

Pengendalian RPM pada Motor DC Encoder dengan PID Controller berbasis Simulink

Tsany Ammar Rasyid¹, Sheisyia Rhieyanetta Divanny², Kevin Raffie Saputra³,
Muhammad Rizqi Dwi Arifuddin⁴, Hasna Annisa Hazmi⁵, Ardy Seto Priambodo⁶
Program studi Teknik Elektronika, Universitas Negeri Yogyakarta
Kulon Progo, Daerah Istimewa Yogyakarta 55652
Email: ¹tsanyammar.2022@student.uny.ac.id

Abstract- This research focuses on the development and implementation of a DC motor speed control system using the PID (Proportional Integral Derivative) approach in the C++ programming language. DC motors have a crucial role in various applications in the engineering and automatic fields. Accurate speed regulation on DC motors is the main key to achieving the desired results in various applications. In this research, we integrated the use of an encoder sensor to measure revolutions per minute (RPM) of a DC motor and implemented a PID algorithm using the C++ programming language to regulate the motor speed. The application of the PID method allows accurate and responsive control of desired speed changes, with the aim of achieving certain reference values. Experimental results show that this system is able to provide effective control and produce a consistent response to changes in speed, according to the desired RPM value. This research contributes to a deeper understanding of the application of PID in controlling DC motors and expands knowledge regarding the C++ programming language, which can be the basis for practical applications in various fields that require sophisticated control of DC motors.

Intisari- Penelitian ini fokus pada pengembangan serta implementasi sistem kontrol kecepatan motor DC dengan memanfaatkan pendekatan PID (Proportional Integral Derivative) dalam bahasa pemrograman C++. Motor DC memiliki peran krusial dalam berbagai aplikasi di bidang teknik dan otomatisasi. Pengaturan kecepatan yang akurat pada motor DC menjadi kunci utama untuk mencapai hasil yang diharapkan dalam berbagai aplikasi. Dalam penelitian ini, kami mengintegrasikan penggunaan sensor encoder untuk mengukur putaran per menit (RPM) motor DC dan menerapkan algoritma PID menggunakan bahasa pemrograman C++ untuk mengatur kecepatan motor. Penerapan metode PID memungkinkan kendali yang akurat dan responsif terhadap perubahan kecepatan yang diinginkan, dengan tujuan mencapai nilai referensi tertentu. Hasil percobaan menunjukkan bahwa sistem ini mampu memberikan kendali yang efektif dan menghasilkan respons yang konsisten terhadap perubahan kecepatan, sesuai dengan nilai RPM yang ditargetkan. Penelitian ini berkontribusi pada pemahaman yang lebih dalam tentang penerapan PID dalam mengendalikan motor DC dan memperluas pengetahuan terkait bahasa pemrograman C++, yang dapat menjadi dasar untuk aplikasi praktis dalam berbagai bidang yang memerlukan pengendalian yang canggih pada motor DC.

kata kunci- Motor DC, Bahasa C++, Encoder, RPM, PID.

I. PENDAHULUAN

Di era industri modern, hanya sedikit aplikasi industri yang tidak menggunakan motor arus searah (DC) [1]. Hal ini disebabkan kemudahan pengendalian, biaya rendah, dan ketahanan motor DC dalam berbagai aplikasi. Motor arus searah (DC) adalah perangkat yang mengubah energi listrik

arus searah menjadi energi mekanik. Motor DC memiliki banyak aplikasi di bidang robotika [2] dan aplikasi industri [3]. Aplikasi industri dimana motor DC biasa digunakan meliputi peralatan mesin, pabrik kertas, industri tekstil, kendaraan penggerak listrik, dan robotika[4]. Motor DC tersebar luas di wilayah kendali manufaktur untuk waktu yang lama, karena banyaknya karakteristik yang layak, reaksi tinggi, kemudahan untuk menjadi karakteristik kendali torsi awal linier, dll. [5]. Spesifikasi utama motor DC adalah kecepatan yang biasa diatur dengan mengubah tegangan yang diberikan[6].

Motor DC dengan encoder adalah jenis motor DC yang dilengkapi dengan perangkat encoder optik atau magnetik yang digunakan untuk mengukur putaran atau posisi motor dengan tingkat presisi yang tinggi. Motor DC encoder merupakan salah satu jenis motor DC yang memiliki encoder sebagai sensor kecepatan umpan balik pada sistem robotik [7]. Motor DC menghasilkan putaran per menit (RPM) dalam arah tertentu (searah jarum jam atau berlawanan arah jarum jam). Prinsip kerja motor DC sendiri adalah arah medan magnet rotor harus selalu berlawanan dengan arah medan magnet stator, sehingga saling tolak menolak dan juga luas kumparan tempatnya berada. Medan listrik dihasilkan, yang menciptakan medan magnet yang mengelilingi kumparan dalam arah tertentu. Kontrol kecepatan motor DC dicapai dengan mengubah nilai tegangan yang diberikan. Ketika diberi beban maka nilai tegangannya berkurang dan kecepatan motor DC juga berkurang. Oleh karena itu diperlukan suatu pengontrol yang dapat mencapai nilai kecepatan yang stabil meskipun beban pada motor DC berubah[8].

Salah satu pengendalian pada motor DC dapat menggunakan kendali PID, kendali ini telah memainkan peran penting dalam bidang kendali industri karena strukturnya yang sederhana, stabilitas tinggi, dan daya tahan yang tinggi [9]. Dalam berbagai aplikasi industri, terutama pada penggunaan motor DC, penyesuaian parameter dalam pengendalian PID sering diperlukan. Namun, dalam pengendalian PID konvensional, proses penyetelan parameter cenderung kompleks dan memerlukan waktu yang signifikan [10], dan akurasi kendali sering tidak mencapai tingkat yang diharapkan. Penggunaan pengendali PID memiliki peran yang signifikan ketika diterapkan pada motor arus searah (DC), di mana masalah utamanya adalah mengendalikan dan menjaga kecepatan sudut motor pada tingkat referensi tertentu. Kecepatan putaran motor DC dapat bervariasi tergantung pada faktor beban dan seringkali tidak stabil. Penyebab variasi ini

sering kali dapat ditelusuri pada nilai tegangan masukan [11]. PID mempunyai parameter kendali proporsional (P), parameter kendali integral (I), dan parameter kontrol Derivative (D) [12]. Ketiga nilai tersebut menjadi parameter utama saat menggunakan pengontrol PID, yang masing-masing mempunyai kelebihan dan kekurangan. Dengan begitu, nilai ketiga parameter ini dapat diubah sesuai kebutuhan program untuk menghasilkan keluaran respons yang responsif dan akurat. Adapun kelebihan masing-masing aksi kontrol ini yaitu aksi kendali proporsional mempunyai keunggulan rise time yang cepat dan mengurangi error keadaan stabil, aksi kontrol integral mempunyai kelebihan yaitu mengurangi error keadaan stabil dan aksi kontrol derivative mempunyai keunggulan untuk meredam overshoot.

Dengan kelebihan yang terdapat pada sistem kontrol PID dan pengaruhnya terhadap kendali Motor DC, dimana PID merupakan salah satu jenis kontrol umpan balik yang dapat digunakan untuk mengatur posisi, kecepatan, maupun arah rotasi Motor DC. Pada penelitian ini, telah dilakukan penelitian serta uji coba untuk mengendalikan RPM pada Motor DC dengan judul **“Pengendalian RPM pada Motor DC Encoder dengan PID Controller berbasis Simulink”** yang dirancang melalui simulasi menggunakan software Tinkercad. Tujuan dari penelitian ini diharapkan mampu memberikan kontribusi dalam meningkatkan kinerja motor DC menggunakan kendali PID guna mempertahankan kecepatan dan mengurangi penyimpangan dari setpoint yang diinginkan, sehingga motor DC dapat digunakan dalam berbagai aplikasi yang memerlukan pengendalian kecepatan yang stabil dan akurat.

II. METODE PENELITIAN

Penelitian ini mengusulkan penggunaan PID sebagai kontroler terhadap motor DC. PID sendiri memiliki tiga parameter pengendali utama, yaitu proporsional (P), integral (I), dan derivatif (D). Saat menggunakan pengontrol PID, tiga nilai ini menjadi aspek kunci yang perlu dipertimbangkan. Setiap parameter memiliki keunggulan dan kelemahan sendiri. Oleh karena itu, diperlukan mengatur nilai-nilai parameter ini sesuai dengan persyaratan program untuk mencapai respons yang responsif dan akurat pada keluaran sistem. Adapun beberapa komponen yang digunakan sebagai berikut:

1. Arduino Uno

Arduino merupakan sebuah perangkat elektronik yang bersifat *open source* dan sering digunakan untuk merancang dan membuat perangkat elektronik serta software yang mudah untuk digunakan. Arduino UNO termasuk papan mikrokontroler berdasarkan ATmega328 (lembar data). Arduino UNO memiliki 14 pin input/output digital (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, osilator kristal 16MHz, port USB, colokan listrik, header ICSP, dan tombol reset. Arduino UNO mencakup semua yang diperlukan untuk mendukung mikrokontroler. Cukup sambungkan ke komputer dengan kabel USB, nyalakan dengan adaptor AC-DC, atau nyalakan menggunakan baterai[13].

2. DC Motor

DC motor adalah singkatan dari *“Direct Current Motor,”* yang merupakan jenis motor listrik yang mengubah energi listrik searah (DC) menjadi gerakan mekanik. DC motor

bekerja dengan menggunakan medan magnet yang tetap (seperti magnet permanen) dan medan magnet yang berubah (seperti yang dihasilkan oleh arus listrik) untuk menghasilkan gerakan.

3. Driver Motor DC

Perangkat driver motor DC adalah sebuah sistem elektronik penting yang digunakan untuk mengontrol motor DC (*Direct Current*) dengan mengelola arus dan tegangan yang diberikan ke motor. Driver motor DC memberikan kemampuan bagi pengendali untuk mengatur kecepatan motor, mengubah arah putaran motor, dan menjaga motor dari potensi kerusakan akibat arus berlebihan. Driver sering dilengkapi dengan berbagai fitur perlindungan, seperti pelindung terhadap kelebihan panas, serta mendukung kemampuan kontrol yang lebih canggih melalui penggunaan mikroprosesor atau mikrokontroler. Selain itu, driver motor DC sering menggunakan teknik *Pulse Width Modulation* (PWM) untuk mengatur kecepatan motor dengan tingkat presisi yang tinggi. Dengan driver ini, pengendali dapat mencapai kontrol yang akurat atas motor DC, memastikan motor beroperasi sesuai dengan kebutuhan aplikasi dengan tingkat akurasi yang optimal.

4. Potensiometer

Potensiometer termasuk dalam kategori resistor variabel. Fungsi utama potensiometer adalah sebagai pembagi tegangan yang dapat diadaptasi sesuai kebutuhan.

5. Kapasitor

Kapasitor adalah salah satu komponen dalam elektronika yang digunakan untuk menyimpan muatan listrik. Komponen ini terdiri dari dua plat yang terpisah oleh bahan dielektrik. Ketika tegangan diterapkan ke kapasitor, muatan listrik akan disimpan pada plat-plat tersebut.

6. Resistor

Resistor adalah komponen elektronika yang berfungsi untuk menghambat atau membatasi aliran listrik yang mengalir dalam suatu rangkaian elektronika[14]. Resistor dirancang untuk memiliki resistansi yang telah ditentukan sebelumnya, yang diukur dalam satuan ohm (Ω). Resistansi adalah ukuran seberapa sulit aliran listrik melalui resistor. Semakin tinggi resistansinya, semakin besar hambatannya terhadap arus listrik. Fungsi utama dari resistor adalah untuk mengendalikan arus listrik, membagi tegangan, atau menghasilkan panas dalam beberapa aplikasi khusus.

A. Diagram Blok Sistem Tertanam

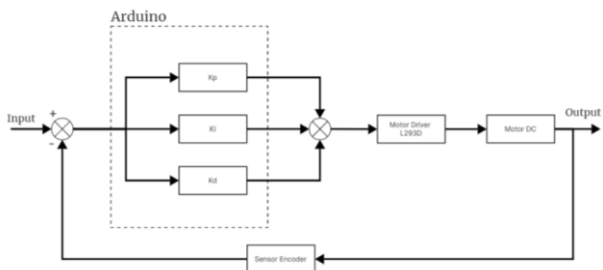


Gambar 1. Diagram Blok Tertanam

Terlihat pada gambar 1 diagram blok tertanam bahwa arduino uno berperan sebagai input, menerima data pulsa yang digunakan untuk mengukur kecepatan putaran motor DC. Arduino Uno menggunakan fitur timer dan counter untuk memproses data kecepatan putaran, sehingga data tersebut dapat ditampilkan pada serial monitor dan serial plotter untuk melihat kurva respons sistem. Untuk mengatur pada tegangan

masuk motor DC digunakan metode *Pulse Width Modulation* (PWM) bersama dengan motor driver L293D untuk mengubah sinyal PMW menjadi sinyal analog yang dapat mengatur kecepatan motor DC sesuai dengan keinginan. Hasil keluaran dari sistem ini adalah motor DC yang dilengkapi dengan sensor encoder yang ter intergritas pada motor DC.

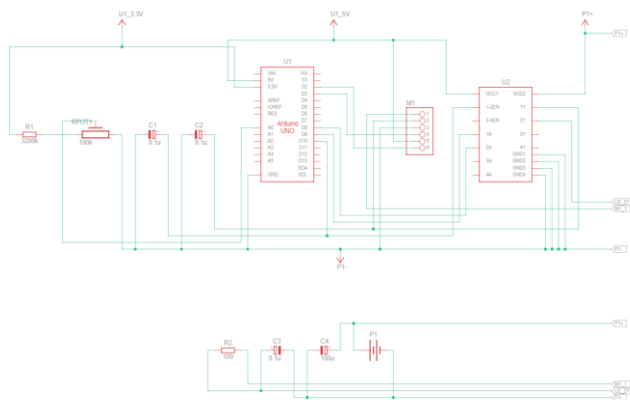
B. Diagram Blok Sistem Kendali



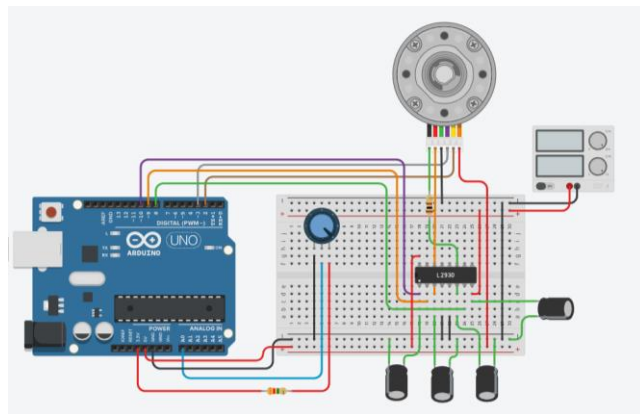
Gambar 2. Diagram Blok Sistem Kendali

Kemudian pada diagram blok untuk sistem kontrol ditampilkan dalam gambar 2. Terlihat bahwa sistem ini merupakan representasi dari sistem kendali dengan umpan balik menggunakan encoder. Set Point adalah nilai referensi yang harus dicapai oleh sistem motor DC. Sensor encoder ini berperan dalam membaca nilai kecepatan motor, yang kemudian dibandingkan dengan nilai referensi (set point) untuk menghasilkan kesalahan (error). Kesalahan ini yang berperan sebagai input bagi kontrol PID, yang bertugas mengendalikan kecepatan motor DC untuk mencapai nilai referensi tersebut. Output dari proses pengendalian PID adalah *Pulse Width Modulation* (PWM). Melalui mikrokontroler Arduino UNO, sinyal PWM dengan rentang nilai antara 0 hingga 255 diteruskan ke driver motor L293D. Ini mengubah nilai PWM dan mengkonversi tegangan digital (PWM) menjadi tegangan analog, yang akhirnya mengatur kecepatan motor DC sesuai dengan keluaran dari kontrol PID. Proses ini terus berlangsung secara kontinu hingga kecepatan motor DC mencapai nilai referensi (Set Point) yang telah ditentukan.

C. Skema Pengkabelan Rangkaian



Gambar 3. Skematik rangkaian Motor DC



Gambar 4. Circuit Motor DC

Tabel 1. Pin Arduino Uno

NOMOR PIN	FUNGSI
8	Arah motor driver
9	Arah motor driver
10	PWM
2	Encoder A
3	Encoder B
A0	Potensio

D. Perhitungan Kecepatan (RPM) dan PID controller

```
//calculate rpm
float currentAngle = getCurrentAngle();
float delta = currentAngle - prvAngle;

if(prvAngle > 270 && currentAngle < 90) {
    delta += 360.0;
}
else if(prvAngle < 90 && currentAngle > 270) {
    delta -= 360.0;
}

currentRPM = delta/360.0/Ts*1000*60;
prvAngle = currentAngle;
```

Gambar 5. Program untuk perhitungan RPM

Program diatas berfungsi untuk mengukur kecepatan motor (RPM). Lalu nilai RPM didapatkan dari perhitungan rumus yang ada pada bagian 'currentRPM'. Variabel 'currentRPM' berisi rumus yang menghitung RPM dengan membagi delta oleh 360.0 (untuk menghitung satu putaran penuh). Kemudian membaginya dengan 'Ts' yang memiliki satuan milidetik untuk menghitung perubahan sudut per milidetik, dan mengalikannya dengan 60 untuk mengkonversinya ke dalam sudut dalam satuan per menit.

```
float target = analogRead(PIN_POTENTIO); // *1100.0/1028.0;

//pid program
error = target - currentRPM;
float derivatif = error - erroterakhir;
integral += error;
float out = kp*error + kd*derivatif + ki*integral;
setSpeed(out);
erroterakhir = error;
```

$$PID(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t)$$

Gambar 6. Program PID pada Motor DC

Program di atas membahas PID yang dipakai sama kita dalam penelitian. Pada perhitungan $error = target - currentRPM$ ini di gunakan untuk menghitung selisih antara nilai target kecepatan (target) dan kecepatan saat ini (currentRPM). Ini adalah komponen Proporsional (P) dalam pengendalian PID.

Lalu pada perhitungan $float\ derivatif = error - erroterakhir$ digunakan untuk menghitung turunan kesalahan (derivative) dengan mengurangi kesalahan saat ini dengan kesalahan sebelumnya (erroterakhir). Ini adalah komponen Derivatif (D) dalam pengendalian PID.

Lalu untuk $integral += error$ ini adalah mengakumulasi kesalahan seiring waktu (integral) dengan menambahkan kesalahan saat ini ke integral sebelumnya. Ini adalah komponen Integral (I) dalam pengendalian PID.

Pada perhitungan $float\ out = kp*error + kd*derivatif + ki*integral$ adalah perhitungan kontrol PID utama, di mana tiap komponen (P, I, dan D) dikalikan dengan koefisien yang sesuai (kp, ki, dan kd) dan kemudian dijumlahkan untuk menghasilkan sinyal kontrol akhir (out).

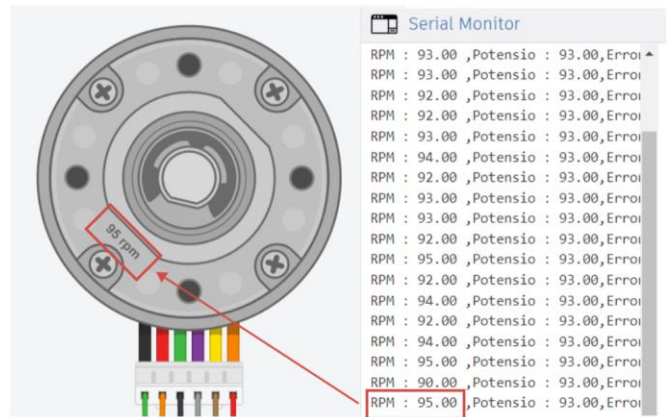
Tabel 2 . Nilai parameter PID

NO	Kp	Ki	Kd
1	1e-4	4e-4	1e-4
2	3e-4	5e-4	1e-4
3	7e-4	9e-4	9e-4

Dalam penelitian ini akan menggunakan 3 nilai dari parameter PID controller yang berbeda untuk menjadi perbandingan antara satu sama lain.

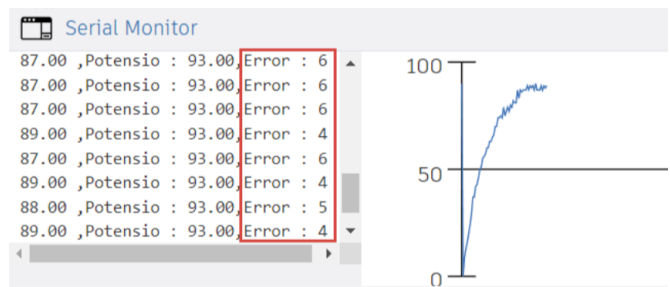
III. PEMBAHASAN

Pada bagian ini adalah penjelasan dari tiga tahap pengujian yang akan dibandingkan antara nilai yang paling baik dari respon yang dihasilkan. Dalam pembahasan ini tiga pengujian akan mengacu pada nilai PID yang ada pada tabel di bab sebelumnya. Dimana maksud dari 1e-4 pada nilai PID adalah 1×10^{-4} , begitupun dengan nilai parameter yang lain. Sebelum masuk pada pengujian, nilai dari RPM pada serial Monitor harus sama dengan nilai *real time* pada rangkaian.



Gambar 7. Nilai RPM pada Motor DC dan serial monitor

Pada gambar diatas sudah dapat membuktikan bahwa nilai dari RPM pada motor dan pada serial monitor sudah sama, hal ini diperlukan agar tidak terjadi perbedaan data yang dihasilkan oleh program dan yang dihasilkan oleh motor DC.



Gambar 8. serial monitor pengujian 1

Pengujian pertama menggunakan nilai PID no 1 pada tabel diatas, dari error yang tertera diatas didapatkan bahwa nilai error yang dihasilkan masih terlalu banyak dan jauh dari nilai set point. Dalam pembuktiannya dalam perhitungan error sebagai berikut:

$$93 - 87 = 6 \quad (1)$$

Persamaan diatas diambil dari data pertama pada gambar diatas. Jika dirata-rata nilai error pada diatas didapatkan :

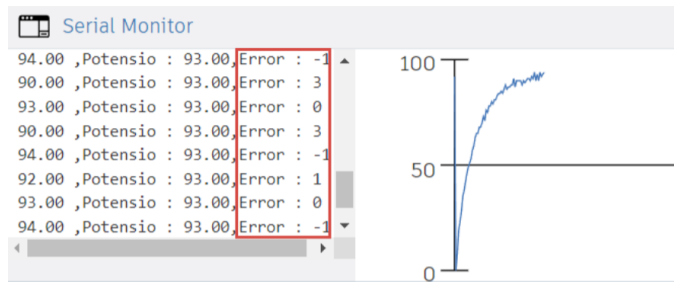
$$\frac{6 + 6 + 5 + 4 + 6 + 4 + 5 + 4}{8} = 4,875 \quad (2)$$

Selanjutnya nilai dari *transient respon* pada pengujian kedua ini sebagai berikut :

Tabel 3. Nilai Transient Response pengujian 1

Rise Time	Overshoot	Error	Settling Time
0,1s	0	4,875	5,44 s

Pada pengujian kedua nilai kan menggunakan nilai parameter PID no 2 pada tabel. Nilai dari PID dari



Gambar 9. serial monitor pengujian 2

Pada pengujian kedua nilai kan menggunakan nilai parameter PID no 2 pada tabel. Nilai dari PID yang digunakan adalah $K_p = 3e-4$, $K_i = 5e-4$, dan $K_d = 1e-4$. Jika melihat dari grafik yang dihasilkan sendiri sudah mengalami sedikit perbaikan pada output respon, dimana nilai dari error sendiri menjadi lebih sedikit. Pembuktian dari nilai error pada gambar diatas didapatkan dari persamaan berikut :

$$93 - 94 = -1 \tag{3}$$

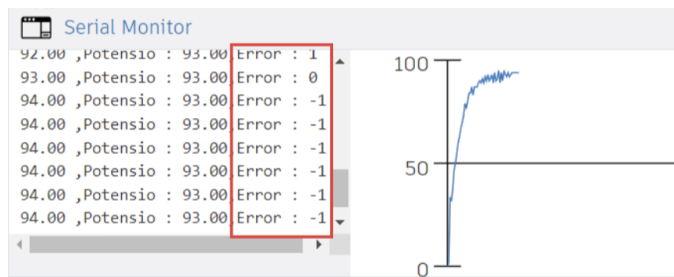
Nilai dari persamaan diatas didapatkan dari data pertama pada gambar. Untuk mengetahui rata-rata error yang ada pada pengujian kedua nilainya akan dimasukkan pada persamaan berikut :

$$\frac{1 + 3 + 0 + 3 + 1 + 1 + 0 + 1}{8} = 1,25 \tag{4}$$

Pada persamaan ini semua nilai akan dimultak agar tidak ada yang bernilai minus (-). Nilai dari transient respon pada pengujian kedua ini sebagai berikut :

Tabel 3. Nilai Transient Response pengujian 2

Rise TIme	Overshoot	Error	Settling Time
0,1s	1,07%	1,25	3,3s



Gambar 8. serial monitor pengujian 3

Pada pengujian ketiga, nilai PID yang digunakan adalah PID no 3 pada tabel. Jika dilihat dari nilai error yang dihasilkan, nilainya cenderung kecil dan tetap yang membuat output respon menjadi lebih stabil. Untuk mengetahui rata-rata error pada data diatas maka digunakannya persamaan :

$$\frac{1 + 0 + 1 + 1 + 1 + 1 + 1 + 1}{8} = 0,875 \tag{5}$$

Sama seperti pengujian kedua, nilai dari error juga akan dimultakkan untuk menghilangkan data yang bernilai minus(-

). Selanjutnya nilai dari *Transient Response* yang dihasilkan oleh data di atas sebagai berikut :

Tabel 3. Nilai Transient Response pengujian 3

Rise TIme	Overshoot	Error	Settling Time
0s	1,07%	0,875	2,54s

IV. KESIMPULAN

Respon dari sistem yang baik adalah yang memiliki karakteristik nilai dari *Transient Response* yang kecil atau bernilai 0, yang dimaksud dari *Transient Response* sendiri adalah nilai dari *Overshoot*, *Rise time*, *Steady State Error*, dan *Settling time*. Nilai PID yang digunakan akan memperbaiki hasil respon sistem. Dengan mengganti-ganti nilai dari 3 parameter PID *controller* akan memperlihatkan perbedaan antara respon PID dengan nilai yang berbeda dalam memperbaiki respon keluaran.

Pada 3 pengujian yang sudah dilakukan output respon yang dihasilkan sudah cukup baik dan responsif dengan sedikit nilai error dan *overshoot*. Namun respon yang dihasilkan masih berubah-ubah dan tidak bisa stabil pada *Steady State Value* atau set point yang telah ditetapkan.

UCAPAN TERIMAKASIH

Terima kasih kepada Universitas Negeri Yogyakarta dan segenap tim yang membantu dalam penelitian ini, sehingga dapat terselesaikan. Terima kasih pula kepada Tim dari JTE UNIBA yang telah meluangkan waktunya untuk melakukan proses paper ini untuk bisa dipublikasikan.

REFERENSI

- [1] W. Cui, Y. Gong, and M. H. Xu, "A permanent magnet brushless DC motor with bifilar winding for automotive engine cooling application," *Jurnal Trans. Magn*, Vol. 48, No. 11, pp. 3348-3351, 2012
- [2] Kuncoro, W. Mulyo Utomo, S. Winardi, and K. Eko Susilo, "Perancangan Kontroller Proportional Integral Derivative Robot Segway Berbasis Mikrokontroler Arduino Nano," Jun. 2019
- [3] A. S. Arifin and R. D. Puriyanto, "Rancang Bangun Pemberian Pakan Ayam Petelur Otomatis Menggunakan PLC," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 1, p. 19, 2019.
- [4] M. M. Sabir and J. A. Khan, "Optimal design of PID controller for the speed control of DC motor by using metaheuristic techniques," *Advances in Artificial Neural Systems*, pp 1 - 8, 2014.
- [5] W. M. Elsgrogy, M. A. Fkirin, and M. A. M. A Hassan, "Speed Control of DC Motor Using PID Controller Based on Artificial Intelligence Techniques," *IEEE Industrial Electronic and Control Department, Menofia University, Egypt*, Vol. 5, pp 232-238, Oct. 2013

- [6] Ph. Q. Dzung, and Le. M. Phuong, "ANN - Control System DC Motor," Faculty of Electrical and Electronic Engineering HCMC University of Technology Ho Chi Minh City-Vietnam.2002
- [7] Hudati, A. P. Aji, dan S. Nurrahma, "Kendali Posisi Motor DC dengan Menggunakan Kendali PID," Jurnal Listrik, Instrumentasi, dan Elektronika Terapan, Vol. 2, No. 2, pp. 9-16, Oktober 2021
- [8] Alfian ma'arif, Ryan istiarno, Sunardi, "Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning", Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, Vol. 9, pp. 374 - 388, April 2021
- [9] W. Zhang, "Increment PID controller based on immunity particle swarm optimization algorithm", Jurnal Microcomputer Information, Vol. 28, No. 7, pp. 67-69, 2013.
- [10] B. Liu, J. Tan, and C. Huang, "Research and Application of an Improved PID Control Algorithm", Jurnal Microcomputer Information, Vol. 6, No. 1, pp. 15-17, 2007.
- [11] Widagdo Purbowaskito, Chung-Hao Hsu, "Sistem Kendali PID untuk Pengendalian Kecepatan Motor Penggerak Unmanned Ground Vehicle untuk Aplikasi Industri Pertanian," Jurnal Infotel, vol. 9, no. 4, p. 376 - 281, November 2017
- [12] Muhammad Reza Aditya Nurkholis Putera, Rahmat Hidayat, "Kendali Kecepatan Motor DC Menggunakan Pengendali PID dengan Encoder sebagai Feedback", Jurnal STRING, Vol. 7 No. 1, pp 50 - 56, Agustus 2022
- [13] Asriningati, Esa Apriaskar, Djuniadi, "Prototipe Sistem Kendali Otomatis Pada Smart Home Dengan Arduino Uno Bertenaga Surya", Jurnal Teknik Elektro UNIBA, Vol. 04, No. 02, April 2020
- [14] Saftiadi, A. Asni B, Aswadul Fitri Saiful Rahman, "Perancangan Sistem Kontroler Alat Elektronik Rumah Tangga Berbasis Mikrokontroler Arduino Uno Dengan Perintah Suara", Jurnal Teknik Elektro UNIBA, Vol. 3. No. 2, April 2019