

PENGEMBANGAN TRAINER PEMBELAJARAN KENDALI POSISI MOTOR DC DENGAN ALGORITMA PID

Oleh : Wahyu Sulistiyo¹, Aminuddin Rizal², Sheilla Nadia Valina³, Devi Yesitasari⁴

Teknik Elektro, Politeknik Negeri Semarang, Semarang, Indonesia

E-mail: wahyu.sulistiyo@polines.ac.id

Abstrak

Penelitian ini membahas perancangan dan implementasi sebuah trainer kendali posisi motor DC berbasis pengendali Proportional–Integral–Derivative (PID) untuk mendukung pembelajaran sistem kontrol di lingkungan pendidikan vokasi. Sistem menggunakan motor DC yang dilengkapi rotary encoder sebagai sensor umpan balik posisi, sedangkan unit pemroses utama menggunakan Arduino Uno untuk menjalankan algoritma PID secara real-time. Data posisi, sinyal kendali, serta parameter PID dikirimkan melalui komunikasi serial ke komputer dan divisualisasikan melalui aplikasi berbasis web, sehingga pengguna dapat melakukan pemantauan dan penyetelan parameter secara interaktif. Selain implementasi perangkat keras, penelitian ini juga menyediakan simulasi rangkaian sistem menggunakan Proteus untuk memvalidasi fungsi dasar sebelum realisasi fisik. Hasil pengujian menunjukkan bahwa trainer mampu mencapai kendali posisi dengan respon yang stabil dan error akhir yang rendah setelah penyesuaian parameter PID yang tepat. Trainer ini diharapkan dapat menjadi media pembelajaran yang efektif dalam memahami prinsip pengendalian PID, karakteristik motor DC, serta pemanfaatan sensor encoder dalam sistem kendali tertutup.

Kata kunci : antarmuka web, kontrol posisi, motor DC, PID.

Abstract

This study presents the design and implementation of a position-control trainer for a DC motor using a Proportional–Integral–Derivative (PID) controller, developed to support control-system learning in vocational education. The system employs a DC motor equipped with a rotary encoder as the feedback sensor, while an Arduino Uno serves as the main processing unit to execute the PID algorithm in real time. Position data, control signals, and PID parameters are transmitted via serial communication to a computer and visualized through a web-based application, enabling users to monitor the system and adjust controller parameters interactively. In addition to the hardware implementation, a Proteus simulation of the system is provided to validate basic functionalities prior to physical realization. Experimental results show that the trainer achieves stable position control with low steady-state error after appropriate PID tuning. The proposed trainer is expected to serve as an effective instructional tool for understanding PID control principles, DC motor characteristics, and the use of encoder sensors in closed-loop control applications.

Keywords : DC motor, PID, position control, web interface.

1. Latar Belakang

Sistem kendali merupakan salah satu fondasi utama dalam bidang teknik elektro, mekatronika, dan otomasi. Penguasaan konsep kendali, khususnya pengendali *Proportional–Integral–Derivative* (PID), menjadi kompetensi penting bagi mahasiswa vokasi karena pengendali ini digunakan secara luas dalam industri untuk mengatur posisi, kecepatan, dan

variabel proses lainnya. Statistik industri menunjukkan bahwa lebih dari 90% sistem kendali proses modern masih memanfaatkan PID karena kestabilan, kemudahan implementasi, dan ketangguhan terhadap gangguan sistem (Åström & Hägglund, 1995; Bennett, 1993).

Motor DC adalah aktuator yang banyak digunakan pada robotika, otomasi industri,

hingga perangkat presisi karena karakteristiknya yang linier dan mudah dikendalikan (Mohan, 2012; Rahman & Haque, 2010). Untuk mencapai pengendalian posisi yang akurat, diperlukan sistem umpan balik. Rotary encoder merupakan sensor yang umum digunakan untuk mengukur posisi dan kecepatan motor DC karena resolusi yang tinggi dan kemampuan deteksi gerakan dua arah (Fraden, 2010; Dorf & Bishop, 2017). Integrasi motor DC, encoder, dan algoritma PID menghasilkan sistem kendali tertutup (*closed-loop control*) yang mampu mencapai target posisi dengan error yang minimal.

Namun, penggunaan sistem kendali sering kali menjadi tantangan dalam proses pembelajaran karena mahasiswa perlu memahami konsep teoretis sekaligus mampu melakukan implementasi langsung (Nuzhnyy et al., 2021; Sholihin et al., 2022). Oleh karena itu, diperlukan trainer atau alat bantu pembelajaran yang dapat menjembatani teori dan praktik. Trainer kendali posisi berbasis PID menjadi solusi efektif karena memungkinkan mahasiswa mempelajari dinamika motor DC, karakteristik sinyal umpan balik, serta proses tuning PID secara interaktif (Hasan et al., 2020; Banzi & Shiloh, 2014).

Dalam penelitian ini, dikembangkan sebuah *trainer* kendali posisi motor DC berbasis PID menggunakan Arduino Uno sebagai pengendali utama. *Encoder rotary* digunakan sebagai sensor posisi, sementara antarmuka pemantauan dan *tuning* parameter dibangun berbasis web dengan komunikasi serial ke komputer. Untuk memvalidasi desain sistem, dilakukan simulasi rangkaian menggunakan Proteus sebelum direalisasikan dalam bentuk perangkat fisik. Pendekatan ini tidak hanya meningkatkan keandalan sistem, tetapi juga mempermudah proses pembelajaran karena mahasiswa dapat memahami hubungan antara simulasi dan implementasi nyata (Ibrahim, 2012; Hossain et al., 2019).

Trainer ini diharapkan dapat menjadi media pembelajaran yang komprehensif bagi mahasiswa vokasi dalam mempelajari konsep dasar sistem kendali, tuning PID, dan

implementasi *embedded system* secara langsung, sehingga mendukung peningkatan kompetensi dalam bidang otomasi dan kontrol industri (Franklin, Powell, & Emami-Naeini, 2015; Ogata, 2010; Dorf & Bishop, 2017).

2. Tinjauan Pustaka

2.1 Sistem Kendali dan Pengendali PID

Sistem kendali merupakan mekanisme yang digunakan untuk mengatur keluaran suatu sistem sehingga mengikuti nilai referensi yang diinginkan. Pada sistem kendali tertutup, sinyal umpan balik digunakan untuk mengurangi error antara keluaran dan masukan referensi.

Pengendali PID adalah salah satu jenis pengendali yang paling banyak digunakan dalam berbagai aplikasi industri karena desainnya yang sederhana, stabil, dan mudah diimplementasikan baik secara analog maupun digital. Pengendali PID bekerja berdasarkan kombinasi aksi proporsional (P), integral (I), dan derivatif (D) untuk mengatur sinyal kendali secara optimal.

Berbagai metode tuning PID seperti Ziegler–Nichols, Cohen–Coon, dan trial-and-error telah dikembangkan untuk memperoleh parameter yang sesuai dengan dinamika sistem (Franklin et al., 2015; Ogata, 2010).

2.2 Motor DC dalam Sistem Kendali Posisi

Motor DC merupakan aktuator yang banyak diaplikasikan pada robotika, otomasi industri, dan sistem mekanik yang memerlukan kontrol kecepatan atau posisi. Keunggulan motor DC meliputi respon cepat, karakteristik linier pada rentang tertentu, dan kemudahan pengaturan tegangan serta arus untuk mengontrol putaran (Rahman & Haque, 2010).

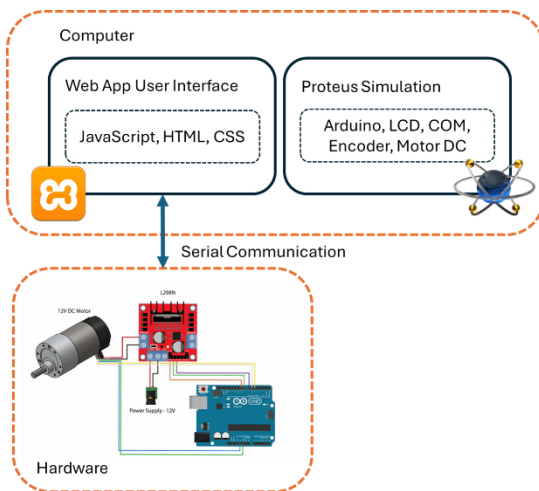
Dalam aplikasi kendali posisi, motor DC biasanya digunakan bersama sensor umpan balik sehingga dapat membentuk sistem kendali loop tertutup. Dengan

umpan balik posisi yang tepat, akurasi dan stabilitas kinerja motor dapat ditingkatkan secara signifikan.

2.3 Rotary Encoder sebagai Sensor Umpan Balik

Encoder rotary adalah sensor yang digunakan untuk menentukan posisi, arah, dan jumlah putaran motor secara presisi. Terdapat dua jenis encoder utama, yaitu incremental encoder dan absolute encoder, di mana incremental encoder lebih umum digunakan dalam sistem berbasis Arduino karena kemudahan integrasi dan biaya yang lebih rendah.

Encoder memberikan pulsa digital yang dapat dihitung oleh mikrokontroler untuk menentukan posisi aktual motor. Dengan resolusi tertentu (misalnya 200–600 pulses per revolution), encoder mampu memberikan informasi posisi yang cukup detail untuk kebutuhan kendali posisi (Fraden, 2010; Dorf & Bishop, 2017).



Gambar 1 diagram blok sistem utama

3. Metode

Metode penelitian ini diawali dengan perancangan arsitektur sistem kendali posisi motor DC yang terdiri dari dua bagian utama, yaitu bagian perangkat keras (*hardware*) dan bagian perangkat lunak (*software/web-based interface*) seperti yang terlihat pada Gambar 1. Kedua bagian ini saling terhubung melalui komunikasi serial sehingga proses kendali dan

pemantauan dapat dilakukan secara *real-time*.

Diagram sistem menunjukkan bahwa Arduino Uno, motor DC dengan enkoder, dan driver motor L298N berada pada sisi *hardware*, sedangkan antarmuka *monitoring* dan *tuning* PID dibangun menggunakan *web application* yang berjalan di komputer dan terhubung melalui XAMPP sebagai *local server*.

3.1 Perangkat Keras (*Hardware*)

Desain perangkat keras pada sistem trainer kendali posisi motor DC ini terdiri atas beberapa komponen utama yang bekerja secara terintegrasi, yaitu motor DC dengan *rotary encoder*, driver motor L298N, mikrokontroler Arduino Uno, serta catu daya eksternal. Seluruh komponen dirancang untuk membentuk sistem kendali loop tertutup (*closed-loop control system*) yang memungkinkan pengujian algoritma PID secara *real-time*. Gambar 2 memperlihatkan perangkat keras yang dikembangkan

3.1.1 Motor DC dengan *Encoder*

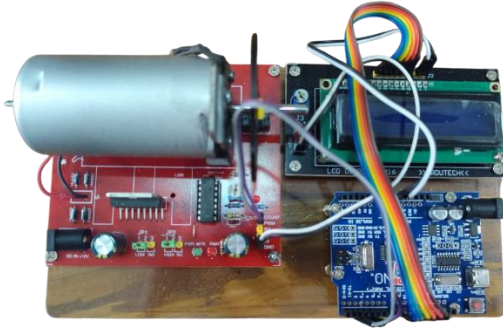
Aktuator utama pada sistem ini adalah motor DC 12 V yang dilengkapi rotary encoder incremental. Motor DC digunakan karena memiliki karakteristik dinamika yang sederhana, respon yang cepat, serta mudah dikendalikan melalui pengaturan tegangan dan arah arus. Keunggulan ini menjadikan motor DC sebagai aktuator yang ideal dalam eksperimen sistem kendali dan aplikasi robotika.

Rotary encoder incremental berfungsi sebagai sensor umpan balik (*feedback sensor*) untuk mengukur posisi sudut dan kecepatan putaran poros motor. *Encoder* menghasilkan dua sinyal pulsa kuadratur (A dan B) yang berbeda fase 90°. Dengan mendeteksi hubungan fase kedua sinyal tersebut, sistem dapat menentukan:

- arah putaran motor (*clockwise* atau *counterclockwise*),
- jumlah pulsa per satuan waktu (untuk perhitungan RPM),
- total akumulasi pulsa (untuk perhitungan posisi absolut atau relatif).

Penggunaan *incremental encoder* memberikan resolusi tinggi dan akurasi deteksi posisi yang sangat baik, sehingga sesuai untuk implementasi

algoritma kendali presisi seperti PID.



Gambar 2 realisasi perangkat keras

3.1.2 Driver Motor

Driver L298N digunakan sebagai penguat daya (*power stage*) yang menjembatani sinyal kendali logika dari Arduino dengan kebutuhan arus dan tegangan motor DC. L298N merupakan driver motor tipe H-Bridge ganda berbasis transistor BJT yang mampu mengendalikan motor DC secara:

- dua arah (*bidirectional*),
- kontrol kecepatan melalui PWM,
- kontrol mode energi (ENABLE/IN).

Pada penelitian ini, L298N menerima tiga sinyal input dari Arduino:

1. PWM (*Pulse Width Modulation*) — menentukan kecepatan motor.
2. IN1 & IN2 — menentukan arah putaran poros motor.

Driver ini dipilih karena kestabilannya, ketersediaannya yang luas, dan kemampuannya menangani arus motor hingga 2 A pada tegangan 5–12 V.

3.1.3 Mikrokontroler

Arduino Uno berbasis mikrokontroler ATmega328P berfungsi sebagai unit kendali utama yang menjalankan algoritma PID secara digital (Banzi & Shiloh, 2014). Peran Arduino dalam sistem meliputi:

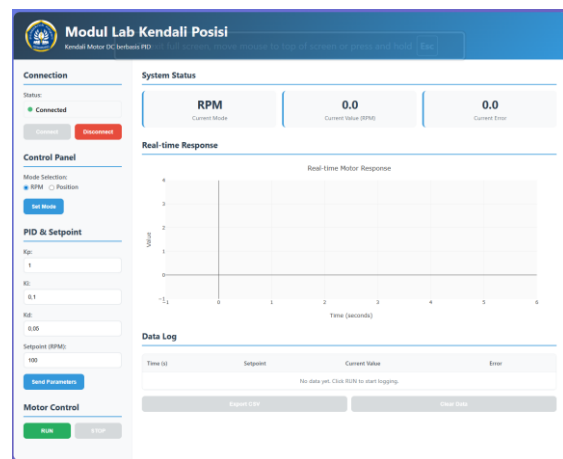
1. Pembacaan sensor encoder melalui interrupt eksternal, sehingga data pulsa dapat diperoleh secara akurat dan real-time.
2. Perhitungan kecepatan (RPM) dan posisi berdasarkan sinyal encoder.
3. Eksekusi algoritma PID untuk mengatur output aktuasi motor.

4. Pengiriman data (posisi, RPM, error, PWM) ke aplikasi web.
5. Penerimaan parameter kendali (K_p , K_i , K_d , mode, setpoint) melalui komunikasi serial.

Frekuensi operasi 16 MHz dan dukungan interrupt hardware membuat Arduino Uno sangat cocok untuk sistem kendali yang memerlukan respons cepat.

3.2 Perangkat Lunak (Software)

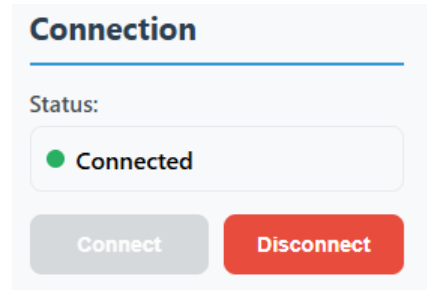
Antarmuka pengguna (User Interface/UI) dirancang berbasis web untuk memungkinkan pemantauan dan pengaturan sistem kendali motor DC secara real-time. Antarmuka ini diakses melalui browser seperti Chrome atau Edge dan berkomunikasi langsung dengan Arduino melalui port serial. Desain UI dibuat sederhana, responsif, dan mudah dipahami untuk mendukung proses pembelajaran dan eksperimen kontrol PID.



Gambar 3 tampilan utama user interface

UI terdiri dari beberapa komponen utama yang ditunjukkan pada Gambar 3.

3.2.1 Panel Koneksi Arduino



Gambar 4 panel koneksi Arduino

Panel ini berfungsi untuk menghubungkan komputer dengan perangkat Arduino Uno seperti pada Gambar 4. Fungsinya meliputi:

- Menampilkan status koneksi (Connected/Disconnected)
- Tombol Connect Arduino dan Disconnect
- Informasi persyaratan koneksi:
 - Browser kompatibel
 - Sketch Arduino telah mendukung komunikasi serial
 - Baud rate 9600
 - Koneksi USB aktif

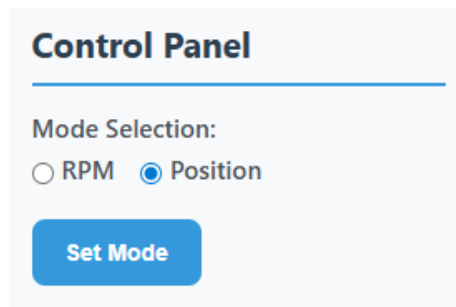
Panel ini memastikan pengguna melakukan koneksi perangkat dengan benar sebelum menjalankan sistem.

3.2.2 Panel Kontrol Mode

Panel ini digunakan untuk memilih mode operasi motor:

- RPM Mode digunakan untuk pengendalian kecepatan
- Position Mode digunakan pengendalian posisi berbasis encoder

Setelah memilih mode seperti pada Gambar 5, pengguna menekan tombol Set Mode untuk mengirim konfigurasi ke Arduino.



Gambar 5 panel mode

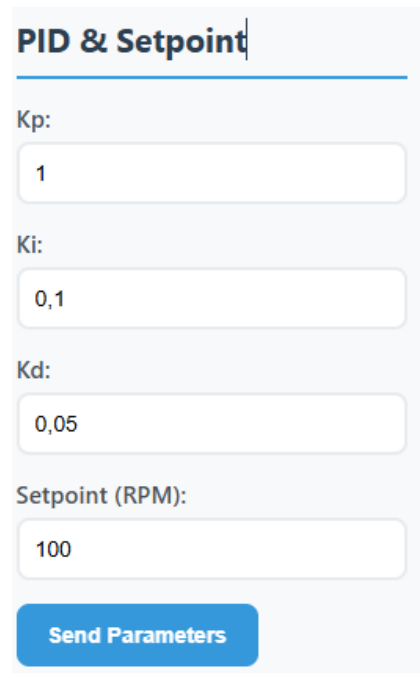
3.2.3 Panel Setting PID dan Set Point

Panel ini memungkinkan pengguna melakukan pengaturan parameter PID, yaitu:

- Kp (konstanta proporsional)
- Ki (konstanta integral)
- Kd (konstanta derivative)
- Setpoint (RPM atau posisi, tergantung mode)

Tombol *Send Parameters* mengirimkan nilai parameter kontrol ke Arduino untuk diimplementasikan secara *real-time*. Panel ini

memungkinkan proses *tuning* PID yang interaktif (Hasan et al., 2020). Penampakan panel ini terlihat pada Gambar 6.



Gambar 6 panel setting parameter

3.2.4 Real-time Data

Bagian ini menampilkan indikator numerik untuk:

- Mode berjalan (RPM/Position)
- Current Value (nilai aktual, rpm atau posisi encoder)
- Current Error (selisih antara setpoint dan nilai aktual)

Display ini memberikan visualisasi cepat terhadap kondisi sistem seperti yang ditampilkan pada Gambar 7.



Gambar 7 panel visualisasi data

3.2.5 Real-time Visualisasi

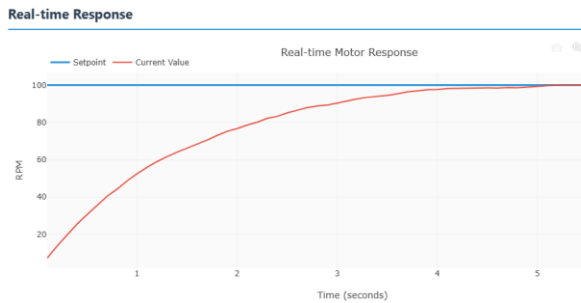
Grafik utama menampilkan dinamika motor terhadap waktu:

- nilai setpoint
- nilai aktual
- error
- respon PID

Grafik ini memungkinkan pengguna menganalisis:

- rise time
- overshoot
- settling time
- kestabilan sistem

Visualisasi ini sangat penting dalam proses tuning dan pembelajaran teori control seperti Gambar 8.



Gambar 8 panel tampilan grafik

3.2.6 Data Log dan Eksport

Panel ini (Gambar 9) berfungsi mencatat seluruh data real-time yang dikirimkan Arduino, mencakup:

- waktu
- setpoint
- nilai aktual
- error

Fitur tambahan:

- Export CSV digunakan menyimpan data untuk analisis lanjutan (MATLAB, Python, Excel).
- Clear Data digunakan untuk menghapus log untuk memulai percobaan baru.

Time (s)	Setpoint	Current Value	Error
45.0	100.0	102.8	-3.1
45.1	100.0	102.7	-2.8
45.2	100.0	102.4	-2.7
45.3	100.0	102.3	-2.4
45.4	100.0	102.4	-2.3
45.5	100.0	102.3	-2.4

Buttons: Export CSV, Clear Data

Gambar 9 panel tampilan data logger

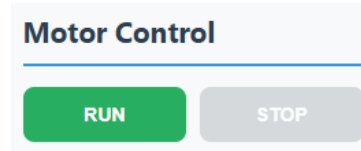
3.2.7 Kontrol Motor

Terdiri dari dua tombol:

- RUN Motor
- STOP Motor

Panel ini mengirimkan perintah langsung ke Arduino untuk memulai atau menghentikan

aktuasi motor DC. Tampilan panel seperti Gambar 10.



Gambar 20 panel menjalankan program

3.3 Logika Kerja

Arduino Uno digunakan sebagai pengendali utama yang menjalankan algoritma kendali PID secara *real-time* berdasarkan data umpan balik dari *rotary encoder*. Mode dan parameter yang diproses merupakan masukan dari *user interface*. Dan juga data hasil pemrosesan juga dikirimkan ke *user interface* untuk ditampilkan dan disimpan. Program Arduino terdiri dari tiga blok logika utama, yaitu pembacaan encoder, perhitungan feedback (posisi dan RPM), serta algoritma PID dan pengendalian motor, yang semuanya bekerja terintegrasi dalam interval waktu sampling tertentu. Diagram alir terlihat pada Gambar 11 untuk proses di mikrokontroler.

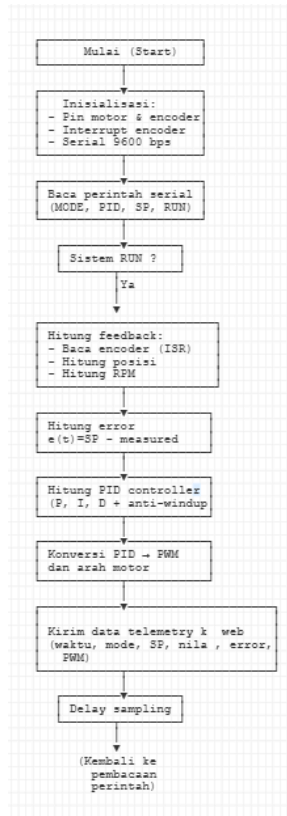
3.3.1 Pembacaan Rotary Encoder

Rotary encoder incremental digunakan untuk memperoleh informasi posisi dan arah putaran motor. Pembacaan encoder dilakukan menggunakan interrupt eksternal pada pin digital 2 dan 3. Setiap perubahan kondisi (*rising/falling edge*) pada kanal A dan B menghasilkan interupsi yang menambah atau mengurangi variabel penghitung pulsa.

Penggunaan interrupt memastikan bahwa pembacaan pulsa encoder memiliki akurasi tinggi dan bebas kehilangan pulsa, bahkan pada kecepatan motor yang tinggi. Nilai encoderCount ini menjadi dasar untuk menghitung:

- Posisi aktual (dalam satuan pulsa), dan
- Kecepatan putaran (RPM) berdasarkan selisih pulsa dalam interval waktu sampling.

Dengan demikian, Arduino memiliki umpan balik posisi yang presisi untuk kebutuhan kendali tertutup.



Gambar 31 flowchart program mikrokontroler

3.3.2 Perhitungan Posisi dan RPM

Pada setiap periode sampling (misalnya 50 ms), Arduino menghitung nilai umpan balik sebagai berikut:

1. Posisi aktual diperoleh langsung dari nilai total pulsa encoder. Nilai ini digunakan pada mode kendali posisi.
2. Kecepatan putaran (RPM) dihitung menggunakan selisih jumlah pulsa sejak periode sebelumnya (delta pulses). Rumus digunakan:

$$RPM = \frac{\Delta \text{pulses}}{PPR} \times 60 \quad (1)$$

Dengan:

- Δpulses = perubahan pulsa dalam interval sampling,
- PPR = pulses per rotation encoder,
- 60 untuk konversi detik ke menit.

Hasil ini digunakan pada mode kendali kecepatan (RPM control mode). Pemrosesan ini berjalan secara periodik untuk memberikan feedback real-time bagi algoritma PID.

3.3.3 Implementasi Logika PID

Arduino menjalankan algoritma Proportional–Integral–Derivative (PID) untuk mengatur kecepatan atau posisi motor. Parameter PID (K_p , K_i , K_d) dapat diubah langsung dari antarmuka web sehingga proses tuning dapat dilakukan secara interaktif.

Komponen logika PID yang diimplementasikan meliputi:

1. Error calculation

$$e(t) = r(t) - y(t) \quad (2)$$

Di mana $r(t)$ adalah setpoint, dan $y(t)$ adalah nilai posisi/RPM aktual.

2. Aksi proporsional (P)
Menghasilkan perubahan kendali sebanding besarnya error.
3. Aksi integral (I)
Mengakumulasi error dari waktu ke waktu sehingga mampu mengurangi *steady-state error*.
Integral dilengkapi anti-windup berupa batasan maksimum agar tidak terjadi limpahan (overshoot besar).
4. Aksi derivatif (D)
Mendeteksi perubahan laju error untuk meredam osilasi dan mempercepat stabilisasi.
5. Output PID
Keluaran PID dikonversi menjadi nilai PWM -255 hingga +255.

Nilai ini menentukan:

- Besarnya kecepatan (duty cycle PWM)
- Arah putaran (melalui kontrol pin DIR1 dan DIR2 pada driver L298N)

Dengan algoritma PID tersebut, sistem mampu mencapai target posisi atau kecepatan dengan error minimal.

3.3.4 Pengendalian Motor

Sinyal kendali dari PID digunakan untuk mengatur:

- Arah putaran motor (HIGH/LOW pada pair pin arah)
- Kecepatan motor (PWM pada pin enable)

Jika nilai PID > 0 maka motor berputar maju.

Jika nilai PID < 0 maka motor berputar mundur.

Jika nilai PID = 0 → maka motor berhenti.

Dengan demikian, keluaran mikrokontroler mengendalikan aktuator secara proporsional terhadap error.

3.3.5 Komunikasi User Interface

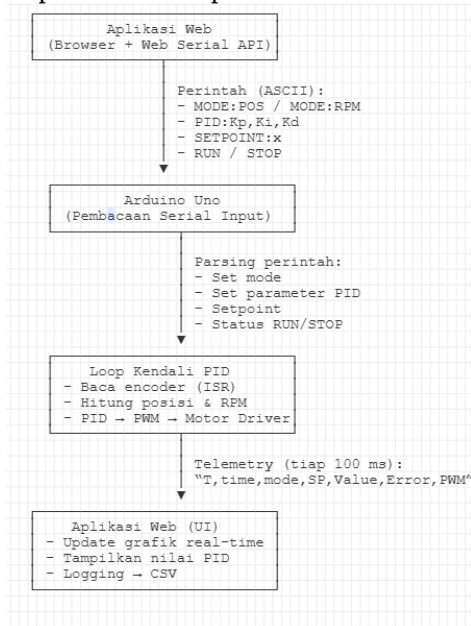
Arduino dan antarmuka web berkomunikasi melalui port serial dengan format pesan berbasis teks (ASCII). Perintah yang diterima dari web meliputi:

- memilih mode kendali (POSITION / RPM)
- mengirim parameter PID
- mengirim setpoint
- menjalankan atau menghentikan motor

Selain menerima perintah, Arduino juga mengirim telemetry real-time, mencakup:

- waktu (ms)
- mode yang aktif
- nilai setpoint
- nilai aktuator (posisi/RPM)
- error
- nilai PWM yang dihasilkan PID

Data ini kemudian divisualisasikan dalam grafik pada antarmuka web, serta dapat disimpan dalam format CSV untuk analisis lebih lanjut. Detail proses terlihat pada Gambar 12.



Gambar 42 flowchart program user interface

4. Eksperimen dan Hasil

Bagian ini membahas prosedur eksperimen yang dilakukan pada trainer kendali posisi motor DC berbasis PID, meliputi pengujian respon posisi, pengujian kecepatan (RPM), pengaruh

perubahan parameter PID, serta validasi hasil melalui visualisasi pada antarmuka web. Hasil eksperimen ditampilkan dalam bentuk tabel dan uraian analitis.

4.1 Setup Eksperimen

Pengujian dilakukan menggunakan konfigurasi perangkat keras:

- Motor DC 12 V dengan encoder 400 PPR
- Driver motor L298N
- Arduino Uno sebagai pengendali utama
- Aplikasi web sebagai media pemantauan dan pengaturan PID
- Sampling time Arduino: 50 ms
- Pengiriman telemetry ke web: 100 ms

Pengujian dilakukan dalam dua mode:

1. Position Control Mode
2. Speed (RPM) Control Mode

4.2 Eksperimen Kendali Posisi Motor DC

Pengujian dilakukan dengan memberikan beberapa nilai setpoint posisi (dalam satuan pulsa encoder). Respon posisi diuji pada parameter PID tertentu.

Hasil pada Tabel 1 menunjukkan bahwa sistem mampu mencapai posisi dengan kesalahan akhir yang sangat kecil (<10 pulsa), yang setara dengan error posisi $<4^\circ$. Overshoot masih dalam rentang yang dapat diterima untuk sistem mekanik ringan.

Tabel 4. Setpoint dan Respon Kendali Posisi

No	Setpoint (pulses)	Rise Time (ms)	Overshoot (%)	Steady-State Error (pulses)	Settling Time (ms)
1	500	210	3.2%	4	480
2	1000	260	5.8%	7	620
3	1500	300	6.1%	9	650

4.3 Eksperimen Kendali Kecepatan Motor (RPM Mode)

Pengujian RPM dilakukan dengan memberikan nilai setpoint kecepatan motor antara 1000 – 2500 RPM.

Sistem mampu mempertahankan kecepatan dengan error di bawah 1%. Hal ini menunjukkan pengendalian PID bekerja efektif terhadap

perubahan beban kecil.

Tabel 2 Setpoint dan Respon RPM

No	Setpoint (RPM)	RPM Aktual Rata-rata	Error (RPM)	Stabilitas	Waktu Mencapai Steady State (ms)
1	1000	995	-5	Stabil	350
2	1500	1490	-10	Stabil	420
3	2000	1982	-18	Stabil	480

4.4 Eksperimen Tuning Parameter PID

Untuk menganalisis pengaruh parameter PID, dilakukan pengujian perubahan nilai Kp, Ki, dan Kd dengan setpoint 1000 pulsa.

Tabel 3. pengaruh parameter PID

Kombinasi PID (Kp-Ki-Kd)	Rise Time	Overshoot	Settling Time	Steady-State Error
1.0 – 0.0 – 0.0	Cepat	Tinggi (18%)	Lama	Kecil (~10)
1.2 – 0.2 – 0.05	Sedang	Rendah (4%)	Cepat	Sangat kecil (~3)
0.8 – 0.3 – 0.10	Lambat	Hampir 0%	Sedang	Minimal (~2)

Interpretasi:

- Kp terlalu besar menjadikan overshoot meningkat
- Ki terlalu besar menghasilkan gerakan lamban dan cenderung osilatif
- Kd membantu meredam overshoot dan mempercepat settling time

Kombinasi optimal ditemukan pada Kp = 1.2, Ki = 0.2, Kd = 0.05 untuk perangkat keras yang digunakan.

4.5 Hasil Visualisasi pada Antarmuka Web

Antarmuka web menunjukkan grafik real-time dari:

- Setpoint
- Nilai posisi/RPM aktual
- Error
- Nilai PWM

Hasil pengamatan menunjukkan:

- Kurva posisi mengikuti setpoint dengan pola respons yang sesuai karakteristik PID.
- Error menyempit menuju nol seiring

waktu stabilisasi.

- PWM menunjukkan perubahan sesuai dinamika error (besar di awal, mengecil saat mendekati posisi).

Visualisasi ini memudahkan mahasiswa untuk:

- melihat respon sistem secara langsung,
- memahami efek tuning PID,
- menganalisis rise time, settling time, dan overshoot.

4.6 Validasi dengan Simulasi Proteus

Pengujian simulasi dilakukan dengan model Arduino, motor, encoder virtual.

Tabel 4 Perbandingan Simulasi dan Hardware

Parameter	Hasil Simulasi	Hasil Riil	Selisih
Rise Time	240 ms	260 ms	+20 ms
Overshoot	4%	5.8%	+1.8%
Steady-State Error	0 pulses	7 pulses	7

Selisih disebabkan oleh gesekan mekanis, delay driver motor, dan inersia sistem yang tidak termodelkan dalam simulasi.

5. Diskusi

Hasil eksperimen menunjukkan bahwa sistem *trainer* kendali posisi motor DC berbasis PID yang dikembangkan mampu memberikan kinerja yang stabil dan akurat. Pada pengujian kendali posisi, respon sistem memperlihatkan *rise time* yang relatif cepat dan *steady-state error* yang kecil, yaitu kurang dari 10 pulsa encoder atau sekitar 4° rotasi poros. Nilai *overshoot* yang muncul berkisar antara 3–6%, yang masih sesuai karakteristik umum sistem kendali proporsional-derivatif. *Settling time* yang berada antara 480–650 ms memperlihatkan bahwa sistem mampu mencapai kondisi stabil setelah beberapa osilasi kecil, dan hal ini selaras dengan teori bahwa aksi derivatif berfungsi meredam laju perubahan error dan menghindari *overshoot* yang berlebihan.

Pada mode kendali kecepatan, performa sistem menunjukkan error kecepatan di bawah 1%, menandakan bahwa perhitungan RPM berbasis selisih pulsa per interval sampling sudah cukup

akurat untuk mengendalikan kecepatan motor. Waktu mencapai *steady-state* pada mode RPM sedikit lebih cepat dibandingkan mode posisi karena karakteristik dinamika kendali kecepatan cenderung lebih sederhana dan tidak melibatkan integrator tambahan seperti pada kendali posisi. Stabilitas sistem yang baik pada berbagai nilai setpoint memperlihatkan bahwa algoritma PID digital yang diterapkan bekerja efektif meskipun terjadi variasi beban ringan.

Eksperimen tuning parameter PID mempertegas hubungan antara parameter K_p , K_i , dan K_d dengan karakteristik respon sistem. K_p yang besar memperpendek *rise time* namun meningkatkan overshoot, sementara K_i memberi kontribusi terhadap pengurangan *steady-state error* tetapi dapat memperlambat respon jika nilainya terlalu besar. K_d terbukti berperan penting dalam menurunkan overshoot serta mempercepat proses stabilisasi. Kombinasi parameter PID yang memberikan respon paling baik adalah $K_p = 1.2$, $K_i = 0.2$, dan $K_d = 0.05$. Hal ini menjadi pembelajaran penting bagi mahasiswa karena mereka dapat mengamati bagaimana perubahan setiap parameter PID memengaruhi dinamika sistem secara langsung melalui antarmuka web.

Perbandingan hasil eksperimen fisik dengan simulasi Proteus menunjukkan bahwa model simulasi mampu memprediksi perilaku dasar sistem, namun masih terdapat perbedaan yang disebabkan oleh faktor mekanis dan elektrik yang tidak termodelkan secara sempurna, seperti gesekan mekanik, inersia motor, delay switching pada driver L298N, serta noise pada pembacaan encoder. Meskipun demikian, perbedaan seperti error posisi dan settling time masih dalam batas yang dapat diterima, sehingga simulasi tetap efektif sebagai tahap awal validasi rancangan.

Peran antarmuka web pada sistem ini juga memberikan nilai tambah yang signifikan. Visualisasi real-time terhadap nilai setpoint, posisi/RPM aktual, error, dan PWM memudahkan pengguna dalam mempelajari konsep-konsep penting seperti *rise time*, *overshoot*, dan *settling time*. Kemampuan untuk melakukan tuning PID secara langsung dan melihat pengaruhnya terhadap grafik respon membantu memperkuat pemahaman mahasiswa

terhadap teori kendali. Selain itu, fitur penyimpanan data telemetry ke format CSV memungkinkan analisis lebih lanjut menggunakan perangkat lunak lain seperti MATLAB atau Python.

Walaupun sistem telah menunjukkan performa yang baik, masih terdapat beberapa keterbatasan seperti resolusi encoder yang terbatas, efisiensi driver L298N yang rendah, serta adanya delay akibat waktu sampling dan pengiriman data. Faktor mekanis dan ketidakidealan perangkat keras juga memengaruhi performa kendali secara keseluruhan. Namun demikian, sistem ini tetap memenuhi standard sebagai trainer pendidikan yang efektif dan mampu memberikan pemahaman menyeluruh mengenai implementasi kendali PID pada motor DC.

Secara umum, hasil penelitian membuktikan bahwa trainer kendali PID yang dikembangkan dapat memberikan pengalaman pembelajaran yang komprehensif, realistis, dan mudah dipahami. Respon sistem yang sesuai teori, kemudahan visualisasi, dan integrasi dengan antarmuka web menjadikan trainer ini layak digunakan sebagai media praktikum pada mata kuliah sistem kendali, mekatronika, atau *embedded system* di pendidikan vokasi.

6. Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sebuah *trainer* kendali posisi motor DC berbasis algoritma PID yang dapat digunakan sebagai media pembelajaran pada bidang sistem kendali dan mekatronika. Berdasarkan hasil eksperimen, sistem mampu mencapai performa kendali yang stabil baik pada mode posisi maupun kecepatan. Pada pengujian kendali posisi, sistem menunjukkan *rise time* yang cepat, overshoot yang rendah, serta *steady-state error* yang kecil, yaitu kurang dari 10 pulsa encoder. Hal ini menandakan bahwa konfigurasi PID yang digunakan telah mampu mengendalikan dinamika motor secara efektif. Sementara itu, pada pengujian kendali kecepatan, sistem mempertahankan error kecepatan di bawah 1%, yang memperlihatkan bahwa metode perhitungan RPM menggunakan incremental encoder cukup akurat dan responsif terhadap perubahan kondisi.

Antarmuka web yang dikembangkan terbukti memberikan kemudahan dalam proses pemantauan dan tuning parameter PID secara real-time. Visualisasi kurva respon, nilai error, dan sinyal PWM memberikan pemahaman intuitif mengenai karakteristik sistem kendali, sehingga menjadikan trainer ini relevan sebagai alat praktikum pendidikan vokasi. Selain itu, hasil validasi menggunakan simulasi Proteus menunjukkan konsistensi yang baik dengan hasil pengujian perangkat keras, meskipun terdapat perbedaan kecil yang disebabkan oleh faktor mekanis dan ketidakidealan komponen fisik.

Secara keseluruhan, trainer kendali posisi motor DC berbasis PID ini telah memenuhi tujuan penelitian, yaitu menyediakan platform pembelajaran praktis yang efektif, mudah dioperasikan, dan mampu memperlihatkan hubungan erat antara teori sistem kendali dengan implementasi nyata. Sistem ini dapat digunakan sebagai sarana pengajaran pada mata kuliah sistem kendali, robotika dasar, atau *embedded system*, serta berpotensi dikembangkan lebih lanjut menjadi modul pembelajaran yang lebih komprehensif.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada pemberi dana penelitian ini, yaitu DIPA Politeknik Negeri Semarang skema Penelitian Terapan Pratama (PTP) tahun 2025 dengan SK Nomor 0214/PL4.7.2/SK/2025, serta kepada pihak-pihak yang membantu pelaksanaan. Kami juga mengucapkan terima kasih kepada pihak-pihak yang terlibat dalam penelitian ini, dan kepada Politeknik Negeri Semarang atas dukungan tempat yang diberikan sehingga penelitian ini dapat terlaksana sesuai yang diharapkan.

DAFTAR PUSTAKA

Åström, K. J., & Hägglund, T. (1995). *PID Controllers: Theory, Design, and Tuning* (2nd ed.). Instrument Society of America.

Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino* (3rd ed.). Maker Media.

Bennett, S. (1993). *A History of Control Engineering*. Institution of Engineering and Technology.

Dorf, R. C., & Bishop, R. H. (2017). *Modern Control Systems* (13th ed.). Pearson.

Erickson, R. W., & Maksimovic, D. (2001). *Fundamentals of Power Electronics*. Springer.

Fraden, J. (2010). *Handbook of Modern Sensors* (4th ed.). Springer.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2015). *Feedback Control of Dynamic Systems* (7th ed.). Pearson.

Hasan, M. K., Islam, S., Hossain, M., & Sulaiman, R. (2020). Web-based monitoring and control systems: A review. *Journal of Industrial Information Integration*, 19, 100160.

Hossain, M., Ali, M. A. S., & Rahman, M. (2019). Simulation and implementation of DC motor control using microcontroller in Proteus. *International Journal of Scientific Research in Engineering and Management*, 3(5), 15–21.

Ibrahim, A. M. (2012). *Proteus PCB Design and Simulation*. Newnes.

Lin, J., & Chen, Z. (2021). Web-based real-time monitoring systems in modern automation. *International Journal of Automation Technology*, 15(4), 455–463.

Mohan, N. (2012). *Electric Machines and Drives: A First Course*. Wiley.

Nuzhnyy, S., Khomiak, Y., & Gerasymenko, V. (2021). Development of educational laboratory stands for studying control systems. *International Journal of Engineering Pedagogy*, 11(2), 25–36.

Ogata, K. (2010). *Modern Control Engineering* (5th ed.). Prentice Hall.

Rahman, M. A., & Haque, M. E. (2010). Control of DC motor using PWM and PID controller. *International Journal of Engineering*

Research and Applications, 2(3), 11–19.

Journal of Engineering Education, 35(6), 1850–1858.

Shinners, S. M. (1998). *Advanced Modern Control System Theory and Design*. Wiley.

Sholihin, M., Santoso, H., & Haryanto, A. (2022). Implementation of mechatronics training kits to improve vocational competencies. *Journal of Vocational Education Studies*, 5(1), 45–53.

Singh, A., & Kumar, R. (2019). Development of low-cost educational control trainer using DC motor and microcontroller. *International*