

# KONTROL POSISI PUTARAN MOTOR DC MENGUNAKAN ARDUINO DAN PID

Adhitiya Satiawan Suryadata<sup>1)</sup>, Agus Halim<sup>2)</sup> dan Rosehan<sup>2)</sup>

<sup>1)</sup>Mahasiswa Teknik Mesin, Fakultas Teknik Universitas Tarumanagara

<sup>2)</sup>Dosen Program Studi Teknik Mesin, Fakultas Teknik Universitas Tarumanagara

e-mail: <sup>1)</sup>adhitiya.515180040@stu.untar.ac.id, <sup>2)</sup>agush@ft.untar.ac.id, <sup>3)</sup>rosehan@ft.untar.ac.id

**Abstract:** Industrial technology's development is growing, especially in the field of control systems. One often-used controller, PID, is composed of proportional, integrals, and derivatives, which each control has a different role. In determining each control's value, trials are sometimes needed to find the correct value. The advantages of this type of controller are that the algorithm used is simple but with accurate output. In this study, a potentiometer is used for determining the set point and then an encoder that helps read DC motor rotation and Arduino Uno as a medium in implementing the PID system. This article aims to be a learning medium to determine the parameters of the PID. The method is a systematic method that starts with making a literature study to determine the detailed requirements. After that, it continues with the microcontroller circuit's design using the Tinkercad website and doing 2D and 3D formats. The method results that this circuit can work perfectly because every input given by the potentiometer can drive the motor smoothly and accurately through the encoder reading.

**Keywords:** Arduino; PID; DC motor; Encoder

## PENDAHULUAN

Seiring dengan kemajuan teknologi industri, maka kebutuhan akan sistem aktuator maupun sistem kontrol akan meningkat. Aktuator digunakan sebagai penggerak komponen bersama dengan sistem kontrol untuk dapat diaplikasikan pada berbagai bidang. Dalam melakukan kontrol putaran motor DC. Arduino Uno berperan sebagai mikrokontroler dan Sistem kontrol PID sehingga perputaran motor dapat gegas dan akurat.

Sistem PID merupakan sistem kendali tertutup yang menggunakan umpan balik [1]. Sistem kontrol ini banyak digunakan pada dunia industri karena algoritma yang sederhana namun dengan *output* yang baik. Konfigurasi standar kontroler PID memiliki parameter-parameter *proportional*, *integrals*, dan *derivatives* yang dihitung agar karakteristik variabel sesuai dengan kriteria desain yang diinginkan [2]. Parameter umum dalam desain adalah *rise-time*, *settling-time*, *overshoot* dan *steady state error* [3].

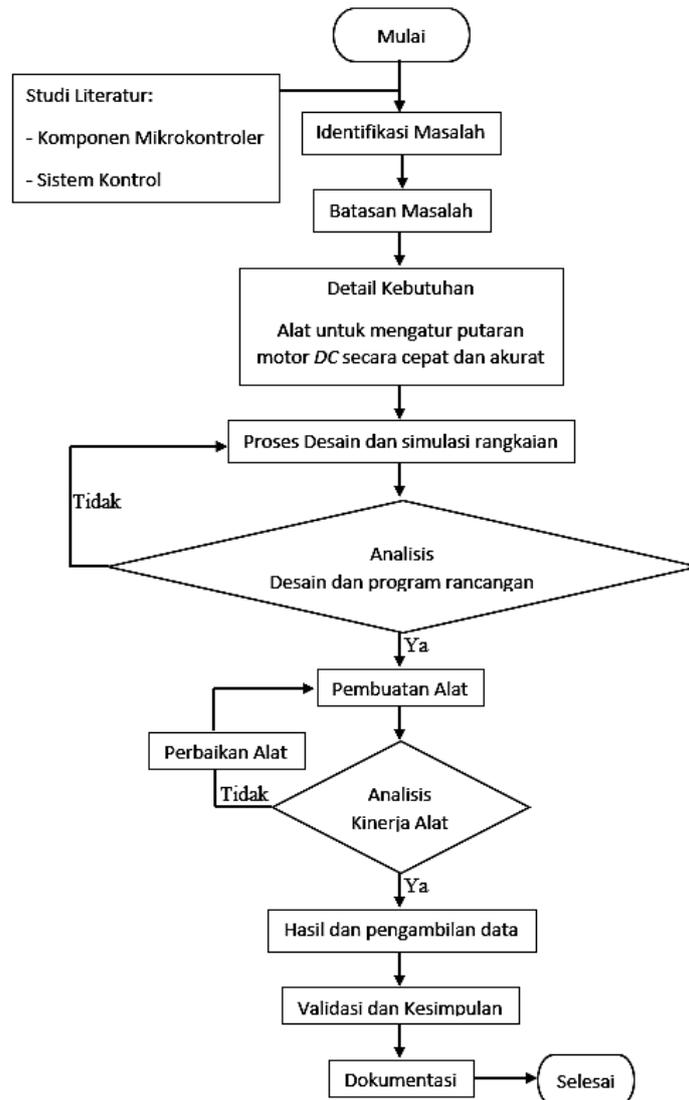
Sebagai sumber *input* terhadap mikrokontroler digunakan potensiometer yang nilai resistansi dapat diatur sesuai dengan kebutuhan rangkaian elektronika ataupun kebutuhan pengguna [4]. Dalam membaca putaran motor digunakan *encoder* dengan jenis magnetik, komponen ini dapat membaca putaran motor secara presisi [5]. Mikrokontroler Arduino digunakan sebagai pusat kendali rangkaian ini karena kemudahan dalam melakukan pemrograman dan komponen pendukung yang beragam [6]. Setelah mikrokontroler melakukan proses maka dihubungkan dengan *driver* motor L298N yang berfungsi sebagai mengontrol kecepatan serta arah perputaran motor DC.

Dalam melakukan perancangan juga terdapat batasan seperti merancang komponen yang ringkas dengan kinerja yang baik, pemilihan motor DC *built in encoder* yang terbatas oleh dimensi dan daya yang digunakan, penggunaan potensiometer jenis analog yang menyebabkan putaran *setpoint* terbatas.

Dengan masalah yang sudah ada maka perancangan ini bertujuan untuk menentukan variabel sistem kontrol PID yang tepat sebagai pengatur arah pada motor DC, dengan ketepatan putaran motor DC dengan potensiometer maka dapat dimanfaatkan dalam hal seperti mengontrol bukaan valve pada celah yang kecil atau dari jarak jauh, maupun mengatur bukaan *throttle* pada sebuah mesin agar dapat menjaga putaran mesin, mengatur pergerakan *automated guided vehicle*, robot *humanoid*, dan sistem *drone*. Selain itu dapat bertujuan untuk menganalisa komponen-komponen yang dibutuhkan serta mengolah data yang telah didapatkan oleh sistem.

## METODE PENELITIAN

Dalam melakukan perancangan kontrol posisi putaran motor *DC* dengan arduino dan *PID* maka digunakan metode sistematis.



Gambar 1. *Flowchart* perancangan

Gambar 1 menunjukkan bahwa perancangan diawali dengan melakukan pengumpulan studi literatur dilanjutkan dengan melakukan identifikasi masalah pada perancangan alat. Tahap selanjutnya dilakukan pemilihan mikrokontroler, beserta komponen pendukung. Setelah itu dilakukan desain rangkaian mikrokontroler pada *website tinkercad*, pada *website* ini juga dapat dilakukan percobaan pembuatan program. Kemudian rangkaian akan dibuat serta penyusunan program agar bisa dilakukan pengujian dan pengambilan data dengan cara mencatat perputaran yang dilakukan oleh motor *DC* setelah potensiometer diputar, sehingga akan didapatkan kesimpulan dari hasil perancangan ini. Hasil dari perancangan alat ini berupa rangkaian komponen dan sistem kontrol dari *PID*.

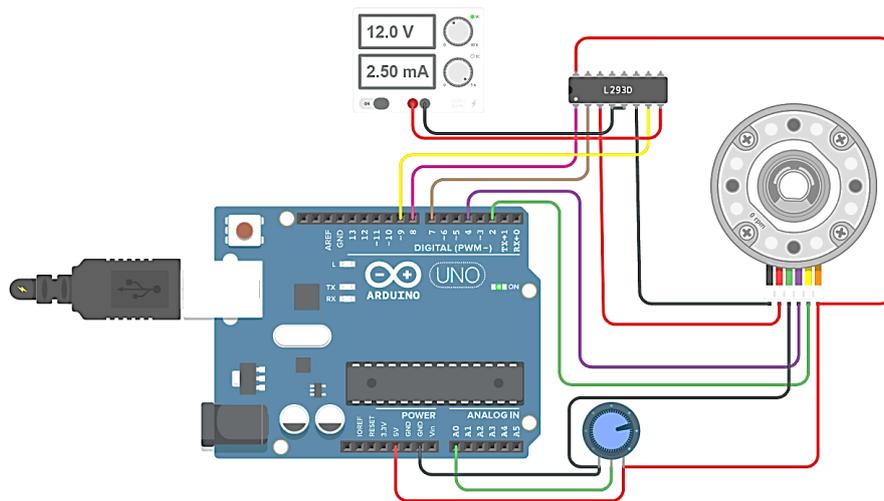
Untuk komponen yang digunakan dalam perancangan ini terdiri dari Arduino Uno sebagai pusat pemrosesan data. *Driver* motor L298N difungsikan untuk mengontrol kecepatan serta arah perputaran motor *DC*. Potensiometer sebagai *setpoint* dalam menjalankan sistem. Motor *DC with encoder* sebagai pelaksana *output* yang diberikan dan dengan sistem yang sudah terintegrasi dengan *encoder* maka memudahkan dalam pembacaan putaran motor. Serta *power supply* sebagai sumber daya untuk Arduino Uno dan *Driver* motor.

## HASIL DAN PEMBAHASAN

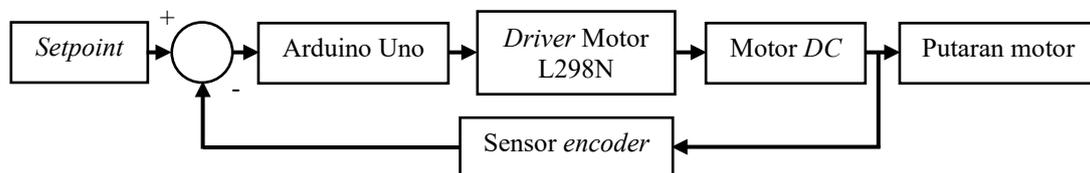
Kontroler adalah komponen yang berfungsi meminimasi sinyal kesalahan. Tipe kontroler yang paling populer ialah kontroler *PID*. Elemen-elemen kontroler *P*, *I* dan *D* masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar. Kontrol *PID* merupakan kontroler mekanisme umpan balik yang banyak dipakai pada sistem kontrol industri. Sebuah kontroler *PID* secara kontinu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur.

Kontrol *PID* terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu *P* (*Proportional*) *Controller* yang bertanggung jawab untuk nilai kesalahan saat ini. Nilai *P* yang besar akan mengakibatkan sistem tidak stabil, *I* (*Integral*) *Controller* yang bertanggung jawab untuk penjumlahan dari *error* sebelumnya. Proses penjumlahan *error* akan berlanjut hingga nilai dari output sistem sama dengan nilai referensi yang diinginkan. Penggunaan kontrol *I* biasanya digabung dengan *P* yang disebut kontrol *PI*. Jika hanya menggunakan kontrol *I* saja akan memberikan respon yang lambat dan mengakibatkan sistem berosilasi. *D* (*Derivative*) *Controller* yang bertanggung jawab untuk nilai dari tingkat perubahan *error*. Kontrol ini akan meningkatkan respon sistem. Nilai besar pada kontrol *D* menyebabkan sistem tidak stabil. Respon dari kontrol *PD* memberikan respon *rising* yang cepat daripada kontroler *P*. Pada dasarnya kontrol *D* mempunyai sifat sebagai *highpass filter* pada sinyal *error*. Berdasarkan sifat masing-masing kontroler memiliki parameter tertentu yang harus diatur untuk dapat beroperasi dengan baik, yang disebut sebagai konstanta.

Berikut ini merupakan hasil perancangan dari kontrol posisi putaran motor *DC* menggunakan arduino dan *PID*. Gambar 2 merupakan penyusunan komponen yang telah dilakukan pada *website*.



Gambar 2. Perancangan pada tinkercad [7]



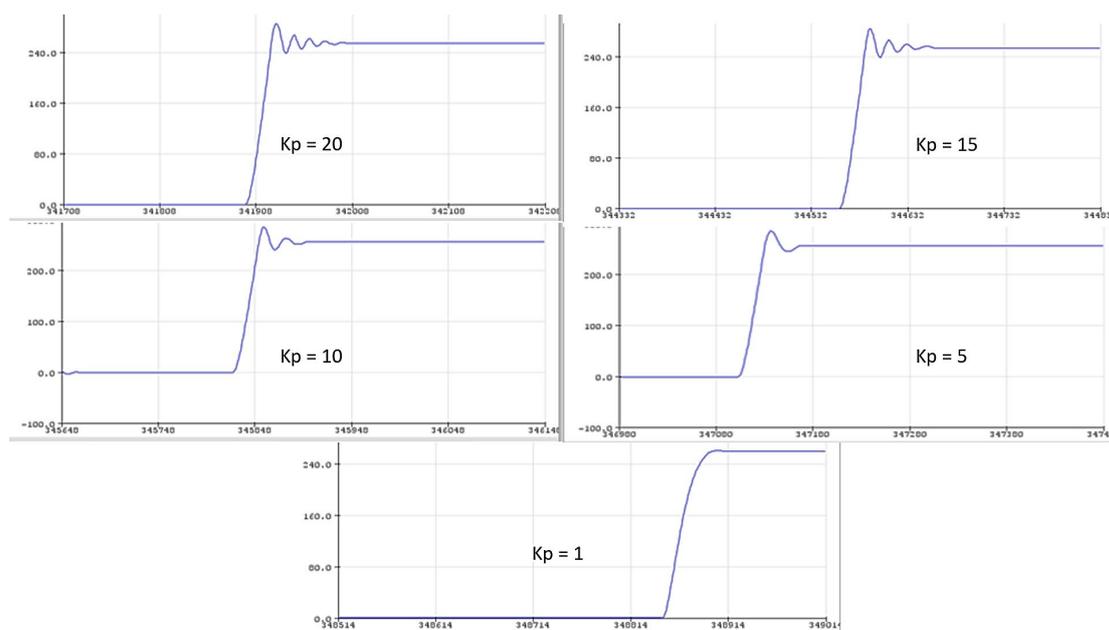
Gambar 3. Blok diagram kontrol sistem

Pada Gambar 3 sistem dimulai dengan mikrokontroler melakukan inisialisasi pada komponen yang terpasang, dilanjutkan dengan pengaturan *setpoint* oleh potensiometer. Kemudian mikrokontroler akan membaca nilai analog yang dihasilkan dari potensiometer dan akan dikalkulasi. Setelah mikrokontroler selesai melakukan proses perhitungan, maka mikrokontroler akan memberikan nilai *output* pada *driver* motor yang akan menggerakkan motor *DC*.

Perputaran motor *DC* akan dibaca oleh *encoder* yang dapat memberikan umpan balik pada mikrokontroler. Sistem *PID* memungkinkan putaran motor sesuai dengan *setpoint* yang telah diberikan. Setelah dilakukan perakitan dan pemasangan komponen perancangan dilanjutkan dengan penyusunan program sistem kontrol *PID* menggunakan aplikasi Arduino IDE. Berikut ini merupakan hasil dari pengujian pada sistem yang telah dirancang pengujian dilakukan pada pembacaan nilai *encoder* dan nilai potensiometer pada *serial monitor* di aplikasi Arduino IDE,

Nilai P, I, dan D akan ditulis sebagai  $K_p$ ,  $K_i$ , dan  $K_d$  dalam program Arduino IDE. Pembacaan yang dilakukan oleh Arduino terhadap potensiometer merupakan nilai analog yang dapat diubah menjadi nilai digital dengan dapat melakukan pembagian maupun perkalian. Pada percobaan diawali pada kondisi nilai potensiometer 0 maka *encoder* akan mengatur motor hingga menunjukkan putaran 0, dan pada saat potensiometer diputar hingga maksimal maka *encoder* akan memutar motor hingga nilai yang sama dengan potensiometer.

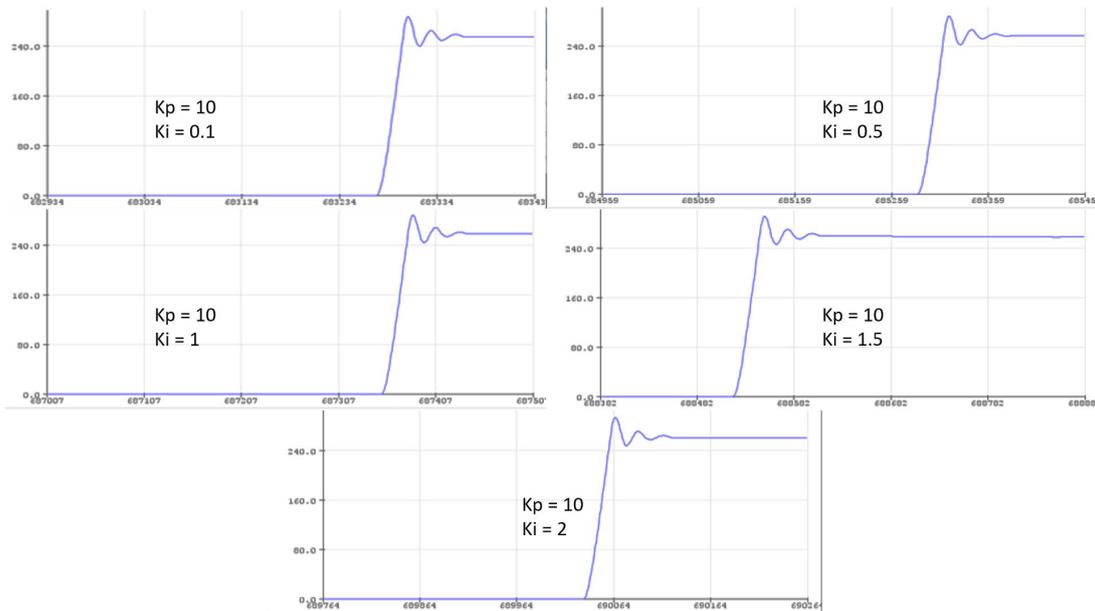
Dalam menentukan nilai pada *PID* maka dilakukan metode *trial-error* dimana pada *software* Arduino IDE pertama akan dimasukan nilai kontrol P ( $K_p$ ) terlebih dahulu untuk nilai yang digunakan berkisar dari 1 hingga 20 dengan kelipatan 5, angka minimal 1 dipilih karena saat nilai P di bawah 1 maka respon sistem sangat lambat hingga tidak ada respon dan nilai di atas 20 menyebabkan sistem tidak stabil. Hasil grafik pada Gambar 4, 5, dan 6 digunakan sebagai acuan respon putaran motor yang dibaca oleh *encoder* sehingga hasil yang ditampilkan hanya membandingkan dari kurva yang dihasilkan. Karena nilai pada sumbu x menampilkan durasi putaran motor dan nilai pada sumbu y menampilkan jumlah putaran motor.



Gambar 4. Respon *Plotter* terhadap Nilai  $K_p$

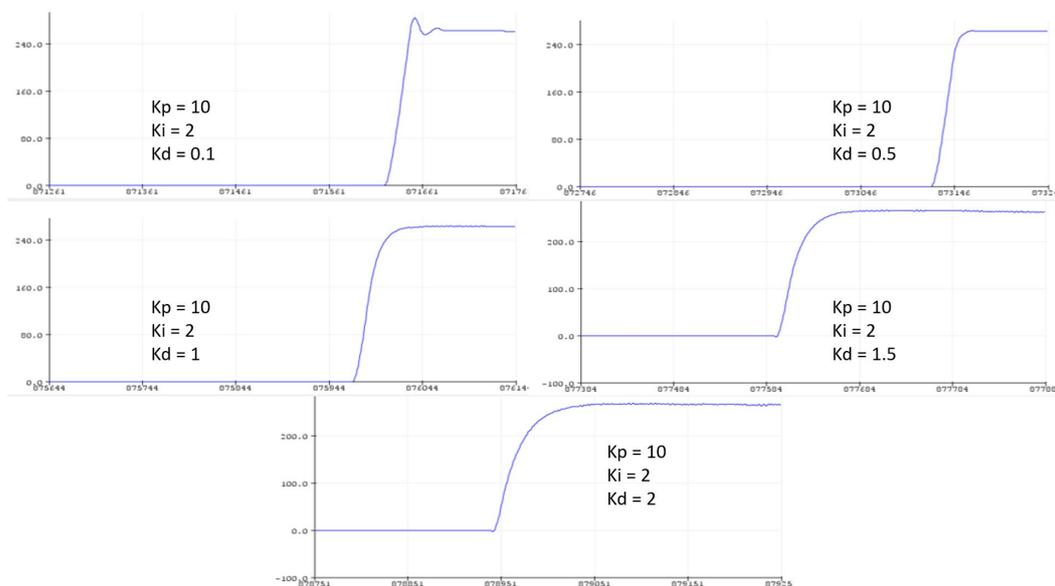
Dari Gambar 4 terlihat bahwa semakin nilai  $K_p$  dinaikan maka akan terjadi *overshoot* dan osilasi pada putaran motor dengan kondisi nilai I dan D tidak dimasukan, namun saat nilai  $k_p$  terlalu kecil maka *Rise time* atau waktu yang dibutuhkan motor untuk mencapai nilai yang diatur akan lambat. Dalam percobaan ini akan digunakan nilai  $k_p$  sebesar 10 karena memberikan *settling time* yang cepat dan untuk *overshoot* dapat diminimalisir oleh tambahan nilai  $K_i$  dan  $K_d$ .

Untuk nilai  $K_i$  akan digunakan nilai antara 0,1 hingga 2 dengan kelipatan 0,5. angka minimal 0,1 dipilih karena saat nilai I di bawah 0,1 maka tidak ada perubahan dalam sistem dan nilai di atas 2 menyebabkan sistem mengalami *overshoot* dan osilasi. Nilai  $K_i$  berfungsi untuk menurunkan nilai *rise time* namun juga dapat meningkatkan *overshoot* dan *settling time*, namun dapat menghilangkan *steady-state error* sehingga hasil akhir putaran motor minim dengan penyimpangan terhadap nilai potensiometer. Perubahan nilai  $K_i$  dapat dilihat pada Gambar 5.



Gambar 5. Respon plotter terhadap nilai Ki

Dari hasil pengujian pada Gambar 5 digunakan nilai 2 sebagai nilai Ki karena menghasilkan *Steady-state error* yang rendah selain peningkatan *overshoot* dan *settling time* yang masih dapat dikoreksi oleh penambahan nilai Kd, sehingga pada percobaan berikut digunakan nilai Kp sebesar 10, nilai Ki sebesar 2, dan untuk nilai Kd yang akan diuji berkisar antara 0.1 hingga 2 dengan kelipatan 0.5. angka minimal 0.1 dipilih karena saat nilai D di bawah 0.1 maka tidak ada perubahan dalam sistem dan nilai di atas 2 menyebabkan sistem tidak stabil. Nilai Kd berfungsi untuk menurunkan *overshoot* dan *settling time*. Perubahan nilai Kd dapat dilihat pada Gambar 6.



Gambar 6. Respon plotter terhadap nilai Kd

Dari hasil pengujian pada Gambar 6 terlihat bahwa jika nilai 0.1 maka tetap terjadi *overshoot* dan osilasi walaupun *rise time* sangat cepat, dan jika nilai 2 maka *overshoot* tidak terjadi sama sekali namun waktu *rise time* akan meningkat dan terjadi ketidakstabilan pada putaran motor yang menyebabkan nilai akhir sulit tercapai. Sehingga untuk nilai Kd ditetapkan nilai 1 sebagai pilihan terbaik.

Berdasarkan pengujian pencarian nilai  $K_p$ ,  $K_i$ , dan  $K_d$ . Kinerja sistem sudah cukup gegas dan tepat, namun terkadang masih bisa terdapat ketidakstabilan pada sistem. Tentu hal ini masih dapat disempurnakan melalui perhitungan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang lebih tepat dengan mengecilkan kelipatan pada perhitungan  $K_i$  dan  $K_d$ . Setelah memasukan parameter-parameter kedalam sistem, maka dapat dirangkum pada Tabel 1.

Tabel 1. Parameter respon sistem pada kontroler *PID*

Parameter dimulai dari Nilai 0	<i>Rise Time</i>	<i>Overshoot</i>	<i>Steady-State Error</i>	<i>Settling Time</i>
$K_p$ Bertambah	Turun	Naik	Turun	Sama
$K_i$ Bertambah	Turun	Naik	Menghilangkan	Naik
$K_d$ Bertambah	Sama	Turun	Sama	Turun

Pertambahan nilai dari masing-masing parameter seperti yang ditunjukkan oleh Tabel 1 akan berbeda, dimana saat mengubah nilai  $K_p$  sedikit saja maka sudah dapat terlihat perbedaan yang signifikan pada *rise time* tetapi nilai *overshoot* juga dapat meningkat dengan tinggi. Selanjutnya terdapat nilai  $K_i$  yang dapat menurunkan nilai *steady state error* walaupun akan berpengaruh pada nilai *settling time*. Untuk nilai  $K_d$  saat ditambahkan maka akan menurunkan potensi *overshoot* dan *settling time* sehingga ketiga parameter *PID* saling bekerja untuk menghasilkan respon yang baik. *Rise Time* merupakan waktu yang dibutuhkan sistem untuk menuju nilai yang telah diatur, *overshoot* merupakan kejadian dimana kurva nilai melewati nilai yang telah diatur, *steady-state error* merupakan penyimpangan antara nilai dari hasil putaran motor dengan nilai yang ditetapkan potensiometer, semakin kecil nilai *error* maka deviasi yang terjadi semakin kecil. *Settling time* merupakan waktu yang dibutuhkan sistem untuk menstabilkan kurva dengan turunnya nilai *settling time* maka nilai *overshoot* dapat diminimalkan juga.

## SIMPULAN

Setelah dilakukan simulasi dan pengujian pada kontrol posisi putaran motor *DC* menggunakan arduino dan *PID* maka disimpulkan. Hasil pembacaan putaran motor oleh *encoder* sudah akurat dan *linear*, Sehingga putaran motor dapat terlaksana sesuai putaran potensiometer dengan begitu jika sistem ini digunakan sebagai pengontrol komponen lain maka dapat bekerja dengan baik. Dari hasil percobaan pada program dihasilkan nilai  $K_p = 10$ ,  $K_i = 2$ , dan  $K_d = 1$ .

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Agus Halim dan Bapak Rosehan sebagai Dosen Program Studi Teknik Mesin Universitas Tarumanagara serta berbagai pihak yang telah membantu dan memberikan saran pada penelitian ini.

## DAFTAR PUSTAKA

- [1] V. Singh and V. K. Garg, "Tuning of *PID* controller for speed control of *DC* motor using soft computing techniques - A review," *Int. J. Appl. Eng. Res.*, vol. 9, no. 9 SPEC. ISSUE, pp. 1141–1147, 2014.
- [2] S. F. Anggraini, A. Ma'arif, and R. D. Puriyanto, "Pengendali *PID* pada Motor *DC* dan Tuning Menggunakan Metode Differential Evolution (DE)," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 6, no. 2, pp. 147–159, 2020, doi: 10.15575/telka.v6n2.147-159.
- [3] Iswanto, A. Ma'arif, R. D. Puriyanto, N. M. Raharja, and S. N. Rahmadhia, "Arduino Embedded Control System of *DC* Motor Using Proportional Integral Derivative," *Int. J. Control Autom.*, vol. 13, no. 4, pp. 658–667, 2020.
- [4] S. Khan *et al.*, "Design and Implementation of Potentiometer-Based Nonlinear Transducer Emulator." Kuala Lumpur, Malaysia, p. 11, 2011.
- [5] Y. C. Chen, C. H. Lee, M. J. Chou, and S. C. Shen, "High-Precision Digital Rotary Encoder Based on Dot-Matrix Gratings," *IEEE Photonics J.*, vol. 10, no. 2, pp. 1–12, 2018, doi:

10.1109/JPHOT.2017.2781465.

- [6] L. Louis, “Working Principle of Arduino and Using it as a Tool for Study and Research,” *Int. J. Control. Autom. Commun. Syst.*, vol. 1, no. 2, pp. 21–29, 2016, doi: 10.5121/ijcacs.2016.1203.
- [7] “Sketsa Perancangan,” 2022.  
<https://www.tinkercad.com/things/1hYVVNKB7dR?sharecode=mDbVip7B8wNrMbkChZVtiNV14Dq7bufVRS-cn7PSGm4> (accessed Jun. 18, 2022).