

## ANALISA PENGARUH PERUBAHAN KONSTANTA PID PADA KENDALI MOTOR DC

Jeffry Gracia Sitohang

Universitas 17 Agustus 1945 Surabaya, Email : sitohangjeffry@gmail.com

Santoso

Universitas 17 Agustus 1945 Surabaya, Email : santoso@untag-sby.ac.id

### Abstract

DC motors often use the PID method for control. The previously mentioned method is praised for its accuracy in regulating the speed of direct current (DC) motors, resulting in consistent motor rotation and sustained speed. However, changing the parameters of a proportional-integral-derivative (PID) controller can affect the operational efficiency and output of a DC motor. Thus, to understand how parameter values affect motor performance and speed in DC motor control, analysis of PID constants is necessary. This study examines how proportional-integral-derivative (PID) parameters affect direct current (DC) motor performance. This multi-stage study included several key components. These components included a comprehensive literature review, development and construction of a DC motor control system, data collection using encoder sensors, application of PID control techniques, data analysis using Python software, and ongoing evaluation of DC motor modifications using PID control methods. The analyses from this study suggest further research. Python can efficiently modify the proportional-integral-derivative (PID) constants. The system works well, meeting the setpoint. Reference points 120, 180, 270, and 360 showed significant deviations. Error regulation and motor angular position matching the setpoint are achieved with proportional-integral-derivative (PID) coefficients in the GM25-CPR DC motor control system. However, determining the proportional-integral-derivative (PID) constant requires trial-and-error, and certain setpoints may still exceed the limit. The Kd value can solve this problem. Empirical studies have shown that a low Kp coefficient is not optimal for error calibration and precise motor degree measurement. In experiments with larger setpoints, the Motor Degree measurement with a higher proportional gain coefficient (Kp) is closer to the target.

Keywords: DC Motor, PID Control, PID Parameters

### Abstrak

Motor DC sering menggunakan metode PID untuk kontrol. Metode yang disebutkan sebelumnya dipuji karena keakuratannya dalam mengatur kecepatan motor arus searah (DC), sehingga menghasilkan putaran motor yang konsisten dan kecepatan yang berkelanjutan. Namun, mengubah parameter pengontrol proporsional-integral-derivatif (PID) dapat memengaruhi efisiensi operasional dan output motor DC. Dengan demikian, untuk memahami bagaimana nilai parameter mempengaruhi kinerja motor dan kecepatan dalam kontrol motor DC, analisis konstanta PID diperlukan. Studi ini meneliti bagaimana parameter proporsional-integral-derivatif (PID) mempengaruhi kinerja motor arus searah (DC). Studi multi-tahap ini mencakup beberapa komponen utama. Komponen-komponen ini termasuk tinjauan literatur yang komprehensif, pengembangan dan pembangunan sistem kontrol motor DC, pengumpulan data menggunakan sensor encoder, penerapan teknik kontrol PID, analisis data menggunakan perangkat lunak Python, dan evaluasi modifikasi motor DC yang sedang berlangsung menggunakan metode kontrol PID. Analisis dari penelitian ini menyarankan penelitian lebih lanjut. Python dapat secara efisien memodifikasi konstanta proporsional-integral-derivatif (PID). Sistem bekerja dengan baik, memenuhi setpoint. Titik referensi 120, 180, 270, dan 360 menunjukkan penyimpangan yang signifikan. Regulasi kesalahan dan posisi sudut motor yang sesuai dengan setpoint dicapai dengan koefisien proporsional-integral-derivatif (PID) dalam sistem kontrol motor DC GM25-CPR. Namun, menentukan konstanta proporsional-integral-derivatif (PID) membutuhkan trial-and-error, dan setpoint tertentu mungkin masih melampaui batas. Nilai Kd dapat mengatasi masalah ini. Studi empiris telah menunjukkan bahwa koefisien Kp yang rendah tidak optimal untuk kalibrasi kesalahan dan pengukuran derajat motor yang tepat. Dalam percobaan dengan setpoint yang lebih besar, pengukuran Derajat Motor dengan koefisien penguatan proporsional yang lebih tinggi (Kp) lebih mendekati target.

Kata kunci : Motor DC, Kontrol PID, Parameter PID

### Pendahuluan

Penggunaan motor DC lazim digunakan dalam pengaturan industri dan non-industri karena kapasitasnya untuk menghasilkan rotasi yang konsisten dengan kecepatan yang dapat dimodifikasi[1]. Namun demikian, perangkat ini memerlukan parameter manipulasi yang cermat dan tepat untuk beroperasi secara efektif.

Kontrol PID (Proportional Integral Derivative) adalah teknik kontrol yang sering digunakan untuk motor DC[2]. Teknik yang disebutkan di atas terkenal dengan ketepatannya dalam mengatur kecepatan motor arus searah, yang mengarah ke putaran motor yang konsisten sambil mempertahankan kecepatan yang diinginkan. Perubahan pada konstanta parameter PID dapat berdampak pada kinerja motor DC dan hasil yang dihasilkannya.

Oleh karena itu, sangat penting untuk memeriksa dampak variasi konstanta PID pada kontrol motor DC untuk memahami pengaruh nilai parameter pada kecepatan dan kinerja motor[3]. Penelitian ini bertujuan untuk menguji korelasi antara parameter PID dan kinerja kontrol motor DC, penelitian ini memberikan analisis yang komprehensif dan logis tentang dampak penyesuaian konstanta PID pada kontrol motor DC. Analisis ini mencakup eksposisi dasar-dasar teoretis motor DC serta pemeriksaan teknik kontrol PID yang digunakan dalam pengoperasiannya[4].

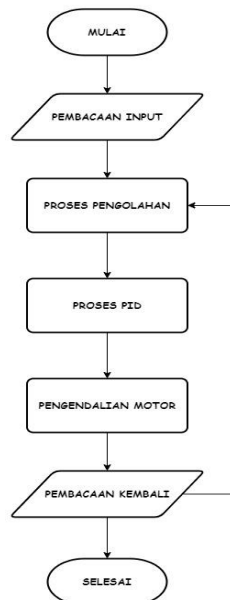
Selain itu, dampak dari berbagai konstanta PID pada kinerja motor DC dieksplorasi[5]. Penelitian ini menganalisis kinerja motor DC di bawah kendali PID dengan konstanta PID yang berbeda. Tujuannya adalah untuk memberikan wawasan bagi mereka yang terlibat dalam penelitian dan pengembangan kontrol industri.

### Metode

Penelitian ini menggunakan metode penelitian multi-tahap, yang meliputi tinjauan literatur, desain dan pembuatan sistem kontrol motor DC, akuisisi data melalui sensor encoder, penerapan teknik kontrol PID, analisis menggunakan perangkat lunak Python, dan menentukan efek modifikasi yang sedang berlangsung pada motor DC melalui penggunaan metode kontrol PID. Temuan dari penelitian ini memberikan rekomendasi untuk penelitian selanjutnya, berdasarkan analisis yang dilakukan.

### Diagram Alir Sistem

Pengaruh konstanta PID pada kontrol motor DC digambarkan dalam diagram alir sistem yang disajikan di bawah ini.



**Gambar 1 Flowchart PID Sistem**

Arduino Uno menggunakan pengontrol PID untuk mengatur kecepatan atau posisi motor DC[6]. Pengontrol PID secara konstan membandingkan kecepatan atau



Algoritme pengkodean yang disajikan di sini memanfaatkan pengontrol PID untuk memandu motor menuju posisi target 120 derajat. Sensor encoder digunakan untuk mengukur rotasi motor, sementara variabel kesalahan menghitung perbedaan antara posisi motor yang diinginkan dan yang diukur. Kontrol PID dihitung dengan menggunakan nilai tertentu untuk Kp, Ki, dan Kd. Motor kemudian diarahkan untuk berputar searah atau berlawanan arah jarum jam berdasarkan output kontrol. Rincian mengenai posisi motor, posisi encoder, dan kesalahan ditampilkan pada Serial Monitor.

### Desain Coding Python

Penjelasan mendalam mengenai algoritme yang mendukung kode grafis real-time berbasis Python tersedia di bawah ini[11].

```

import serial
import matplotlib.pyplot as plt
from datetime import datetime

# Arduino Data = serial.Serial('COM3', 9600) # Adjust the 'COM3' to your Arduino's serial port
plt.ion() # Enable interactive mode for matplotlib

encoder_position = 0
motor_degree = 0
error = 0

def plot_values():
    plt.clf()
    plt.plot(plt.gca().get_lines()[0].get_ydata(), '-or')
    plt.plot(encoder_position, '-or')
    plt.ylabel('Encoder Position')
    plt.xlabel('Time')
    plt.grid(True)
    plt.pause(0.001)
    plt.plot(motor_degree, '-or')
    plt.ylabel('Motor Degree')
    plt.xlabel('Time')
    plt.grid(True)
    plt.pause(0.001)
    plt.plot(error, '-or')
    plt.ylabel('Error')
    plt.xlabel('Time')
    plt.grid(True)
    plt.pause(0.001)

while True:
    if Arduino_data.in_waiting == 3:
        data = Arduino_data.readline().decode('utf-8').strip()
        pos, enc, deg = data.split(',')
        encoder_position.append(float(pos))
        motor_degree.append(float(deg))
        error.append(float(pos) - 120)
        drawnow(plot_values)
    except KeyboardInterrupt:
        break
    except Exception as e:
        print(e)

```

Gambar 4 Desain Coding Python

Algoritma pengkodean ini menghasilkan grafik untuk data yang diperoleh dari motor DC yang dikontrol melalui Arduino menggunakan PID. Algoritma ini bergantung pada komunikasi serial antara Arduino dan komputer, serta pustaka serial, matplotlib, dan drawnow di Python.

Untuk memulai, pustaka yang diperlukan diimpor, diikuti dengan mengatur port serial untuk Arduino. Mode interaktif matplotlib diaktifkan. Beberapa daftar diinisialisasi untuk menyimpan data yang diterima dari Arduino. Sebuah fungsi bernama plot\_values() didefinisikan untuk memplot data yang diterima menggunakan matplotlib.

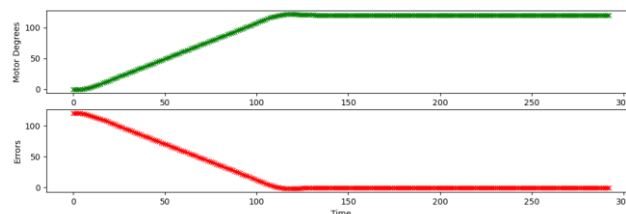
Di dalam perulangan while, data diterima dari Arduino dan diuraikan menjadi tiga variabel. Variabel-variabel ini kemudian ditambahkan ke daftar yang telah diinisialisasi sebelumnya. Fungsi drawnow(plot\_values) dipanggil untuk menghasilkan grafik setiap kali data baru diterima. Perulangan terus dijalankan selama program berjalan, kecuali jika terjadi kesalahan. Jika terjadi kesalahan, sebuah pesan kesalahan akan dicetak.

### Hasil Dan Pembahasan

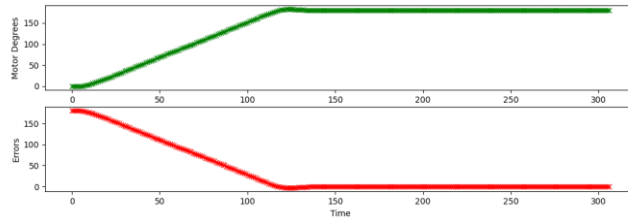
Eksperimen dan pengujian dilakukan dalam bab ini untuk menentukan bagaimana perubahan nilai konstanta PID mempengaruhi sistem kontrol motor DC. Kit Pengembangan Perangkat Lunak (IDE) Arduino dan Python digunakan untuk pengujian.

### Grafik Pengujian PID

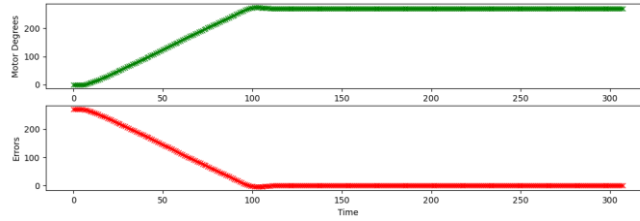
Grafik yang dihasilkan selama pengujian PID akan mengilustrasikan prosedur penyetulan konstanta PID dengan menggunakan metode Trial and Error. Pendekatan ini bertujuan untuk mengoptimalkan kinerja motor dan mencapai setpoint yang diinginkan.



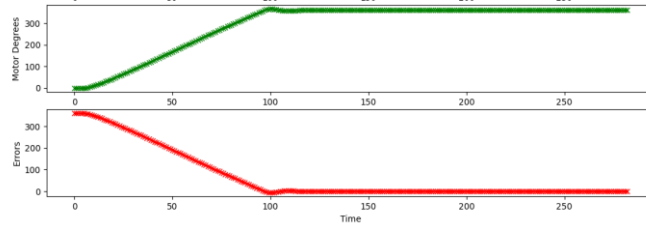
**Gambar 5 Grafik Setpoint 120**



**Gambar 6 Grafik Setpoint 180**



**Gambar 7 Grafik Setpoint 270**



**Gambar 8 Grafik Setpoint 360**

Dengan menggunakan Python, konstanta PID dapat disetel secara efektif. Sistem ini menunjukkan hasil yang baik, selaras dengan setpoint yang telah ditetapkan sebelumnya. Namun, ada overshoot yang nyata pada setpoint 120, 180, 270, dan 360. Untuk mengurangi overshoot ini, disarankan untuk meningkatkan nilai  $K_d$ [12].

### Tabel Pengujian PID

Tabel di bawah ini menyajikan data eksperimen yang dikumpulkan dari berbagai pengujian yang dilakukan untuk mengubah konstanta PID dalam kontrol Motor DC menggunakan perangkat lunak Arduino IDE.

Kp	Ki	Kd	Setpoint	Motor Degrees	Errors
30	0.0001	10	15	14.32	0.68
30	0.0002	10	15	14.27	0.73
30	0.0003	10	15	14.29	0.71
30	0.0001	10	30	29.65	0.35
29	0.0001	10	30	29.66	0.34
28	0.0001	10	30	29.59	0.41
30	0.0001	10	45	45.33	-0.33
30	0.0001	9	45	45.00	0.00
30	0.0001	8	45	45.01	-0.01
30	0.0004	10	60	60.00	0.00
30	0.0005	10	60	59.98	0.02
30	0.0006	10	60	60.02	-0.02
27	0.0001	10	90	90.83	-0.83
26	0.0001	10	90	90.91	-0.91
25	0.0001	10	90	91.00	-1.00
30	0.0001	7	120	120.23	-0.23
30	0.0001	6	120	120.19	-0.19
30	0.0001	5	120	120.00	0.00
30	0.0007	10	180	179.60	0.40
30	0.0008	10	180	179.60	0.40
30	0.0009	10	180	179.58	0.42
24	0.0001	10	270	269.13	0.87
23	0.0001	10	270	269.33	0.67
22	0.0001	10	270	269.24	0.76
30	0.0001	4	360	360.70	-0.70
30	0.0001	3	360	360.42	-0.42
30	0.0001	2	360	360.66	-0.66

Gambar 9 Tabel Pengujian PID

Tabel ini menguraikan hasil pengujian pada pengontrol PID dengan menggunakan nilai Setpoint, Kp, Ki, dan Kd yang bervariasi. Hasil ini menunjukkan bahwa nilai Derajat Motor semakin mendekati target yang diinginkan saat Setpoint naik. Meskipun demikian, kesalahan tetap ada, bahkan dengan penerapan konstanta PID untuk kontrol[13].

Jika nilai Kp terlalu rendah, kontrol kesalahan menjadi tidak efektif, sehingga gagal mencapai nilai Derajat Motor yang dapat diandalkan. Misalnya, Setpoint 270 digabungkan dengan nilai Kp 22 menghasilkan nilai Kesalahan 0,76 dalam pengujian. Ini menunjukkan bahwa motor 0,76 derajat dari target yang diinginkan[14].

### Kesimpulan

Penggunaan konstanta PID dalam sistem kontrol motor DC GM25-CPR telah terbukti efektif dalam mengatur kesalahan dan mencapai nilai Derajat Motor yang sesuai dengan setpoint yang ditentukan. Namun, pendekatan coba-coba diperlukan untuk mengatur konstanta PID, dan beberapa setpoint mungkin masih menunjukkan overshoot. Untuk mengatasinya, nilai Kd dapat ditingkatkan.

Selain itu, percobaan empiris telah menunjukkan bahwa menggunakan koefisien Kp yang kecil kurang optimal dalam mengatur kesalahan dan menghasilkan pengukuran derajat motor yang andal. Koefisien Kp yang lebih tinggi dapat menghasilkan pengukuran Derajat Motor yang lebih dekat dengan target, terutama dalam eksperimen dengan setpoint yang lebih besar[15].

### Ucapan Terima Kasih

Terima kasih koordinator tugas akhir teknik elektro dan dosen pembimbing saya atas saran dan sumbangan berharga mereka dalam penelitian jurnal saya. Saya juga ingin mengucapkan terima kasih kepada LPPM atas dukungan mereka dalam mengakomodasi jurnal penelitian saya.

## Daftar Pustaka

- [1] I. Hudati and U. G. Mada, "KENDALI MRAC PID PADA KECEPATAN MOTOR DC DENGAN," vol. 13, no. 2, pp. 1-14, 2022.
- [2] M. Pid, P. Robot, and P. Makanan, "PROSIDING SEMINAR NASIONAL," 2022.
- [3] M. Bahtiar *et al.*, "Penentuan Konstanta PID Sistem Kendali Satelit Sumbu X , Y , dan Z Menggunakan Metode Root Locus," vol. 20, no. 1, pp. 68-80, 2022.
- [4] E. W. Suseno, A. Ma, and R. D. Puriyanto, "Tuning Parameter Pengendali PID dengan Metode Algoritma Genetik pada Motor DC Tuning of PID Controller Parameters with Genetic Algorithm Method on DC Motor," vol. 8, no. 1.
- [5] M. Lamatenggo and Z. Acc, "Perancangan Balancing Robot Beroda Dua Dengan Metode Pengendali PID Berbasis Arduino Nano," vol. 2, pp. 39-43, 2020.
- [6] M. R. A. Nurkholis Putera and R. Hidayat, "Kendali Kecepatan Motor DC Menggunakan Pengendali PID dengan Encoder sebagai Feedback," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 7, no. 1, p. 50, 2022, doi: 10.30998/string.v7i1.13026.
- [7] A. Murtono, L. Kamajaya, and M. Shulton, "Implementasi Kontrol PID untuk Analisis Pengaturan Kecepatan Motor DC Menggunakan STM32," no. September, pp. 310-314, 2021.
- [8] A. Idir, "Design of an Optimally Tuned Fractionalized PID Controller for DC Motor Speed Control Via a Henry Gas Solubility Optimization Algorithm," vol. 15, no. 3, pp. 59-70, 2022, doi: 10.22266/ijies2022.0630.06.
- [9] C. Systems, E. S. Rahayu, and A. Cakan, "Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor," vol. 2, no. 2, pp. 435-447, 2022.
- [10] A. S. Ismailov, "Study of arduino microcontroller board," vol. 3, no. 3, pp. 172-179, 2022.
- [11] J. Marsh, C. Dunlap, S. Pierson, and H. Hu, "Introducing LabVIEW and Arduino as Data Acquisition System Alternatives," *ASEE Midwest Sect. Conf. 2022*, 2022.
- [12] R. Hartayu, S. Santoso, A. O. U. Kaleka, and M. K. Musakhhol, "Desain Simulasi Robot Keseimbangan Dua Roda Dengan Kecerdasan Buatan," *J. Sains dan Inform.*, vol. 6, no. 2, pp. 175-182, 2020, doi: 10.34128/jsi.v6i2.232.
- [13] S. Muharom, I. Masfufiah, R. A. Firmansyah, A. Hamid, and S. Oetomo, "Implementasi Kontrol Suhu Menggunakan Metode PID pada Aplikasi Inkubator Infant Warmers," *Cyclotron*, vol. 4, no. 1, pp. 55-59, 2021.
- [14] R. Muhandian and K. Krismadinata, "Kendali Kecepatan Motor DC Dengan Kontroller PID dan Antarmuka Visual Basic," *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 6, no. 1, p. 328, 2020, doi: 10.24036/jtev.v6i1.108034.
- [15] M. S. Zuhrie, P. W. Rusimamto, and N. Kholis, "Rancang Bangun PID Controller Dengan Tuning Ziegler Nicholas Untuk Pengendalian Posisi Sudut Motor DC," *J. Tek. Elektro*, vol. 10, no. 2, pp. 537-545, 2021.