

# SISTEM PEMBELAJARAN KONTROL PID (PROPORSIONAL INTEGRAL DERIVATIF) PADA PENGATUR KECEPATAN MOTOR DC

## PID (PROPORTIONAL INTEGRAL DERIVATIVE) CONTROL LEARNING SYSTEM ON DC MOTOR SPEED CONTROLLER

**Panji Saka Gilap Asa<sup>1</sup>, Sigit Priyambodo<sup>2</sup>, Subandi<sup>3</sup>**

<sup>1</sup>Penulis, Mahasiswa Alih Jalur S-1 Jurusan Teknik Elektro, IST Akprind Yogyakarta

<sup>2</sup>Pembimbing I, Dosen Jurusan Teknik Elektro, IST Akprind Yogyakarta

<sup>3</sup>Pembimbing II, Dosen Jurusan Teknik Elektro, IST Akprind Yogyakarta

**S1 Teknik Elektro IST Akprind Yogyakarta**

**Jl. Kalisahak 28 Yogyakarta**

---

### Intisari

Kendali otomatis telah memegang peranan yang sangat penting dalam perkembangan ilmu dan teknologi. Penerapan sistem kendali mencakup bidang yang sangat luas termasuk kestabilan pesawat ruang angkasa, peluru kendali, sistem pengendalian pesawat, dan sebagainya. Karena kemajuan dalam teori dan praktek, kontrol otomatis memberikan kemudahan dalam mendapatkan performa dari sistem dinamik, meningkatkan kualitas dan menurunkan biaya produksi sehingga meningkatkan laju produksi. Sehingga pemahaman yang baik dan mendalam di bidang sistem kendali menjadi suatu hal yang wajib bagi ilmuwan sekarang.

Penelitian ini, terfokus pada aplikasi salah satu jenis kendali otomatis yaitu kendali PID (*Proportional, Integral, Derivative*). Untuk mempermudah pemahaman konsep PID perlu dibuat sistem yang bisa menggambarkan *set-point* dan membaca nilai keadaan sesungguhnya dalam hal ini kecepatan motor secara real time. Yaitu dengan memanfaatkan *user-interface* yang dibuat dengan IDE Processing yang dihubungkan dengan Arduino sebagai pengolah data. Sedangkan untuk sistem hardwarenya sendiri telah dibuat *prototype* berupa kendali kecepatan motor DC.

Tujuan utama adalah untuk mempermudah pemahaman pengaruh parameter *Proportional, Integral* dan *Derivative* pada tanggapan transien suatu sistem. Baik buruknya, stabil tidaknya sistem kendali dapat dilihat dari nilai tanggapan transient yang berupa *Steady state error, Rise time, Settling time, dan Overshoot*.

**Kata kunci:** Kendali Otomatis, Algoritma PID, *Real-Time*, Motor DC, Tanggapan Transien.

---

### Abstract

*Automatic control has a very important role in the development of science and technology. Implementation of the control system covers a very wide field, including the stability of the spacecraft, missiles, aircraft steering system, and so on. Because of advances in theory and practice, automatic controls provide ease in getting performance of dynamical systems, improving quality and reducing production costs and increasing the production rate. So good and deep understanding in the field of the control system into a mandatory thing for a scientist nowadays.*

*This study, focused on the application of one type of automatic control, namely control of the PID (Proportional, Integral, Derivative). To facilitate understanding of the concept of PID need to be created a system that can describe the set-point and read the value of the real situation in this case the motor speed in real time. Namely by using user-interface made with IDE Processing connected with the Arduino as a data processor. As for the hardware itself has created a system prototype in the form of a DC motor speed control.*

*The main purpose is to facilitate the learning process of the influence on the Proportional, Integral and Derivative parameters to the transient response of a system. was it good or bad, whether stable control system can be seen from the transient response in the form Steady state error, Rise time, settling time, and overshoot.*

**Keywords:** Automatic control, PID algorithm, *Real-Time*, DC Motor, Transient Response.

---

## I. PENDAHULUAN

Kendali otomatis telah memegang peranan yang sangat penting dalam perkembangan ilmu dan teknologi. Penerapan sistem kendali mencakup bidang yang sangat luas termasuk kestabilan pesawat ruang angkasa, peluru kendali, sistem pengendalian pesawat, dan sebagainya. Karena kemajuan dalam teori dan praktek, kendali otomatis memberikan kemudahan dalam mendapatkan performansi dari sistem dinamik, meningkatkan kualitas dan menurunkan biaya produksi sehingga meningkatkan laju produksi. Sehingga pemahaman yang baik dan mendalam dibidang sistem kendali menjadi suatu hal yang wajib bagi ilmuwan sekarang, [6].

perhitungan model matematika yang kompleks membuat mata kuliah sistem kendali justru menjadi momok yang menakutkan bagi sebagian besar mahasiswa teknik elektro maupun elektronika. Padahal sebagai ilmuwan maupun praktisi dibidang kendali, tidak akan terlepas dari pengetahuan otomasi industri. Sehingga perlu dibuat terobosan yang mampu menjadikan pembelajaran PID (Proportional, Integral, Derivative) tanpa pemodelan matematika kompleks sehingga dapat membantu proses pemahaman [7].

Studi ini, terfokus pada aplikasi salah satu jenis kendali otomatis yaitu kendali PID. Untuk mempermudah pemahaman konsep PID perlu dibuat sistem yang bisa menggambarkan set point dan membaca nilai keadaan sesungguhnya dalam hal ini kecepatan motor secara real time. Yaitu dengan memanfaatkan user interface yang dibuat dengan IDE Processing sebagai penampil grafik transien yang disambung dengan Arduino sebagai pengolah data. Sedangkan untuk sistem hardwarenya sendiri telah dibuat prototype berupa kendali kecepatan motor DC.

## II. TINJAUAN PUSTAKA

Penelitian yang berhubungan dengan sistem kendali otomatis sudah banyak dilakukan. Penelitian penelitian tersebut antara lain telah dilakukan oleh Baskoro (2009), Drian (2013), Nugraha (2007) Wijanarko (2009), Hastra (2010), dan Rifai (2014).

Penelitian yang dilakukan oleh Baskoro adalah pembuatan sistem pengaturan PID (*Proportional, Integral, Derivative*) posisi linier motor DC dengan mikrokontroler. Metode penelitian yang digunakan dari penelitian tersebut, ialah menggunakan eksperimen dengan melakukan perancangan hardware maupun software. Hasil penelitian yang dicapai, ialah untuk setiap masing - masing set point (posisi yang diinginkan) didapat nilai  $K_p$ ,  $K_i$ , dan  $K_d$  optimal untuk mencapai respon sistem yang baik. Respon sistem yang baik berarti mempunyai *error* yang kecil atau mendekati nilai set point yang diinginkan dan waktu respon yang cepat. Untuk set point 15 cm dengan nilai  $K_p = 850$ ,  $K_i = 7$ ,  $K_d = 8000$  *error* sebesar -0.1 dengan waktu respon 0.33s. Untuk set point 40 cm dengan nilai  $K_p = 2000$ ,  $K_i = 80$ ,  $K_d = 8000$ , didapat *error* sebesar -0.6 dengan waktu respon 1.21s. Untuk set point 75 cm yaitu nilai  $K_p = 12000$ ,  $K_i = 100$ ,  $K_d = 30000$  didapat *error* sebesar 0.2 dengan waktu respon 1.94s [1].

Telah dibuat prototype sistem kendali PID untuk kecepatan motor DC berbasis mikrokontroler ATmega16. Sensor optocoupler digunakan untuk membaca *input* yang

berasal dari putaran motor DC yang dihubungkan dengan rotary encoder kemudian oleh mikrokontroler ATmega16 akan dikonversi menjadi kecepatan motor DC, sedangkan kendali PID digunakan sebagai umpan balik untuk mengoreksi kecepatan motor DC. Kecepatan motor DC dalam RPM (Rotation Per Minute) [3].

Terdapat penelitian robot lengan 3 sumbu yang memiliki kemampuan Point To Point (PTP) oleh Nugraha. Robot ini digunakan untuk memindahkan benda dari titik yang berubah-ubah. Secara garis besar robot ini menerapkan pengendalian motor DC dengan menggunakan mikrokontroler AT89C51. Prinsip pergerakan robot ini adalah penggunaan motor DC sebagai penggerak mekanik robot dengan menetapkan panjang langkah pergerakan menggunakan switch sebagai sensor jumlah counter panjang pergerakan [5].

Penelitian tentang simulasi speedometer dan odometer digital berbasis mikrokontroler ATmega8535. Penelitian ini menggunakan roda pencacah berupa piringan CD dengan satu lubang yang terhubung dengan oros motor DC 12 Volt. Roda pencacah tersebut digunakan sebagai sinyal *input* yang dibaca oleh sebuah sensor optocoupler dengan tipe "U". Data tersebut akan diolah di dalam sistem minimum mikrokontroler ATmega8535 untuk diproses menjadi nilai kecepatan dan jarak tempuh roda pencacah kemudian hasilnya ditampilkan pada LCD 2x16 [8].

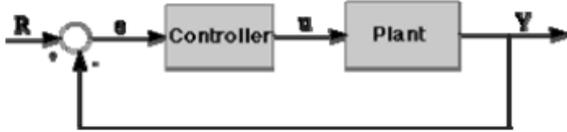
Penelitian tentang navigasi robot penjejak garis dengan kendali PID. Penelitian ini merancang sebuah robot yang berjalannya mengikuti garis hitam dengan dasar putih. Sensor cahaya digunakan sebagai *input* dalam pengendalian PID berdasarkan intensitas cahaya yang diterima. Selanjutnya hasil *input* tersebut diolah oleh mikrokontroler untuk mengatur kecepatan gerak motor DC pada robot penjejak [4].

Penelitian oleh rifai sudah terdapat penelitian aksi kendali PID terlihat sistem sudah mempunyai tanggapan *transient* yang cukup bagus, bisa dilihat dari *rise time* kecil, nilai *overshoot* kecil dan offset hilang sama sekali. Namun penelitian belum berakhir, perlu diuji sekali lagi dengan mengubah ubah nilai setpoint atau kecepatan yang diinginkan. Apakah sistem akan merespon sesuai dengan kecepatan tersebut [7].

Dari sejumlah penelitian yang disebutkan diatas, yang membahas tentang dinamika tanggapan *transient* hanya penelitian Baskoro [1], dan Rifai [7] Itupun masih sekedar fokus pada waktu *response*, belum termasuk nilai *overshoot* dan nilai kesalahan *offset*. Dan bentuk grafik yang masih kasar serta ketergantungan software berbayar yaitu microsoft excel.

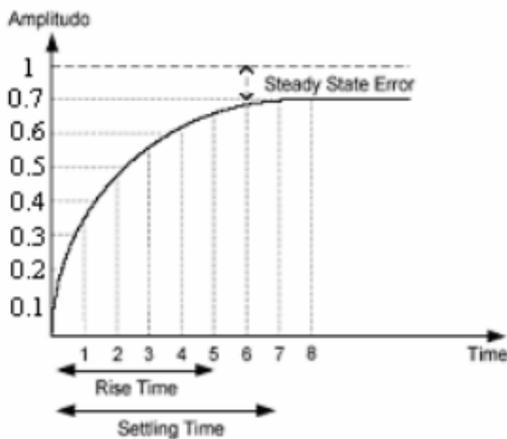
## III. KENDALI PID

Kendali PID (*proportional-integral-derivative*) digunakan dalam sebuah sistem *loop* tertutup yang melibatkan umpan balik dari *output* sistem guna mencapai respon yang diinginkan. Sistem PID dapat mengendalikan variabel *input* dengan memanipulasi variabel *output* sehingga diperoleh variabel *input* baru agar menghasilkan *output system* yang sesuai. Contoh dapat dilihat blok diagram pada gambar 1.



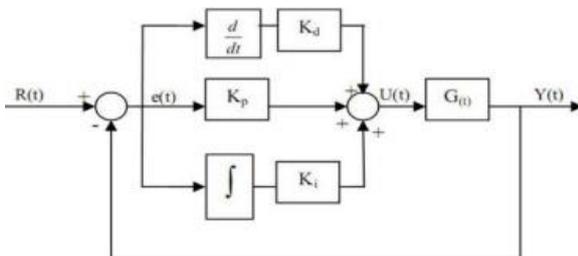
Gambar 1. blok diagram sistem closed loop.

Sistem kontrol *closed loop* dimana kontroler bekerja sebagai penggerak *plant* ( objek fisik yang digerakkan dalam sistem ) dan mengontrol sifat *plant*. Sistem PID sebagai kontroler akan bekerja untuk menggerakkan *plant* sebagaimana ia seharusnya menghasilkan respon yang diinginkan. Yang dikontrol oleh sistem PID adalah variabel *output* sistem yaitu Y. Agar diperoleh variabel Y yang sesuai maka sistem PID akan memanipulasi variabel *input* R. Variabel yang dimanipulasi (R baru) merupakan hasil komputasi dari variabel R, Y (*feedback*) dan sinyal *error* (e). Sinyal *error* ini dihasilkan oleh *output* Y yang dibawa dalam komponen *feedback* untuk dikirim ke PID kontroler sehingga dapat dijadikan pengukuran *error output*. Dari variabel manipulasi inilah, diperoleh *output* yang sesuai dengan *error* yang minimum seperti pada gambar 2 [6].



Gambar 2. Grafik keadaan sistem.

Algoritma PID yang merupakan bentuk dari manipulasi *error* dari system sebelumnya, bentuk dari lintasan proses dijabarkan dalam diagram blok seperti ditunjukkan gambar 3. Parameter parameter tersebut yang akan mempengaruhi kerja sistem secara keseluruhan, masing masing akan memiliki efek – efek yang *distinctive*.

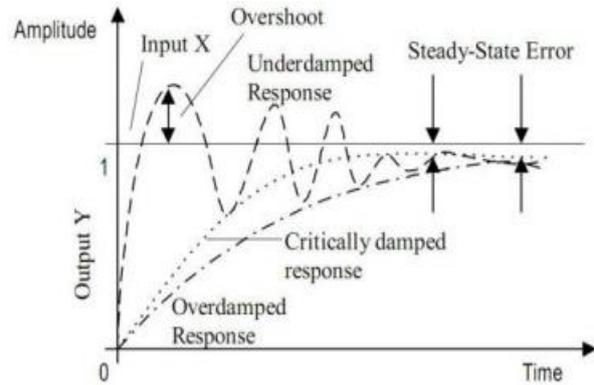


Gambar 3. Diagram Blok kendali PID.

#### IV. TANGGAP TRANSIEN

Ketika *input* sebuah sistem berubah secara tiba-tiba, keluaran atau *output* membutuhkan waktu untuk merespon

perubahan itu. Bentuk respon *transient* atau peralihan bias ditunjukkan seperti gambar 4.



Gambar 4. Bentuk sinyal tanggapan transien.

Pendekatan yang paling banyak dipakai pada sistem kendali adalah sistem orde II yang mempunyai spesifikasi ukuran kualitas tanggapan *transient* sebagai berikut:

**Rise Time (TR):** Ukuran waktu yang di ukur mulai respon mulai  $t = 0$  s/d respon memotong sumbu *steady state* yang pertama.

**Settling Time (TS):** Ukuran waktu yang menyatakan respon telah masuk 5% atau 2% atau 0,5% dari respon *steady state*.

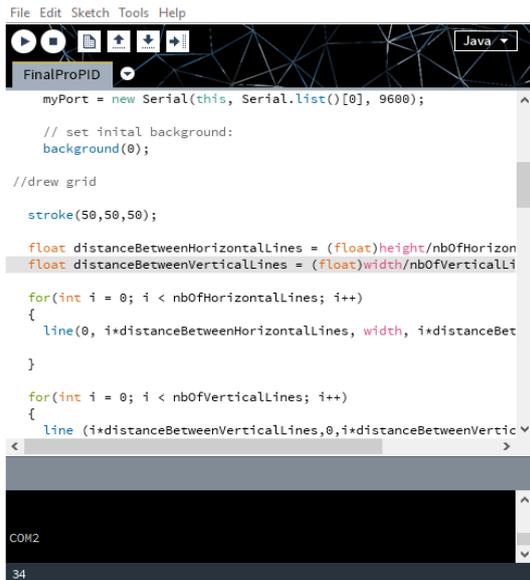
**Delay Time (TD) :** Ukuran waktu yang menyatakan factor keterlambatan respon *output* terhadap *input*, di ukur mulai  $t = 0$  s/d respon mencapai 50% dari respon *steady state*.

**Overshoot (MP) :** Nilai relatif yang menyatakan perbandingan harga maksimum respon yang melampaui harga *steady state* dibanding dengan nilai *steady state*.

**Offset:** selisih *error* yang terjadi antara nilai yang diinginkan dan nilai sebenarnya [2].

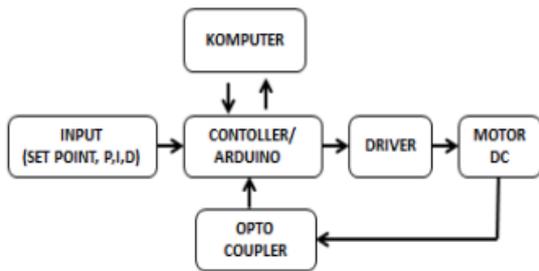
#### V. PERANCANGAN SISTEM

Algoritma PID akan diterapkan pada *system real-time* yang artinya akan secara langsung diamati setiap perubahan yang mungkin terjadi pada setiap saat. sehingga dibutuhkan software khusus yang bisa membaca data dari Arduino Uno menggunakan *serial communication*. Software yang digunakan pada penelitian skripsi ini adalah Processing yang mampu menghubungkan data dari Arduino langsung digambarkan pada layar komputer, *software* ini merupakan *freeware* yang bisa kita *download* secara gratis. Gambar 5. menunjukkan *screenshot* Processing. Processing menggunakan struktur pemrograman yang sama dengan Arduino sehingga keduanya mudah dipelajari bersamaan.



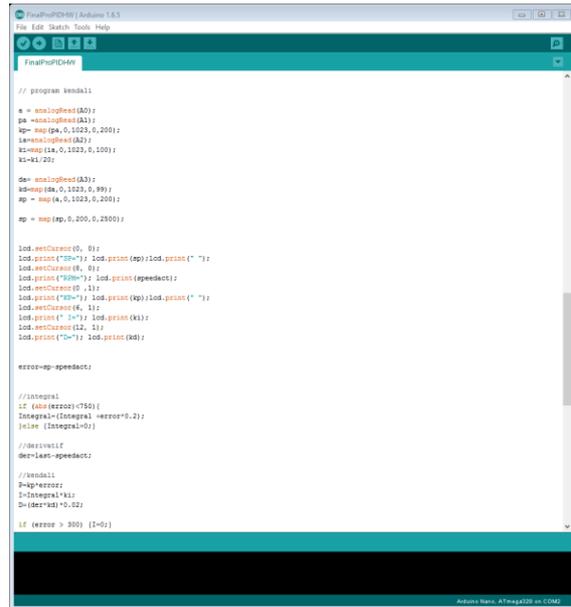
Gambar 5. IDE Processing.

Actuator yang dipakai adalah Motor DC dengan driver sederhana menggunakan transistor NPN untuk mengendalikan kecepatan motor. Kecepatan motor akan dideteksi oleh optocoupler yang berupa pulsa sebagai acuan perhitungan kecepatan putar motor DC dalam satuan RPM yang akan diteruskan dalam proses PID didalam arduino. Arduino akan terus melakukan penghitungan dan proses pembacaan RPM yang diteruskan dalam proses PID, sehingga nilai kendali motor dapat di aktuasi langsung ke motor dan dikirimkan langsung ke antarmuka Processing. Berikut diagram kerja sistem:



Gambar 6. Diagram Blok Sistem.

Dari blok diagram gambar 6. terlihat bahwa input sistem terdiri dari Set Point, Parameter P,I,D dan sinyal hasil sensor optocoupler. Set Point digunakan untuk menentukan nilai kecepatan yang diinginkan (RPM) sedangkan parameter (PID) digunakan untuk merubah performa sistem dan menentukan perubahan yang terjadi pada tanggapan transient.



Gambar 7. IDE Arduino

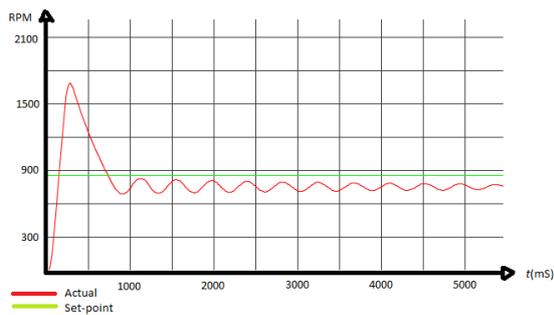
Untuk memprogram controller arduino, digunakan IDE dengan struktur yang sama dengan processing seperti ditunjukkan gambar 7, seluruh algoritma kontrol dan kirim terima data pada aktuasi dan proses pembacaan RPM semua ditulis dalam bahasa pemrogramannya. Dan untuk mengatur parameter-parameter yang ada, dibuatlah display dan knob pada sisi hardware sistem seperti pada gambar 8.



Gambar 8. Antarmuka Parameter Set.

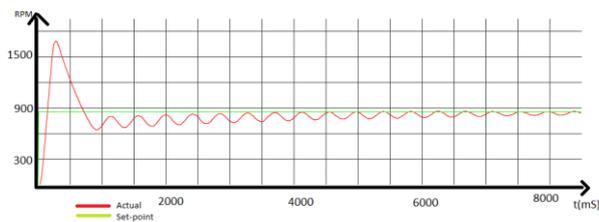
## VI. Hasil Dan Analisa

Perlu diingat bahwa penelitian ini bertujuan untuk mempermudah pemahaman konsep dan algoritma PID beserta pengaruhnya terhadap performansi sistem dengan mengaplikasikan secara langsung pada real time. Sehingga data yang digunakan sebagai hasil tidak harus berupa data numerik, tetapi lebih kearah grafik perubahan sistem terhadap parameter PID. Gambar 9. menunjukkan performa sistem ketika hanya diberi masukan parameter *Proportional* saja.



Gambar 9. Hasil Kendali P pada Software antarmuka

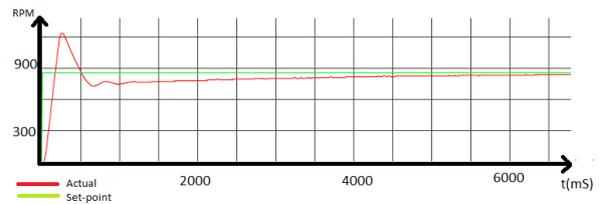
Set point yang diberikan pada sistem sebesar 850 rpm, dan hanya menggunakan aksi kendali *Proportional*. Dari hasil gambar 9. terlihat bahwa kendali *proportional* saja tidak mampu membuat sistem menjadi stabil., Alih alih memberikan offset minimal, bahkan untuk stabil pada kecepatan tertentu saja tidak bisa. Hal ini dikarenakan kendali *proportional* akan memberikan aksi atau *power* sesuai dengan nilai *error*. Semakin tinggi *error*, semakin kuat *power* yang dihasilkan, akan tetapi bila sampai keadaan dimana tidak ada *error*, maka *power* juga ikut hilang yang menyebabkan putaran motor DC drop, dan akan naik lagi sesuai *error*, begitu seterusnya. Sehingga dibutuhkan aksi kendali tambahan untuk mempertahankan *power* yaitu parameter *Integral*.



Gambar 10. Hasil Kendali PI pada Software antarmuka

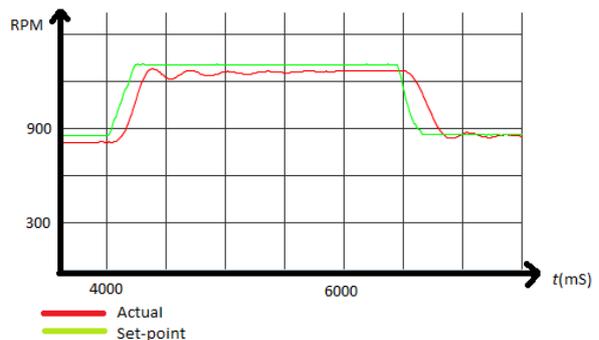
Dapat diamati pada gambar 10. dimana offset dan osilasi dari data aktual dapat diredam, menunjukkan bahwa ketika system diberi aksi kendali *proportional* ditambah *integral*, terlihat bahwa pada awalnya memang masih terjadi osilasi dan tidak stabil, namun pada akhirnya kecepatan motor DC akan stabil mendekati kecepatan yang diinginkan yaitu 850 rpm. Hal ini dikarenakan adanya aksi *integral* yang memberikan *power* setara dengan hasil penjumlahan nilai *error* sebelumnya. Sehingga pada titik tertentu hasil akumulasi *error* tersebut sebanding dengan *power* yang dibutuhkan untuk mempertahankan kecepatan motor DC. Setelah percobaan kedua ini dapat disimpulkan sementara bahwa aksi kendali *integral* dapat menghilangkan offset yang terjadi pada sistem. Namun *overshoot* yang terjadi pada aksi P, juga masih terjadi pada aksi PI, sehingga dibutuhkan aksi kendali tambahan untuk minimal mengurangi adanya *overshoot*, yaitu aksi kendali *Derivative*.

Gambar 11 menunjukkan performa system ketika diberi masukan parameter *Proportional*, *Integral* dan *Derivative*. Hasil transien seperti pada Gambar 11 menunjukkan bahwa penambahan aksi kendali *Derivative* secara signifikan akan mengurangi adanya *overshoot* pada sistem. Dimana titik kerja kecepatan aktual mengalami *overshoot* yang lebih kecil daripada sistem yang tidak menggunakan kendali D, dan osilasi yang teredam dapat dilihat bahwa efek redaman tersebut sangat signifikan.



Gambar 11. Hasil Kendali PID pada Software antarmuka.

Hal ini dikarenakan *power* berlebih hasil dari kendali *proportional* akan diredam oleh *derivative* dengan melihat selisih nilai *error* sekarang dengan nilai *error* sebelumnya. Dengan menggunakan aksi kendali PID terlihat sistem sudah mempunyai tanggapan *transient* yang cukup bagus, bisa dilihat dari *rise time* kecil, nilai *overshoot* berkurang dan offset hilang sama sekali. Namun penelitian belum berakhir, perlu diuji sekali lagi dengan mengubah ubah nilai setpoint atau kecepatan yang diinginkan. Apakah sistem akan merespon sesuai dengan kecepatan tersebut. Gambar 12 menunjukkan aksi kendali PID dengan nilai setpoint yang bervariasi dari 850 rpm, kemudian dinaikkan ke 1200 rpm dan diturunkan lagi ke kecepatan 850 rpm. Respon sistem juga dapat mengikuti kecepatan setpoint. Hal ini menunjukkan bahwa parameter PID bekerja dengan baik.



Gambar 12. Hasil Kendali PID pada Software antarmuka dengan setpoint yang diubah.

Dengan memvariasikan nilai setpoint, dapat dilihat bahwa nilai aktual dan setpoint dapat saling mendekati dengan nilai yang selalu mendekat dengan setpoint dan tentunya tetap stabil.

## VII. Kesimpulan & Saran

Kesimpulan yang dapat diambil adalah:

1. Pemahaman Algoritma kendali PID lebih mudah bila diterapkan secara real time untuk memahami dinamika tanggapan *transient*.
2. Dengan melihat grafik grafik tersebut, dapat kita buktikan dengan mudah dimana parameter parameter tersebut akan berpengaruh pada nilai aktual bacaan terhadap setpoint.
3. Parameter *Proportional* (P) berperan dalam mempercepat *rise time*, tetapi tidak menjamin kestabilan sistem.
4. Parameter *Integral* (I) digunakan untuk menghilangkan offset atau *error* kestabilan yang terjadi pada sistem.
5. Parameter *derivative* (D) berperan untuk meredam atau menurunkan *overshoot*.

Saran untuk penelitian selanjutnya adalah mencoba untuk membuat sistem hardware software dengan optimasi yang lebih seperti penambahan tombol dan mode percobaan agar lebih mempermudah pembelajaran selanjutnya.

## REFERENSI

- [ 1 ] Baskoro, Dwi., 2009. Perancangan Pengendali Posisi Linier Untuk Motor DC dengan Menggunakan PID, Universitas Bina Nusantara, Jakarta.
- [ 2 ] Benjamin, C. Kuo., 1975, Automatic Control Systems ( 3rd.ed.), Prentice-Hall Inc.
- [ 3 ] Drian, D.N., 2013, Perancangan Sistem Kendali Pid Untuk Kecepatan Motor DC berbasis Mikrokontroler Atmega16, Universitas Atma Jaya, Yogyakarta
- [ 4 ] Hastra, M.R., 2010, Robot Penjejak Garis dengan Kendali PID, Skripsi Program Studi Teknik Industri, Universitas Atma Jaya, Yogyakarta.
- [ 5 ] Nugraha D, W., 2011, Pengendalian Robot Yang Memiliki Lima Derajat Kebebasan, Jurnal Ilmiah Foristek Vol.1, No. 1
- [ 6 ] Ogata, Katsuhiko, 1997, Teknik Kontrol Automatik Jilid 1 Edisi Kedua, Erlangga, Jakarta.
- [ 7 ] Rifai, Isnaini N., 2014, Penerapan Algoritma Kendali PID pada sistem real time untuk mempelajari tanggap Transien, Prosiding SENTIA Politeknik Negeri Malang, Malang.
- [ 8 ] Wijanarko, A.P., 2009, Simulasi Speedometer dan Odometer Digital Berbasis Mikrokontroler ATmega8535, Universitas negeri Yogyakarta.