

Sistem Kendali Proportional Integral Derivative (PID) Menggunakan Mikrokontroler Arduino Pada Thinkercad

Ririn Solekha¹, Ulinnuha Latifa²

^{1,2}Jurusan Teknik Elektro Fakultas Teknik Universitas Singaperbangsa Karawang; email: 2010631160026@student.unsika.ac.id

[Dikirimkan: 10 Januari 2024, Direvisi: 16 Mei 2024, Diterima: 27 Mei 2024]

Corresponding Author: Ririn Solekha

INTISARI — Dalam sebuah percobaan simulasi sistem kendali yaitu dengan melakukan percobaan pengendalian posisi Motor DC menggunakan PID (Proportional Integral Derivative) pada website Thinkercad dimana dilakukan simulasi menggunakan komponen arduino uno R3 sebagai mikrokontroler sistemnya (otak sistem) kemudian Motor Driver (IC L293D) untuk menggerakkan motor DC dan Motor DC with Encoder memberikan umpan balik posisi untuk memungkinkan pengendalian yang akurat. Selain itu, untuk mendapatkan respon sistem yang baik dalam percobaan ini maka diperlukan pemilihan parameter nilai Kp (Konstanta Proportional), Ki (Konstanta Integral) dan Kd (Konstanta Derivative) yang tepat karena nilai Kp dapat mengontrol respon transient yang dihasilkan sistem, kemudian nilai Ki dapat mengontrol sejauh mana sistem menanggapi kesalahan steady state dan nilai Kd dapat mengontrol sejauh mana sistem merespons perubahan cepat dalam variabel masukan yang juga dapat mengurangi osilasi. Selanjutnya serial monitor yang dihasilkan dari run simulation pada matlab diambil 200 sample untuk kemudian di export ke matlab untuk mengetahui nilai rise time dan *settling* dari nilai Kp, Ki, Kd yang telah disimulasikan. Adapun percobaan ini dikatakan sebagai sistem embedded dikarenakan telah melakukan percobaan yang mencakup penggunaan perangkat keras dan perangkat lunak yaitu kombinasi perangkat keras (mikrokontroler, motor driver, dan motor DC dengan encoder) dan perangkat lunak (algoritma PID) yang terintegrasi dalam suatu sistem.

KATA KUNCI — Arduino, PID, Sistem Embedded, Thinkercad

I. PENDAHULUAN

Dalam era perkembangan teknologi yang dinamis, sistem kendali otomatis telah menjadi elemen tak terpisahkan dalam beragam bidang, seperti industri manufaktur dan teknologi kendaraan canggih. Salah satu pendekatan paling terkenal dan efektif dalam mengontrol respon suatu sistem terhadap perubahan input adalah sistem kendali PID (Proporsional Integral Derivative). Sistem kendali PID merupakan salah satu metode yang paling umum digunakan dalam bidang kontrol otomatis. Metode PID memiliki kemampuan untuk menjaga sistem pada setpoint yang diinginkan dengan mengontrol variabel keluaran berdasarkan perbedaan antara setpoint dan variabel masukan [1].

Penggunaan sistem kendali PID tidak terbatas pada industri semata, namun juga merambah ke berbagai sektor kehidupan sehari-hari, seperti sistem pemanas, kendaraan otomatis, dan bahkan aplikasi dalam bidang kedokteran [2]. Penggunaan teknologi mikrokontroler, seperti Arduino, menjadi solusi yang terjangkau dan mudah diakses bagi para pengembang untuk mempelajari dan menerapkan kontrol otomatis. Namun, dalam pengaplikasiannya, pengaturan parameter PID yang tepat dan pemahaman mendalam mengenai respons sistem terhadap parameter-parameter ini merupakan tantangan tersendiri [3]. Oleh karena itu, penelitian ini akan memberikan kontribusi dalam pemahaman lebih lanjut tentang penyyetelan parameter PID dan evaluasi kinerjanya dalam simulasi.

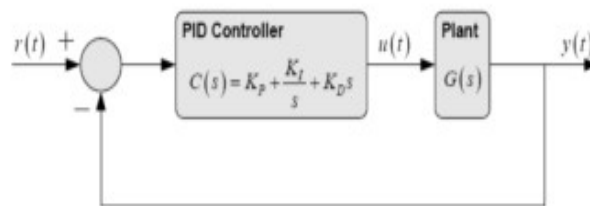
Penelitian ini bertujuan untuk mengeksplorasi penerapan sistem kendali PID dengan menggunakan mikrokontroler Arduino dalam lingkungan simulasi Tinkercad. Arduino telah menjadi platform yang diminati dalam pengembangan prototipe sistem elektronik dan kendali otomatis. Penggunaan lingkungan simulasi Tinkercad memungkinkan peneliti untuk mempelajari, merancang dan menguji kinerja sistem kendali PID tanpa memerlukan perangkat keras fisik, memfasilitasi eksperimen yang efisien dan efektif [4][5].

Penelitian ini akan menelaah konsep dasar dari kontrol PID dan menerapkannya dalam lingkungan simulasi Tinkercad dengan menggunakan Arduino sebagai kontroler. Fokus analisis akan ditujukan pada respons sistem terhadap perubahan input, stabilitas sistem, serta evaluasi kinerja implementasi kontroler PID dalam berbagai skenario simulasi. Diharapkan melalui penelitian ini, akan terbentuk pemahaman yang lebih mendalam mengenai penerapan sistem kendali PID dengan menggunakan mikrokontroler Arduino dalam lingkungan simulasi, yang dapat menjadi dasar bagi pengembangan teknologi yang lebih lanjut dalam aplikasi di dunia nyata.

II. STUDI PUSTAKA

A. KONTROL PID DALAM PEMROGRAMAN

Sistem kendali PID pada rangkaian close-loop secara sederhana diperlihatkan pada gambar 1 yang mana pada rangkaian tersusun sebuah plant dengan kontrol PID yang melakukan kalkulasi kepada error luaran system. Respon yang diinginkan harus memiliki settling time minimum dengan sedikit atau tanpa overshoot dari respon step sistem loop [6].



Gambar 1. Rangkaian system close-loop sederhana dengan kontroler PID

PID Controller sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu P (Proportional) Controller, D (Derivative) Controller, dan I (Integral) Controller [7]. Masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang disebut sebagai konstanta. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel di bawah ini:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \tag{1}$$

Dalam implementasinya di dunia program, persamaan matematis pada tiap komponen control PID perlu diubah kedalam Bahasa yang dimengerti oleh sistem. Sistem computer berjalan secara diskrit, maka dari itu perlu dilakukan konfigurasi pada metode control PID melalui pendekatan matematis seperti pada Tabel I.

TABEL I

PERBANDINGAN PERSAMAAN MATEMATIS DENGAN PERSAMAAN PROGRAM PID

Metode/kontroler	Proporsional	Integrator	Derivative
Matematis	$K_p \times e$	$K_i \int e dt$	$K_d \frac{de}{dt}$
Program	$K_p \times e$	$K_i \times \left(\sum e \times dt \right)$	$K_d \times \frac{e' - e}{dt}$

B. MIKROKONTROLER ARDUINO UNO

Arduino Uno adalah papan mikrokontroler berbasis ATmega328. Uno memiliki 14 pin I/O digital (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, resonator keramik 16MHz, konektor USB, colokan listrik, header ICSP, dan tombol reset. Uno dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan power USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai. Arduino Uno berbeda dari semua papan sebelumnya karena tidak menggunakan chip driver FTDI USBto serial. Di sisi lain, Atmega16U2 (Atmega8U2 hingga versi R2) diprogram sebagai konverter USB ke serial. Uno revisi 2 memiliki resistensi yang menarik HWB 8U2 ke ground, membuat mode DFU lebih mudah digunakan [8].



Gambar 2. Arduino uno

Namun, tidak semua pin Arduino dapat digunakan untuk memproses sinyal analog. Board Arduino Uno memiliki 6 pin analog dari A0 sampai A5. Huruf A di awal nama pin Arduino menunjukkan bahwa pin tersebut dapat digunakan untuk memproses sinyal analog. Pemetaan digital yang tepat dari nilai sinyal analog tergantung pada resolusi ADC. Semakin tinggi resolusi ADC, semakin dekat dengan nilai analog sinyal. Resolusi ADC board Arduino Uno adalah 10 bit. Ini berarti dapat mewakili hingga 1024 level analog diskrit. Beberapa jenis mikrokontroler lain memiliki resolusi 8-bit, 256 level analog diskrit, dan beberapa memiliki resolusi 16-bit dan 65536 level analog diskrit.

C. MOTOR DC

Motor DC memiliki dua bagian penting. Bagian pertama adalah stator yang tidak berputar dan bagian kedua adalah rotor yang berputar. Poros ditempatkan pada rotor ini. Benda yang berputar (seperti puli sabuk) dapat dipasang pada ujung poros.



Gambar 3. Motor DC

D. PULSE WIDTH MODULATION (PWM)

Pulse Width Modulation atau yang biasa disingkat PWM merupakan bentuk modulasi sinyal yang mengatur lebar durasi sinyal pada satu sisinya. PWM digunakan pada tegangan DC untuk mengatur durasi pada tegangan aktif (tegangan puncak) dan tegangan non-aktif (tegangan nol) yang perbandingan tegangan aktif dan non-aktifnya pada suatu periode disebut sebagai Duty Cycle [9]. Dalam mikrokontroler, sinyal PWM beroperasi pada frekuensi rendah dari 500 Hz hingga 2 KHz. Pada board Arduino, pin yang dapat digunakan untuk PWM adalah pin yang ditandai dengan tilde (~): pin 3, 5, 6, 9, 10, dan 11. Pin ini adalah pin yang digunakan untuk membangkitkan sinyal PWM.



Gambar 4. Sinyal PWM

E. THINKERCAD

TinkerCAD adalah layanan online gratis untuk membuat bentuk 3D dasar dan mengembangkan prototipe digital komponen elektronik. Prototipe ini termasuk sirkuit dasar dengan lampu LED, buzzer, sakelar, dan bahkan sensor cahaya.



Gambar 5. Halaman Dashboard Thinkercad

Prototipe ini dapat menyertakan mikroprosesor sebagai bagian dari desain. Mikroprosesor adalah bentuk komputer paling sederhana yang dapat diprogram. Mereka dapat diprogram untuk memanipulasi komponen elektronik seperti lampu LED dan buzzer. Mikroprosesor dapat diprogram untuk mengumpulkan informasi dari sensor dan menginterpretasikan informasi tersebut. Mereka digunakan di berbagai perangkat di sekitar kita. Mereka ada di microwave, lemari es, mobil, komputer, dan banyak perangkat elektronik lainnya [10].

Proses yang digunakan dalam TinkerCAD sering digunakan untuk pembuatan prototipe cepat. Prototyping adalah proses di mana kita dapat mengembangkan komponen dengan cara yang fleksibel daripada yang dapat dengan cepat diperbarui dan dimodifikasi untuk menguji berbagai opsi saat mengembangkan proyek atau produk. Kami akan menggunakan proses pembuatan prototipe ini untuk mempelajari cara membuat rangkaian elektronik dasar. Tinkercad terintegrasi dengan layanan Media Sosial seperti Facebook. Ini juga Terintegrasi dengan layanan seperti Microsoft dan Google. Siswa dapat menggunakan akun distrik mereka untuk masuk ke Tinkercad jika distrik Anda menggunakan akun direktori aktif dengan Google atau Microsoft. Sebuah

proyek dapat dibuat dan dibuka. Sebuah proyek mencakup berbagai komponen elektronik. Komponen elektronik termasuk LED, tombol, resistor, dan sumber daya. Komponen yang dapat kita gunakan ditampilkan dalam panel di bagian bawah halaman.

III. METODE PENELITIAN

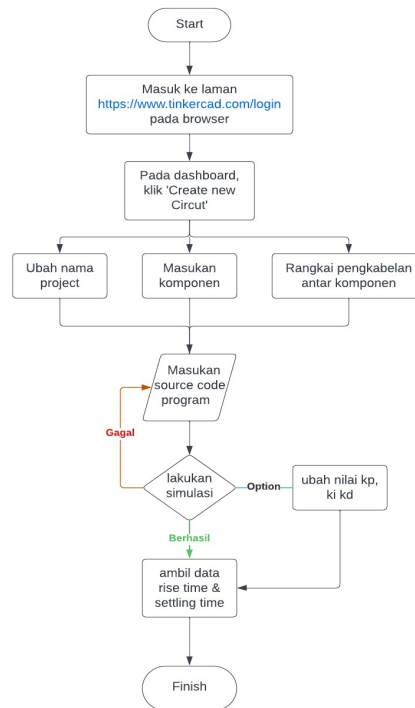
Pada penelitian ini menggunakan metode deskriptif yaitu dengan melakukan pengujian atau simulasi rangkaian menggunakan perangkat lunak atau software. Adapun software yang digunakan yaitu Thinkercad dalam menjalankan simulasi PID yang diintegrasikan dengan arduino uno R3 kemudian data yang didapatkan pada simulasi tersebut digunakan pada software Matlab untuk melakukan percobaan dalam mengetahui nilai *rise time* (T_r) dan *settling time* (T_s) untuk mengetahui kestabilan pada sistem kendali tersebut [11]. Adapun alat dan bahan yang digunakan beserta langkah percobaannya yaitu sebagai berikut.

A. ALAT DAN BAHAN YANG DIGUNAKAN

- Komputer atau laptop
- Jaringan Internet
- Thinkercad dan Matlab
- Arduino Uno R3, Motor Driver IC L293D, dan Motor DC with Encoder
- Buku catatan dan Alat tulis

B. LANGKAH PERCOBAAN

Adapun langkah-langkah dalam percobaan atau simulasi dapat dilihat pada flowchart dibawah ini.



Gambar 6. Flowchart Langkah Percobaan

Gambar 6 merupakan alur dari langkah percobaan pada pengendalian posisi dengan PID menggunakan Thinkercad dimulai dengan menyusun rangkaian dengan menghubungkan komponen satu dengan komponen lain, lalu memasukan source code program untuk dilakukan simulasi. Apabila simulasi berhasil maka ambil data tersebut dengan mengubah nilai k_p , k_i dan k_d yang selanjutnya akan di integrasi dengan matlab untuk mengetahui kestabilan pada rangkaian sistem kendali tersebut.

IV. HASIL DAN PEMBAHASAN

Setelah mengikuti langkah-langkah percobaan maka didapatkan hasil sebagai berikut.

A. SIMULASI PADA THINKERCAD

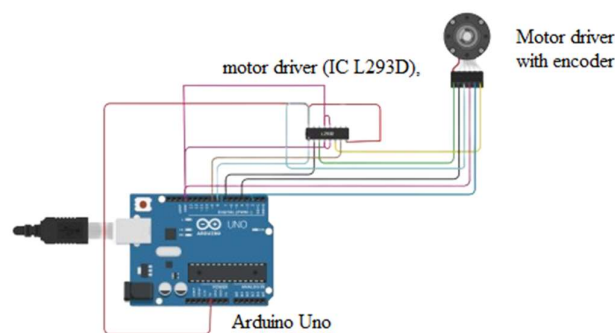
Pada percobaan ini dengan menggunakan Thinkercad melakukan simulasi Pengendalian posisi motor DC dengan PID, yang menggunakan tiga komponen yaitu Arduino Uno R3, Motor Driver (IC L293D) dan Motor DC dengan Encoder. Kemudian merangkai komponen antar satu dengan yang lain sesuai ketentuan pengkabelan atau koneksi antar komponen yaitu sebagai berikut :

TABEL II

KENTENTUAN PENGKABELAN ANTAR KOMPONEN

Motor Driver (IC L293D) > Motor DC dengan Encoder	
Output 1	Motor Positive
Power 1	Encoder Power
Output 2	Motor Negative
Motor Driver (IC L293D) > Arduino Uno R3	
Enable 1 & 2	D8
Input 1	D7
Input 2	D9
Ground	GND
Power 1 & 2	5V
Motor DC dengan Encoder > Arduino Uno R3	
Encoder Ground	GND
Channel A	D2
Channel B	D4

Setelah Mengikuti alur pengkabelan seperti tabel diatas dengan menghubungkan pin pada komponen satu dengan yang lainnya, didapatkan hasil sebagai berikut :



Gambar 7. Hasil Rangkaian pada Thinkercad

Dalam percobaan pengendalian posisi motor DC menggunakan PID pada Thinkercad, digunakan komponen-komponen seperti Arduino, motor driver (IC L293D), dan motor DC dengan encoder karena masing-masing memiliki peran penting dalam implementasi dan pengendalian sistem. Arduino adalah mikrokontroler yang berfungsi sebagai otak atau pusat kendali dalam sistem. Arduino mengambil input dari sensor (encoder) dan menghasilkan output yang dikirim ke motor driver untuk mengendalikan motor DC. Arduino juga menjalankan algoritma PID untuk menghitung sinyal kontrol yang diperlukan [12][13].

Sedangkan Motor driver bertanggung jawab untuk mengontrol arah dan kecepatan motor DC. Motor driver menerima sinyal kontrol dari Arduino dan menggerakkan motor DC sesuai dengan instruksi yang diberikan. sehingga didapatkan implementasi PID untuk menghasilkan sinyal kontrol yang dapat mengatur posisi motor DC dengan akurat. Kemudian Motor DC dengan encoder digunakan untuk memberikan umpan balik posisi motor [14][15]. Encoder adalah sensor yang terpasang pada poros motor dan menghasilkan sinyal yang berkaitan dengan putaran dan pergerakan motor [16][17]. Dengan menggunakan encoder, maka akan mendapatkan informasi real-time tentang posisi aktual motor. Sehingga ini menjadi penting dalam implementasi PID karena memungkinkan sistem untuk membandingkan posisi aktual dengan posisi yang diinginkan (setpoint) dan menghasilkan sinyal kontrol yang sesuai untuk mengurangi kesalahan posisi.

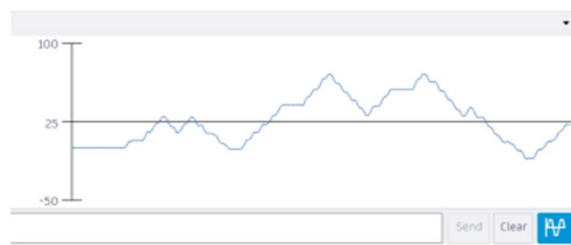
Dengan menggunakan kombinasi Arduino, motor driver (IC L293D), dan motor DC dengan encoder, penulis dapat mengimplementasikan sistem kendali PID yang efektif dalam mengendalikan posisi motor DC. Arduino bertindak sebagai otak sistem, motor driver menggerakkan motor DC, dan encoder memberikan umpan balik posisi untuk memungkinkan pengendalian yang akurat. Kemudian memprogram arduino dengan memasukkan source code program seperti dibawah ini :

```
32 void doEncoderA()
33 {
34   digitalRead(encoderPinB) ? encoderPos-- : encoderPos++;
35 }
36
37 void setup()
38 {
39   //Setup interrupt
40   pinMode(encoderPinA, INPUT_PULLUP);
41   pinMode(encoderPinB, INPUT_PULLUP);
42   attachInterrupt(digitalPinToInterrupt(encoderPinA),
43     doEncoderA, RISING);
44
45   //Setup Motor Driver
46   pinMode(motorDirPin, OUTPUT);
47   pinMode(enablePin, OUTPUT);
48   digitalWrite(enablePin, HIGH);
49   preverr = 0;
50
51   Serial.begin(9600);
52 }
53
54 void loop()
55 {
56   error = targetPos - encoderPos;
57   proporsional = Kp*error;
58   integrator += integrator * dt;
59   derivative = (error+preverr)/dt;
60   control = proporsional + Ki*integrator + Kd*derivative;
61   speed = min(max(control, -255), 255);
```

```
62   if (speed >= 0)
63   {
64     digitalWrite(motorDirPin, CW);
65     analogWrite(motorPWMPin, speed);
66   }
67
68   control = proporsional + Ki*integrator + Kd*derivative;
69
70   speed = min(max(control, -255), 255);
71
72   if (speed >= 0)
73   {
74
75     digitalWrite(motorDirPin, CW);
76
77     analogWrite(motorPWMPin, speed);
78
79   }
80
81   else
82   {
83
84     digitalWrite(motorDirPin, CCW);
85
86     analogWrite(motorPWMPin, 255+speed);
87
88   }
89
90
91
```

Gambar 8. Source code program

Kemudian melakukan *run simulation* untuk melihat hasil sinyal pada serial monitor.



Gambar 9. Grafik Hasil Simulasi

Dikarenakan hasil sinyal tersebut masih dikatakan belum mendapatkan respon sistem terbaik sehingga, penulis melakukan beberapa percobaan dengan mengubah nilai k_p , k_i , k_d pada source code nya untuk memperoleh respon sistem yang menurut penulis terbaik karena Nilai k_p , k_i , dan k_d dalam kendali PID memiliki pengaruh yang signifikan terhadap respons sistem. Perubahan nilai-nilai ini dapat mempengaruhi kinerja dan karakteristik sistem secara keseluruhan. Berikut adalah pengaruh nilai k_p , k_i , dan k_d dalam kendali PID:

1) k_p (GAIN PROPORSIONAL)

- Nilai k_p mengontrol sejauh mana respons sistem terhadap perbedaan antara setpoint dan variabel masukan.
- Semakin besar nilai k_p , semakin cepat sistem akan merespons perbedaan tersebut.
- Namun, jika nilai k_p terlalu besar, sistem dapat menjadi tidak stabil dan mengalami osilasi atau overshoot yang berlebihan.

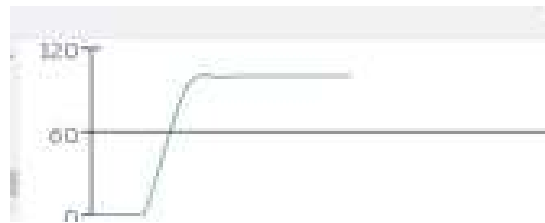
2) KI (GAIN INTEGRAL)

- Nilai Ki mengontrol sejauh mana sistem menanggapi kesalahan steady-state.
- Dengan nilai Ki yang lebih besar, sistem memiliki kemampuan yang lebih baik dalam menghilangkan kesalahan steady-state.
- Namun, jika nilai Ki terlalu besar, sistem dapat menjadi terlalu responsif dan menyebabkan osilasi atau overshoot yang tidak diinginkan.

3) KD (GAIN DERIVATIF)

- Nilai Kd mengontrol sejauh mana sistem merespons perubahan cepat dalam variabel masukan.
- Dengan nilai Kd yang lebih besar, sistem dapat merespons perubahan dengan lebih cepat dan mengurangi osilasi.
- Namun, penggunaan Kd yang terlalu besar dapat menyebabkan sistem menjadi responsif terhadap noise atau gangguan, dan menyebabkan perubahan yang tidak stabil.

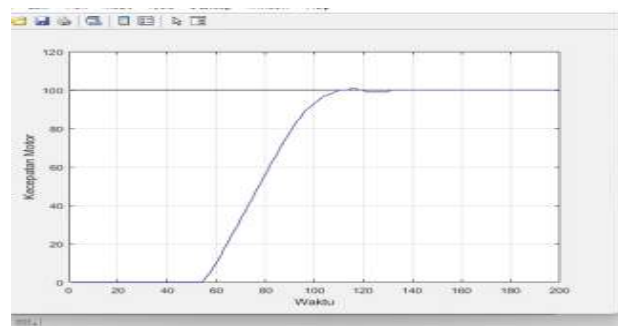
Oleh karena itu penulis memilih parameter nilai $k_p = 5$; $k_i = 0.4$; $k_d = 0,1$ dengan alasan nilai k_p yang besar (5) supaya sistem dapat merespons perbedaan antara setpoint dan variabel masukan dengan cepat, kemudian nilai Ki yang relatif kecil (0.4) dapat memberikan penyesuaian yang cukup untuk mengatasi kesalahan steady-state tanpa risiko terjadinya osilasi yang signifikan dan Nilai Kd yang relatif kecil (0.1) dapat membantu mengurangi osilasi yang berlebihan dan mempertahankan respons yang stabil. Adapun hasil simulasinya dapat dilihat pada gambar 10.



Gambar 10. Hasil Simulasi Pada Thinkercad

B. SIMULASI PADA MATLAB

Kemudian nilai yang terdapat di serial monitor yang telah mencapai 200 sampel terlebih dahulu di *copy-paste* ke excel untuk selanjutnya dirapihkan dan di export ke matlab serta dengan mencantumkan syntax untuk memperoleh nilai rise time dan settling time nya, maka didapatkan hasil sebagai berikut



Gambar 11. Grafik Hasil Export ke Matlab

Sehingga untuk nilai $k_p = 5$; $k_i = 0.4$; $k_d = 0,1$ didapatkan nilai rise time (T_r) = 37 s dan settling time (T_s) = 106 s yang menurut penulis percobaan ini adalah percobaan dengan respon sistem terbaik. Rise Time (T_r) adalah waktu yang dibutuhkan oleh sistem untuk mencapai sekitar 90-95% dari nilai setpoint setelah terjadi perubahan input. Semakin kecil T_r , semakin cepat sistem merespons perubahan dan semakin baik responsnya. Dalam percobaan ini, dengan $T_r = 37$, respons sistem tergolong cepat. Settling Time (T_s) adalah waktu yang dibutuhkan oleh sistem untuk mencapai nilai yang stabil dan berada dalam toleransi tertentu dari setpoint. Semakin kecil T_s , semakin cepat sistem mencapai kestabilan. Dalam percobaan ini, dengan $T_s = 106$ respon sistem dikatakan cepat jika dibandingkan dengan percobaan lainnya yang penulis lakukan.

Selain itu, percobaan pengendalian posisi motor DC menggunakan PID pada Thinkercad, termasuk dalam kategori sistem embedded. Sistem embedded merujuk pada kombinasi perangkat keras (hardware) dan perangkat lunak (software) yang terintegrasi dalam suatu sistem tertentu. Dalam percobaan ini, komponen perangkat keras seperti mikrokontroler Arduino, motor driver (IC L293D), dan motor DC dengan encoder digunakan untuk mengendalikan motor DC. Mikrokontroler Arduino bertindak

sebagai otak atau pusat kendali sistem, sementara motor driver dan motor DC dengan encoder berperan dalam menggerakkan motor dan memberikan umpan balik posisi [18][19].

Dalam pengendalian PID, algoritma PID juga diimplementasikan dalam perangkat lunak pada mikrokontroler Arduino. Algoritma ini mengambil input dari sensor (misalnya, encoder) dan menghasilkan output yang dikirim ke motor driver untuk mengendalikan motor DC. Secara keseluruhan, percobaan pengendalian posisi motor DC menggunakan PID pada Thinkercad memanfaatkan kombinasi perangkat keras (mikrokontroler, motor driver, dan motor DC dengan encoder) dan perangkat lunak (algoritma PID) yang terintegrasi dalam suatu sistem yang dapat dikategorikan sebagai sistem embedded [20].

V. KESIMPULAN

Berdasarkan percobaan yang dilakukan maka didapatkan kesimpulan bahwa Kombinasi Arduino, motor driver (IC L293D), dan motor DC dengan encoder, penulis dapat mengimplementasikan sistem kendali PID yang efektif dalam mengendalikan posisi motor DC. Arduino bertindak sebagai otak sistem, motor driver menggerakkan motor DC, dan encoder memberikan umpan balik posisi untuk memungkinkan pengendalian yang akurat. Kemudian nilai K_p , K_i dan K_d akan mempengaruhi respon sistem yang disimulasikan karena Nilai K_p mengontrol sejauh mana respons sistem terhadap perbedaan antara setpoint dan variabel masukan, Nilai K_i mengontrol sejauh mana sistem menanggapi kesalahan steady-state. Dan Nilai K_d mengontrol sejauh mana sistem merespons perubahan cepat dalam variabel masukan.

Pada percobaan ini penulis memilih nilai $k_p = 5$; $k_i = 0.4$; dan $k_d = 0.1$ yang memperoleh nilai Rise time (T_r) = 37 dan Nilai Settling time (T_s) = 106 yang menurut penulis memperoleh respon sistem terbaik. Serta percobaan pengendalian posisi motor DC menggunakan PID pada Thinkercad memanfaatkan kombinasi perangkat keras (mikrokontroler, motor driver, dan motor DC dengan encoder) dan perangkat lunak (algoritma PID) yang terintegrasi dalam suatu sistem yang dapat dikategorikan sebagai sistem embedded.

KONFLIK KEPENTINGAN

Penulis menyatakan bahwa tidak terdapat konflik kepentingan.

REFERENSI

- [1] I. Setiawan, E. A. Nugroho, N. R. Wibowo, T. R. Mekatronika, and P. E. Indorama, "SISTEM KONTROL KECEPATAN MOTOR UNIVERSAL," vol. 3, no. 1, pp. 22–29, 2023.
- [2] S. Muharom, I. Masfufiah, R. A. Firmansyah, A. Hamid, and S. Utomo, "Implementasi Kontrol Suhu Menggunakan Metode PID pada Aplikasi Inkubator Infant Warmers," vol. 4, pp. 55–59, 2021.
- [3] C. N. Hamdani, A. Azizy, R. Heru, and H. Kurniadi, "RANCANG BANGUN PROTOTYPE SISTEM KONTROL BERTINGKAT MENGGUNAKAN FUZZY-PID BERBASIS ARDUINO," no. 3, pp. 98–105, 2022.
- [4] A. M. A. Arif and R. Istiamo, "Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning," vol. 9, no. 2, pp. 374–388, 2021.
- [5] A. Ma, R. D. Puriyanto, and F. T. Hasan, "Robot Keseimbangan Dengan Kendali Proporsional-Integral-Derivatif (PID) dan Kalman Filter," vol. 4, no. 2, pp. 117–127, 2020.
- [6] P. Baseada, E. M. Arduino, and P. Aprendizado, "ARDUINO BASED PLATFORM FOR PROCESS CONTROL LEARNING," vol. 06, pp. 585–593, 2020, doi: 10.18540/jcecv16iss5pp0585-0593.
- [7] F. Isdaryani, M. Fadhilah, and V. Hesya, "Sintesis Kendali PID Digital dengan Diskritisasi Langsung dan Backward Difference," vol. 9, no. 2, pp. 467–481, 2021.
- [8] D. Sonny and R. Dwi, "Implementation of DC Motor PID Control on Conveyor for Separating Potato Seeds by Weight," vol. 1, no. 1, pp. 15–26, 2021.
- [9] A. Budiarto, W. S. Pambudi, S. Sumari, and A. Yulianto, "PID Control Design for Biofuel Furnace using Arduino," vol. 16, no. 6, pp. 3016–3023, 2018, doi: 10.12928/TELKOMNIKA.v16i6.9770.
- [10] H. M. Asraf, K. A. N. Dalila, A. W. M. Hakim, and R. H. M. F. Hon, "Development of Experimental Simulator via Arduino-based PID Temperature Control System using LabVIEW," vol. 9, no. 1, pp. 53–57, 2020.
- [11] M. D. Ardiansyah, F. Rohman, T. O. Politeknik, and N. Malang, "IMPLEMENTASI DAN ANALISIS KENDALI KECEPATAN MOTOR BLDC 1 kW MENGGUNAKAN ALGORITMA PID," pp. 81–93, 2021.
- [12] T. Pujiati, "Penerapan Kontroler PID pada Sistem Kendali Level Cairan dengan Metode Ziegler-Nichols Berbasis Arduino," vol. 2, no. 1, pp. 55–60, 2021.
- [13] M. Lamatenggo and Z. Acc, "Perancangan Balancing Robot Beroda Dua Dengan Metode Pengendali PID Berbasis Arduino Nano," vol. 2, pp. 39–43, 2020.
- [14] R. F. Anugrah, "Kontrol Motor Brushless DC Menggunakan Six Step Comutation dengan Kontrol PID (Proportional Integral Derivative)," vol. 7, no. 2, 2020.
- [15] W. Mahardika and P. Wiratama, "KOMPARASI KESTABILAN POSISI PANEL SURYA MENGGUNAKAN PENGENDALI PID (PROPORTIONAL, INTEGRAL DAN DERIVATIVE) DENGAN FLC (FUZZY LOGIC CONTROL)," vol. 14, no. 1, pp. 77–88, 2023.
- [16] N. D. Khotimah and M. Rifa, "Desain Sistem Kontrol Kecepatan Motor menggunakan Kendali PID pada Solar E-Bike," vol. 8, no. 9, 2021.
- [17] Y. U. D. Sipul, D. A. Prasetya, and W. Dirgantara, "CONTROL SYSTEM KENDALI KECEPATAN SEPEDA," vol. 1, no. 1, pp. 45–49, 2019.
- [18] D. Fisika and U. Diponegoro, "Rancang bangun sistem kendali pid untuk intensitas cahaya lampu dc menggunakan mikrokontroler arduino uno," vol. 22, no. 1, 2019.
- [19] R. Muhandian, "Kendali Kecepatan Motor DC Dengan Controller PID dan Antarmuka Visual Basic," vol. 06, no. 01, pp. 328–339, 2020.
- [20] M. Pid and B. Plc, "Desain sistem kontrol level cairan dengan metode pid berbasis plc," vol. 6, no. 2, pp. 195–209, 2022.