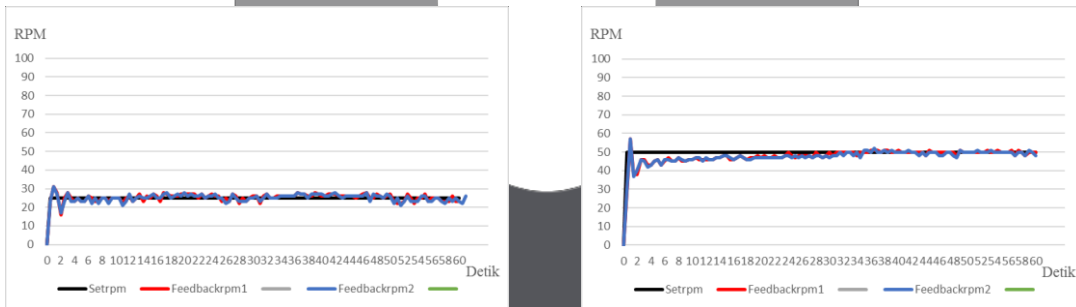


Gambar 10 Grafik Dengan Kontrol PID 25 RPM Dan 50 RPM

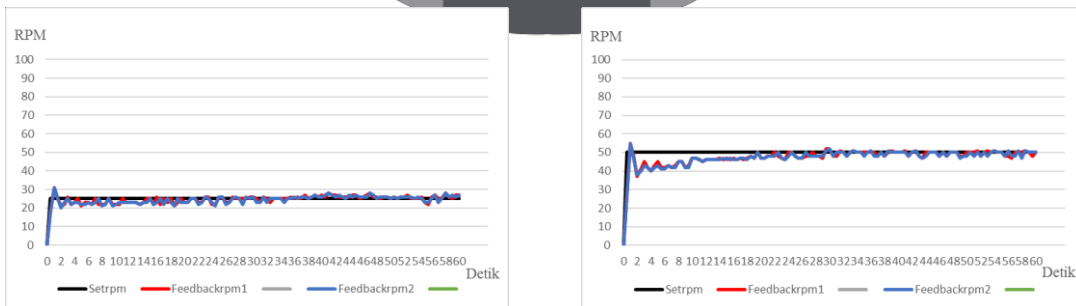
Pada pengujian yang selanjutnya ini akan dilakukan pengujian dengan menggunakan beban tambahan. Letakkan beban diatas bangku perangkat mobil tugas akhir dan letakkan laptop diatas beban mobil tersebut. Beban pertama yang akan diletakkan adalah 8 kg lalu klik maju 1 untuk mendapatkan hasil dari 25 RPM. Setelah jarak dan waktu tertentu klik berhenti pada Android lalu salin dan simpan data dari *serial monitor* kedalam *notepad* lalu tutup *serial monitor*. Setelah pengujian 25 RPM lalu pengujian dengan 50 RPM. Prosedur dan cara pengujian dengan 50 RPM hampir sama tetapi tombol yang di klik pada *software* Android adalah maju 2. Grafik dari 25 RPM dan 50 RPM dengan beban 8 kg akan ditampilkan pada Gambar 11. Pada pengujian-pengujian yang selanjutnya, beban akan ditambahkan 1 kg dari beban sebelumnya sampai dengan pengujian 15 kg dengan prosedur yang hampir sama seperti sebelumnya.



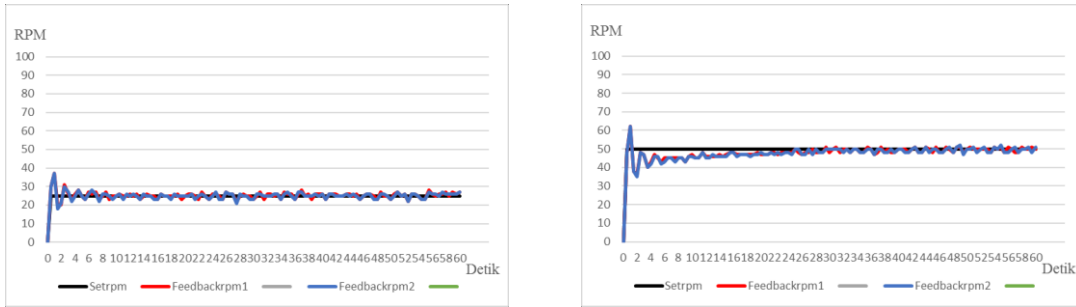
Gambar 11 Grafik Dengan Beban 8 Kg 25 RPM Dan 50 RPM



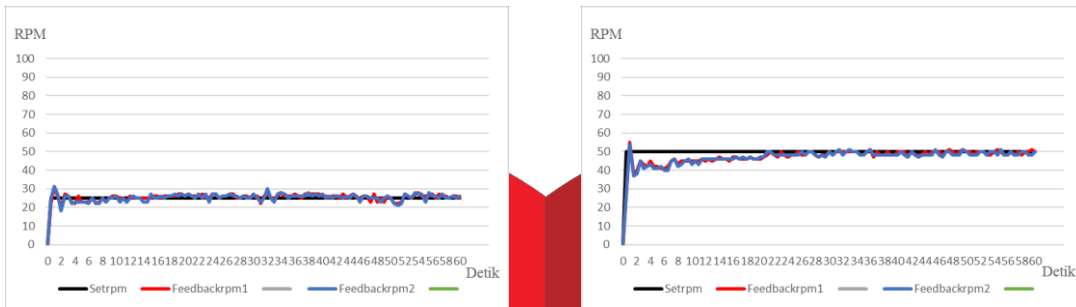
Gambar 12 Grafik Dengan Beban 9 Kg 25 RPM Dan 50 RPM



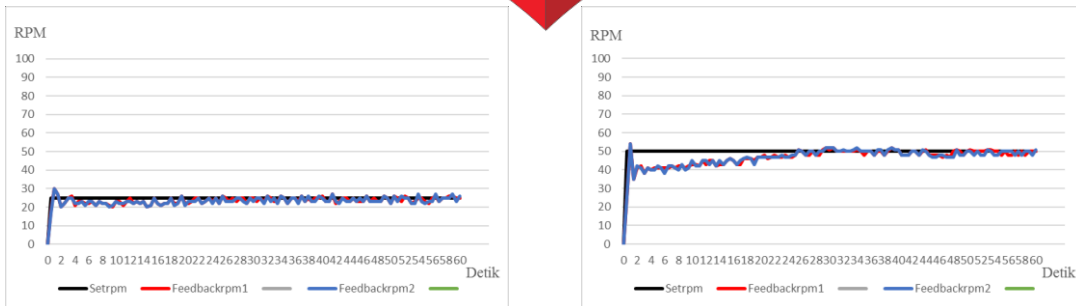
Gambar 13 Grafik Dengan Beban 10 Kg 25 RPM Dan 50 RPM



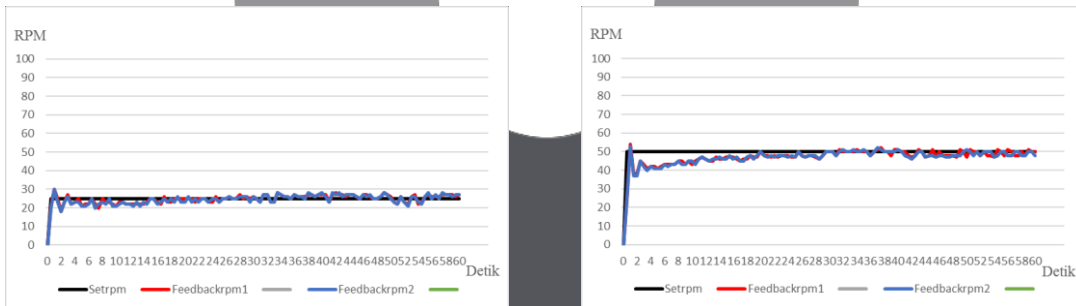
Gambar 14 Grafik Dengan Beban 11 Kg 25 RPM Dan 50 RPM



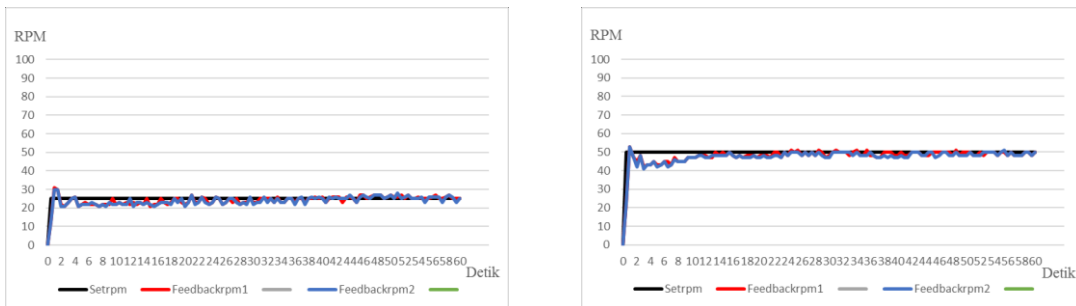
Gambar 15 Grafik Dengan Beban 12 Kg 25 RPM Dan 50 RPM



Gambar 16 Grafik Dengan Beban 13 Kg 25 RPM Dan 50 RPM



Gambar 17 Grafik Dengan Beban 14 Kg 25 RPM Dan 50 RPM



Gambar 18 Grafik Dengan Beban 15 Kg 25 RPM Dan 50 RPM

Untuk pengujian kecepatan rata-rata yang sudah dilakukan dapat disimpulkan bahwa kecepatan rata-rata dan percepatan rata-rata untuk 25 RPM adalah 0.313 m/s atau 1.127 km/jam dan 0.417 m/s², sementara untuk 50 RPM adalah 0.384 m/s atau 1.38 km/jam dan 0.512 m/s².

Dari pengujian-pengujian yang sudah dilakukan, analisis pertama adalah kontrol PI lebih baik jika dibandingkan dengan kontrol P, karena kontrol P pada saat *setpoint* 50 RPM tidak mampu untuk mencapai *settling time*. Jika kontrol PI dibandingkan dengan kontrol PID, maka kontrol PID lebih baik karena *overshoot* untuk *setpoint* 25 RPM yaitu 37 RPM untuk kontrol PID dan 41 RPM untuk kontrol PI, sementara untuk *setpoint* 50 RPM yaitu 62 RPM untuk kontrol PID dan 65 RPM untuk kontrol PI, meskipun *rise time* dari kedua kontrol tersebut hampir sama yaitu ± 0.75 detik. *Settling time* pada kontrol PI lebih baik jika dibandingkan dengan kontrol P yaitu 5 detik dan 7.5 detik untuk 25 RPM, dan untuk 50 RPM kontrol PI lebih baik juga karena kontrol P tidak mampu untuk mencapai *error steady state* atau keadaan tunak. *Settling time* pada kontrol PI juga sedikit lebih baik jika dibandingkan dengan PID yaitu 5.5 detik berbanding 8 detik untuk 25 RPM dan 26.5 detik berbanding dengan 18 detik untuk 50 RPM.

Pada analisis pengujian menggunakan beban dari 8 – 15 kg *rise time* yang didapatkan hampir sama yaitu ± 0.75 detik, hasil yang hampir sama tersebut terjadi karena sistem kontrol mobil anak ini menggunakan *time sampling* sebesar 0.5 detik, yang artinya sensor akan membaca perbedaan hitam dan putih dalam waktu 0.5 detik sekali, karena *time sampling* yang kecil maka kesalahan atau *error* dapat diperbaiki lebih cepat dibandingkan dengan *time sampling* yang lebih besar. Penambahan beban akan berpengaruh pada berkurangnya *overshoot*, seperti pada pengujian dengan kontrol PID tanpa beban dan pengujian dengan menggunakan beban 8 kg, *overshoot* pada *setpoint* 25 RPM untuk pengujian tanpa beban yaitu 37 RPM dan pengujian dengan menggunakan beban yaitu 33 RPM, sementara untuk *setpoint* 50 RPM pada pengujian tanpa beban yaitu 62 RPM dan pengujian dengan beban 8 kg yaitu 58 RPM. Pada pengujian dengan penambahan beban 1 kg mulai dari 8 kg sampai dengan 15 kg, pada pengujian dengan *setpoint* 25 RPM, *overshoot* turun $\pm 1-2$ RPM tetapi hanya beban dari 8 kg ke 9 kg dan 12 ke 13 kg. *Overshoot* pada pengujian dengan *setpoint* 50 RPM sama seperti dengan 25 RPM yaitu overshoot tidak terlalu berbeda jauh yaitu $\pm 1-2$ RPM pada beban dari 8 kg ke 9 kg, 9 ke 10 kg, 12 kg ke 13 kg dan 14 kg ke 15 kg. *Settling time* pada saat menggunakan beban dari 8 kg-15 kg cukup terlihat perbedaan dan pengaruhnya yaitu untuk 25 RPM $\pm 8.5-19$ detik dan untuk 50 RPM yaitu $\pm 22-31.5$ detik, walaupun nilai *settling time* untuk penambahan beban bertambah tetapi ada juga yang berkurang karena pengaruh dari kondisi jalan dan sensitifitas dari sensor yang mengakibatkan pengujian menjadi kurang akurat.

4. Kesimpulan

Pada penelitian kali ini didapat kesimpulan-kesimpulan sebagai berikut:

1. Nilai konstanta KP, KI dan KD yang paling baik pada sistem mobil anak yaitu KP = 4.75, KP2 = 3.75, KI = 0.2, KI2 = 0.15, KD = 0.8 dan KD2 = 0.75.
2. Penambahan beban 1 kg tidak terlalu berpengaruh pada berkurangnya *overshoot*, tetapi terdapat $\pm 1-2$ RPM hanya pada beban dari 8 kg ke 9 kg dan 12 ke 13 kg pada 25 RPM dan untuk 50 RPM yaitu $\pm 1-2$ RPM pada beban dari 8 kg ke 9 kg, 9 ke 10 kg, 12 kg ke 13 kg dan 14 kg ke 15 kg.
3. Pada saat menggunakan beban tambahan dari 8 kg-15 kg untuk 25 RPM dan 50 RPM, *settling time* cukup terlihat perbedaan dan pengaruhnya dengan nilai maksimal yaitu 19 detik dan 31.5 detik, walaupun nilai *settling time* untuk penambahan beban bertambah tetapi ada juga yang berkurang karena pengaruh dari kondisi jalan dan sensitifitas dari sensor yang mengakibatkan pengujian menjadi tidak akurat.
4. Sistem yang dibuat dapat mencapai titik *setpoint* yang diinginkan walaupun dengan beban yang berubah.

Daftar Pustaka:

- [1] Leitman, Seth and Brant, Bob, *Build Your Own Electric Vehicle*, 2nd ed. United States of America : McGraw-Hill, 2008.
- [2] Astrom, Karl J and Hagglund, Tore, *PID Controllers: Theory, Design and Tuning*, 2nd ed. United State of America : Instrument Society of America, 1995.
- [3] Visioli, Antonio, *Practical PID Control*, 1st ed. United of Kingdom : Springer, 2006.