TIRE 1 (2026) 2

Technology, Innovation and Research in Engineering

homepage: https://journal.grahawidya.org/index.php/tire

# Implementasi IoT dalam Monitoring dan Optimasi Kendali Motor Servo Berbasis PID pada Lini Produksi

## *IoT Implementation in Monitoring and Optimization of PID-Based Servo Motor Control in Production Lines*

**Muchammad Amar Shahreza[1]*, Muhamad Rohid Syah Putra[1]**

[1]Jurusan Teknik Mesin, Fakultas Teknik Dan Informatika, Universitas PGRI Semarang, Semarang, Indonesia.

*Corresponding author
*E-mail:* amarshahreza10@gmail.com, rohidfans@gmail.com

INFO ARTIKEL

*Kata kunci*:
Internet of Things;
Kendali PID;
Lini produksi;
Monitoring;
Motor servo
*Keywords*:
*Internet of Things;*
*PID control;*
*Production line;*
*Monitoring;*
*Servo motor*

Received: 11 Juli 2026
Revised: 18 September 2026
Online published: 31 Oktober 2026

**A B S T R A K**

Penelitian ini mengembangkan dan mengimplementasikan sistem monitoring serta optimasi kendali motor servo berbasis Internet of Things (IoT) pada lini produksi. Masalah penelitian berfokus pada respons kendali yang lambat, overshoot, error posisi, dan keterbatasan pemantauan serta tuning dari jarak jauh. Sistem menggunakan ESP32 sebagai pusat kendali, sensor encoder sebagai umpan balik posisi, driver motor, motor servo, Arduino IDE, dan dashboard IoT. Eksperimen divariasikan menjadi tiga skenario: parameter PID awal tanpa tuning, tuning Ziegler–Nichols, dan optimasi PID berbasis IoT melalui remote tuning. Data diambil dari respons posisi motor dan komunikasi IoT, kemudian direduksi menggunakan parameter rise time, overshoot, settling time, steady-state error, latensi, dan keberhasilan pengiriman data. Hasil menunjukkan keberhasilan pengiriman data 95,2–100%. Optimasi IoT menghasilkan rise time 0,4 s, overshoot 2,8%, settling time 0,6 s, dan steady-state error 0,1%, dibandingkan kondisi awal 1,8 s; 25,4%; 3,5 s; dan 5,2%. Hasil tersebut menunjukkan bahwa integrasi IoT dengan PID mempercepat respons, meningkatkan kestabilan dan akurasi, serta memungkinkan pemantauan dan penyesuaian parameter secara jarak jauh pada lini produksi.

**A B S T R A C T**

*This study develops and implements an Internet of Things (IoT)-based monitoring and optimization system for servo motor control in a production line. The research addresses slow control response, overshoot, position error, and limited remote monitoring and tuning. The system uses an ESP32 controller, an encoder for position feedback, a motor driver, a servo motor, Arduino IDE, and an IoT dashboard. Three control scenarios were evaluated: initial PID parameters without tuning, Ziegler–Nichols tuning, and IoT-based PID optimization through remote tuning. Data were collected from motor position response and IoT communication, then reduced using rise time, overshoot, settling time, steady-state error, latency, and delivery success rate. The IoT communication achieved 95.2–100% delivery success. IoT-based optimization produced a rise time of 0.4 s, overshoot of 2.8%, settling time of 0.6 s, and steady-state error of 0.1%, compared with 1.8 s, 25.4%, 3.5 s, and 5.2% in the initial condition. The integration of IoT and PID therefore.*

## 1. Pendahuluan

Perkembangan Internet of Things (IoT) mendorong sistem otomasi industri menuju pengendalian yang terhubung, terdokumentasi, dan dapat dipantau dari jarak jauh. Pada lini produksi, motor servo digunakan ketika posisi dan gerak harus mengikuti setpoint secara presisi. Kontrol Proportional–Integral–Derivative (PID) banyak dipakai karena strukturnya sederhana dan mampu memperbaiki respons serta kestabilan sistem [1–4]. Namun, parameter PID yang tidak sesuai dapat menghasilkan overshoot, osilasi, settling time yang panjang, dan error posisi sehingga menurunkan kualitas respons aktuator.

Permasalahan tersebut menjadi semakin penting ketika proses tuning dan diagnosis harus dilakukan pada mesin yang sedang beroperasi. Penyesuaian parameter secara konvensional membutuhkan akses langsung ke panel atau pengendali, sedangkan data performa belum selalu tersedia secara terpusat. Studi tentang PID berbasis Blynk, pengendalian motor berbasis Arduino, serta monitoring berbasis IoT menunjukkan bahwa konektivitas dapat memperluas kemampuan monitoring dan kendali [2-16]. Dengan

doi
ISSN xxxxxxxx (online), xxxx (print)/© 2026.
Tire is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

demikian, integrasi IoT dengan kontrol PID memiliki urgensi untuk mendukung pengaturan parameter yang lebih fleksibel dan pengawasan kondisi motor secara real-time.

Penelitian terdahulu menggunakan variasi metode dan objek yang berbeda. Karuniawan et al. mengembangkan trainer kendali PID kecepatan motor DC berbasis PLC-HMI [1]. Marizka et al. menerapkan PID motor DC berbasis Blynk untuk meningkatkan fleksibilitas [2]. Andreas et al. menggunakan PID-CSA berbasis Arduino untuk kendali kecepatan motor pompa [3]. Suryatini et al. mengoptimasi PID motor DC dengan particle swarm optimization [4]. Imami et al. menguji kontrol posisi motor DC menggunakan PID berbasis Arduino [9], sedangkan Sholahuddin et al. mengkaji kendali posisi servo menggunakan PID-Ziegler–Nichols [11]. Lioe dan Al Rasyid menunjukkan penggunaan MQTT dan Node-RED untuk monitoring suhu dan getaran motor induksi lini produksi [12]. Putra et al. menerapkan dashboard Blynk untuk monitoring parameter kelistrikan mesin industri [15], sementara Prasad et al. membahas remote tuning dan monitoring PID untuk aktuator industri [16]. Hasil-hasil tersebut memperlihatkan manfaat PID dan IoT, tetapi belum secara spesifik menyajikan satu alur eksperimen yang menghubungkan variasi tuning PID, metrik respons transien, kualitas komunikasi IoT, serta remote tuning pada objek motor servo lini produksi dalam satu sistem.

Gap penelitian terletak pada kebutuhan integrasi antara loop kendali lokal yang cepat dengan lapisan IoT yang menyediakan monitoring dan perubahan parameter dari jarak jauh. Novelty penelitian ini adalah implementasi ESP32 sebagai pusat kendali PID yang menerima umpan balik encoder dan terhubung ke dashboard IoT sehingga tiga kondisi kendali—tanpa tuning, Ziegler–Nichols, dan optimasi berbasis IoT—dapat dibandingkan menggunakan parameter respons yang sama. Kebaruan tersebut relevan karena kualitas kendali tidak hanya dinilai dari respons motor, tetapi juga dari kemampuan komunikasi untuk mendukung operasi dan tuning jarak jauh.

Tujuan penelitian adalah (1) merancang sistem monitoring dan kendali servo berbasis ESP32, encoder, PID, dan IoT; (2) membandingkan performa tiga variasi kendali; dan (3) mengevaluasi kemampuan komunikasi IoT untuk monitoring serta remote tuning. Manfaat penelitian adalah menyediakan rancangan kendali yang dapat dipantau dan disetel tanpa harus selalu berada di lokasi mesin. Implikasi hasilnya bagi otomasi industri adalah tersedianya data performa yang dapat digunakan untuk diagnosis dan penyesuaian parameter secara lebih cepat, dengan catatan implementasi lapangan tetap bergantung pada keandalan jaringan dan.

## 2. Metode

### 2.1. Bahan dan Spesifikasi Sistem

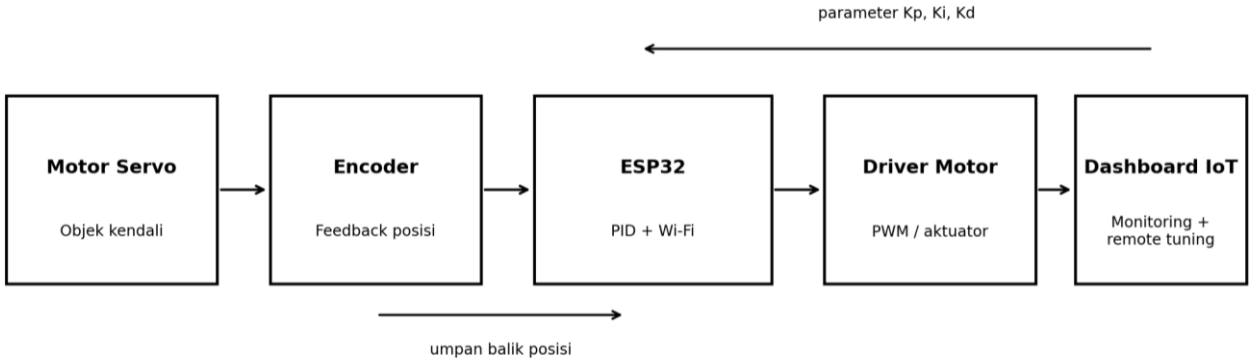
Objek penelitian adalah sistem kendali motor servo pada lini produksi. Berdasarkan naskah sumber, komponen utama yang digunakan adalah ESP32, sensor encoder, motor servo, driver motor, jaringan Wi-Fi, Arduino IDE, dan platform IoT. Naskah sumber tidak mencantumkan tipe/model motor servo, resolusi encoder, model driver, spesifikasi catu daya, serta nama platform IoT final secara spesifik. Karena itu, parameter tersebut tidak diasumsikan dalam revisi ini dan perlu dilengkapi dari catatan eksperimen penulis sebelum submission.

**Tabel 1.** Spesifikasi komponen dan fungsi sistem berdasarkan informasi yang terdokumentasi.

| Komponen | Spesifikasi yang terdokumentasi | Fungsi |
|---|---|---|
| Mikrokontroler | ESP32 | Pusat komputasi PID dan komunikasi Wi-Fi |
| Sensor | Encoder | Umpan balik posisi/kecepatan aktual motor |
| Aktuator | Motor servo | Objek kendali posisi pada lini produksi |
| Penggerak | Driver motor | Menerima sinyal PWM dari pengendali |
| Perangkat lunak | Arduino IDE | Pengembangan program PID dan komunikasi |
| IoT | Dashboard IoT; sumber menyebut Blynk/Node-RED/MQTT sebagai opsi | Monitoring real-time dan remote tuning |
| Jaringan | Wi-Fi; pengujian lokal dan jarak jauh | Transmisi data monitoring |

## 2.2 Model Fisik/Arsitektur Alat

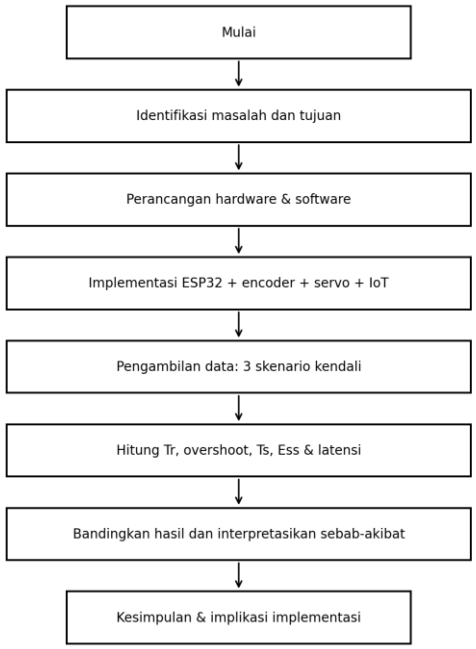
Model fisik konseptual sistem pada Gambar 1 memperlihatkan hubungan antara motor servo sebagai objek kendali, encoder sebagai umpan balik, ESP32 sebagai pengendali PID dan penghubung Wi-Fi, driver sebagai antarmuka aktuator, serta dashboard IoT untuk monitoring dan remote tuning. Gambar ini merepresentasikan konfigurasi sistem yang dijelaskan dalam naskah; foto perangkat aktual belum tersedia pada sumber.




**Gambar 1.** Model fisik konseptual dan arsitektur sistem monitoring serta kendali servo berbasis IoT.

## 2.3 Skema Eksperimen dan Alur Penelitian

Eksperimen dimulai dari identifikasi masalah, perancangan hardware-software, implementasi sistem, pengambilan data pada tiga skenario kendali, reduksi data, dan interpretasi sebab-akibat. Pada setiap skenario, setpoint diberikan pada motor servo dan respons posisi dibaca melalui encoder. Data dikirim ke dashboard IoT sehingga performa kendali dan komunikasi dapat dievaluasi pada sisi lokal maupun jarak jauh.




**Gambar 2.** Alur penelitian dan skema eksperimen.

2.4 Variasi Eksperimen

Tabel 2. Variasi eksperimen kendali PID.

| Skenario | Variasi kendali | Tujuan |
|---|---|---|
| S1 | PID awal tanpa tuning | Menetapkan kondisi dasar respons |
| S2 | PID tuning Ziegler–Nichols | Menguji pengaruh tuning konvensional |
| S3 | Optimasi PID berbasis IoT | Menguji remote tuning dan respons hasil optimasi |

**2.5 Measurement Method dan Pengambilan Data**

Metode pengukuran dilakukan dengan membaca posisi aktual motor dari encoder sebagai feedback terhadap setpoint. Data respons digunakan untuk menentukan rise time (Tr), overshoot, settling time (Ts), dan steady-state error (Ess). Pada lapisan IoT, data dikirim dari ESP32 menuju dashboard dan diukur berdasarkan latensi pengiriman serta keberhasilan penerimaan. Pengujian komunikasi dilakukan pada tiga kondisi yang terdokumentasi: jaringan Wi-Fi lokal, jaringan seluler untuk kendali jarak jauh, dan jaringan padat pada jam sibuk. Setiap hasil dilaporkan sebagai rentang atau persentase sesuai data pada naskah sumber.

$$e(t)=r(t)-y(t) \quad (1)$$

$$u(t)=K_p\, e(t)+K_i \int_0^t e(\tau)d\tau+K_d\, de(t)/dt \quad (2)$$

Bentuk PID tersebut digunakan untuk menjelaskan hubungan matematis antara error dan sinyal kendali [13].

**2.6 Reduksi Data**

Nilai penting hasil penelitian direduksi dari data pengukuran menggunakan metrik berikut. Rise time adalah waktu respons mencapai daerah awal menuju setpoint sesuai definisi pengujian; overshoot menunjukkan deviasi maksimum di atas setpoint; settling time menunjukkan waktu hingga respons berada pada pita kestabilan yang digunakan; dan steady-state error menunjukkan selisih relatif pada kondisi tunak.

$$\text{Overshoot (\%)} = (y_{max} - y_{setpoint}) / y_{setpoint} \times 100\% \quad (3)$$

$$\text{Steady-state error (\%)} = |y_{setpoint} - y_{ss}| / y_{setpoint} \times 100\% \quad (4)$$

$$\text{Latensi rata-rata} = (\Sigma\, t_i) / N \quad (5)$$

$$\text{Keberhasilan pengiriman (\%)} = N_{terima} / N_{kirim} \times 100\% \quad (6)$$

Persamaan (3–6) digunakan untuk memastikan bahwa hasil tabel dapat ditelusuri ke parameter pengukuran. N adalah jumlah paket atau pengamatan, t_i adalah latensi pada pengamatan ke-i, y_max adalah keluaran maksimum, dan y_ss adalah keluaran pada kondisi tunak.

## 3. Hasil dan Pembahasan

**3.1 Visualisasi Objek dan Implementasi Sistem**

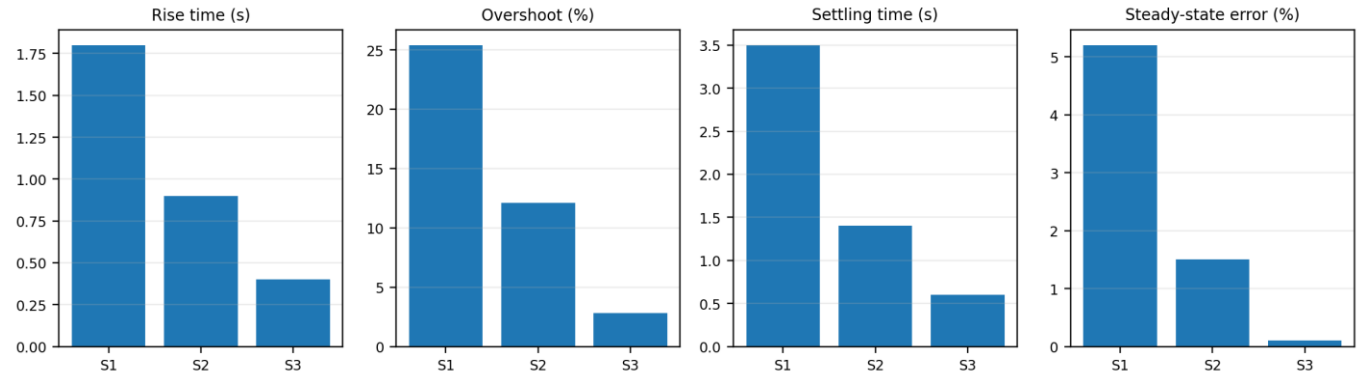
Gambar 1 memperlihatkan objek penelitian sebagai suatu loop kendali tertutup. Encoder mengukur posisi aktual motor dan mengirimkan feedback ke ESP32. ESP32 menghitung aksi PID dan mengirimkan sinyal ke driver motor. Di sisi lain, koneksi Wi-Fi mengirimkan data ke dashboard dan menyediakan jalur perubahan Kp, Ki, dan Kd secara remote. Fenomena utama pada sistem adalah adanya interaksi antara respons motor dan kualitas komunikasi: loop kendali membutuhkan respons cepat, sedangkan fungsi IoT menambahkan latensi komunikasi yang harus dipantau.

**3.2 Fenomena Respons Kendali dan Hubungan Sebab-Akibat**

Hasil pada Tabel 3 menunjukkan pola yang konsisten: ketika parameter PID dioptimasi, rise time, overshoot, settling time, dan steady-state error semuanya menurun. Pada S1, parameter awal menghasilkan rise time 1,8 s dan overshoot 25,4%, disertai settling time 3,5 s serta steady-state error 5,2%. Setelah tuning Ziegler–Nichols pada S2, masing-masing berubah menjadi 0,9 s; 12,1%; 1,4 s; dan 1,5%. Pada S3, optimasi PID berbasis IoT menghasilkan 0,4 s; 2,8%; 0,6 s; dan 0,1%.

Hubungan sebab-akibat yang dapat ditarik dari data adalah bahwa penyesuaian Kp, Ki, dan Kd mengubah keseimbangan antara kecepatan respons, redaman, dan kemampuan sistem menghilangkan error tunak. Parameter awal belum menghasilkan respons yang teredam dengan baik sehingga overshoot dan settling time lebih besar. Tuning Ziegler–Nichols memperbaiki kondisi

tersebut, sedangkan optimasi berbasis IoT memungkinkan parameter disesuaikan secara remote hingga diperoleh respons yang lebih dekat ke setpoint. Kesimpulan kausal ini dibatasi oleh desain eksperimen yang tidak melaporkan nilai numerik Kp, Ki, dan Kd untuk setiap skenario; karena itu, perubahan parameter spesifik tidak dapat direkonstruksi dari naskah sumber.




Gambar 3. Visualisasi perbandingan metrik respons untuk tiga skenario kendali.

Tabel 3. Hasil tuning parameter kendali PID.

| Parameter | S1: tanpa tuning | S2: Ziegler–Nichols | S3: optimasi IoT |
|---|---|---|---|
| Rise time (s) | 1,8 | 0,9 | 0,4 |
| Overshoot (%) | 25,4 | 12,1 | 2,8 |
| Settling time (s) | 3,5 | 1,4 | 0,6 |
| Steady-state error (%) | 5,2 | 1,5 | 0,1 |

### 3.3 Hasil Fungsionalitas IoT

Tabel 4. Hasil pengujian fungsionalitas dan latensi IoT.

| Kondisi jaringan | Latensi rata-rata (ms) | Keberhasilan pengiriman (%) | Status koneksi |
|---|---|---|---|
| Lokal (Wi-Fi) | 12–25 | 100 | Sangat stabil |
| Jarak jauh (jaringan seluler) | 45–78 | 98,6 | Stabil |
| Jaringan padat (peak hours) | 95–120 | 95,2 | Cukup stabil |

Tabel 4 menunjukkan bahwa kualitas komunikasi berubah mengikuti kondisi jaringan. Pada jaringan lokal, latensi berada pada 12–25 ms dengan keberhasilan pengiriman 100%. Pada jaringan seluler, latensi meningkat menjadi 45–78 ms dan keberhasilan pengiriman menjadi 98,6%. Pada jaringan padat, latensi tertinggi mencapai 95–120 ms dan keberhasilan pengiriman turun menjadi 95,2%. Fenomena ini menunjukkan hubungan sebab-akibat antara kepadatan jaringan dan keterlambatan/kehilangan paket. Walaupun komunikasi tetap dapat digunakan untuk monitoring, latensi jaringan perlu dipertimbangkan ketika IoT dipakai untuk fungsi kendali yang sangat sensitif terhadap waktu. Karena arsitektur penelitian menempatkan loop PID pada ESP32 dan IoT terutama untuk monitoring/remote tuning, pemisahan tersebut membantu menjaga kendali lokal tidak sepenuhnya bergantung pada internet.

### 3.4 Perbandingan dengan Penelitian Terdahulu

Temuan penelitian ini sejalan dengan studi terdahulu yang menunjukkan manfaat PID untuk memperbaiki respons motor dan manfaat IoT untuk monitoring atau pengaturan jarak jauh [1–4], [9–12], [15, 16]. Perbedaannya terletak pada objek dan integrasi: penelitian ini menguji motor servo pada lini produksi dan membandingkan tiga kondisi kendali yang sama-sama dievaluasi melalui metrik respons transien, sekaligus mengukur kualitas komunikasi IoT. Hasil numerik penelitian menunjukkan

penurunan rise time dari 1,8 s menjadi 0,4 s, overshoot dari 25,4% menjadi 2,8%, settling time dari 3,5 s menjadi 0,6 s, dan steady-state error dari 5,2% menjadi 0,1%. Angka tersebut menjadi dasar interpretasi bahwa optimasi yang dilakukan meningkatkan respons sistem pada kondisi eksperimen yang dilaporkan.

### 3.5 Implikasi dan Keterbatasan

Implikasi praktis hasil penelitian adalah operator dapat memantau posisi motor dan melakukan penyesuaian Kp, Ki, dan Kd tanpa berada langsung di dekat mesin. Dokumentasi data melalui dashboard juga dapat membantu evaluasi performa dan diagnosis. Namun, hasil belum dapat digeneralisasi untuk semua motor servo karena tipe motor, resolusi encoder, driver, beban, periode sampling, dan nilai Kp–Ki–Kd tidak dicantumkan secara lengkap pada naskah sumber. Selain itu, pengujian komunikasi hanya dilaporkan pada tiga kondisi jaringan dan belum memuat uji ketahanan jangka panjang atau pengaruh kehilangan koneksi terhadap keselamatan mesin. Untuk submission final, parameter eksperimen tersebut perlu dilengkapi dari catatan laboratorium agar reproduktibilitas penelitian meningkat.

## 4. Kesimpulan

Penelitian ini menggunakan metode eksperimen untuk mengembangkan sistem mekatronika berbasis IoT dengan ESP32, encoder, motor servo, driver motor, Arduino IDE, dan dashboard IoT, dengan variasi kendali berupa PID awal tanpa tuning, tuning Ziegler–Nichols, dan optimasi PID berbasis IoT melalui remote tuning. Pengukuran dilakukan terhadap respons posisi motor dan komunikasi IoT, sedangkan data direduksi menggunakan rise time, overshoot, settling time, steady-state error, latensi, dan keberhasilan pengiriman data. Hasil menunjukkan bahwa keberhasilan pengiriman data berada pada 95,2–100% dan skenario optimasi IoT menghasilkan rise time 0,4 s, overshoot 2,8%, settling time 0,6 s, serta steady-state error 0,1%, dibandingkan kondisi awal masing-masing 1,8 s, 25,4%, 3,5 s, dan 5,2%. Implementasi sistem memungkinkan monitoring real-time dan remote tuning parameter PID sehingga dapat mendukung pengoperasian lini produksi yang membutuhkan respons posisi presisi dan pemantauan jarak jauh. Dengan demikian, tujuan pengembangan dan pengujian sistem tercapai pada kondisi eksperimen yang dilaporkan, sementara penerapan lebih luas masih memerlukan pelengkapan spesifikasi perangkat dan pengujian ketahanan jaringan serta operasi jangka panjang.

### Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada pihak yang mendukung pelaksanaan penelitian dan penyusunan artikel ini.

### Kontribusi Penulis

Muchammad Amar Shahreza: Konseptual, Metodologi, Investigasi, Penulisan—draf awal. Muhamad Rohid Syah Putra: Supervisi, Validasi, Penulisan—review dan editing.

### Pernyataan Konflik Kepentingan

Penulis menyatakan tidak memiliki konflik kepentingan finansial atau hubungan pribadi yang diketahui dapat mempengaruhi pekerjaan yang dilaporkan dalam naskah ini.

### Pengungkapan Penggunaan AI

AI digunakan untuk membantu pemformatan dan penyusunan ulang naskah berdasarkan materi sumber dan komentar reviewer. Analisis substantif, interpretasi hasil, dan validasi data tetap menjadi tanggung jawab penulis.

### Referensi

[1] A. Karuniawan, A. Santoso, A. Hari Riyadi, and E. Arvin Karuniawan, “Rancang Bangun Trainer Pembelajaran Kendali PID (Proportional Integral Derivative) Kecepatan Motor DC Berbasis PLC dan HMI,” *J. Sains dan Teknol.*, vol. 6, no. 1, pp. 100–104, 2024, [Online]. Available: https://doi.org/10.55338/saintek.v6i1.3177

[2] D. Auditiya Marizka, A. Lastiyan Ashila, A. Agustina Solihah, H. Adi Prasetyo, F. Riza, and R. Wicaksono, “Rancang Bangun Sistem Kendali PID Motor DC Berbasis Blynk untuk meningkatkan Efisiensi dan Fleksibilitas,” *J. Inov. Teknol. Terap.*, vol. 3, no. 2, pp. 277–283, 2025, doi: 10.33504/jitt.v3i2.345.

[3] A. Andreas, G. Priyandoko, M. Mukhsim, and S. A. Putra, “Kendali Kecepatan Motor Pompa Air Dc Menggunakan Pid – Csa Berdasarkan Debit Air Berbasis Arduino,” *JASEE J. Appl. Sci. Electr. Eng.*, vol. 1, no. 01, pp. 1–14, 2020, doi: 10.31328/jasee.v1i01.3.

[4] F. Suryatini, A. S. Sunarya, and Q. K. ‘Amilin, “Optimasi kendali PID kecepatan motor DC dengan metode tuning particle

swarm optimization," *JITEL (Jurnal Ilm. Telekomun. Elektron. dan List. Tenaga)*, vol. 5, no. 1, pp. 57–66, 2025, doi: 10.35313/jitel.v5.i1.2025.57-66.

[5] M. K. Arafat, S.Kom, "SISTEM PENGAMANAN PINTU RUMAH BERBASIS Internet Of Things ( IoT ) Dengan ESP8266," vol. 7, no. 4, pp. 262–268, 2016.

[6] B. Artono and R. G. Putra, "PENERAPAN INTERNET OF THINGS ( IOT ) UNTUK KONTROL LAMPU MENGGUNAKAN ARDUINO BERBASIS WEB," vol. 05, no. 01, pp. 9–16, 2018.

[7] W. Widodo, M. Ruswiensari, and A. Qomar, "Monitoring Pemakaian Daya Listrik Secara Realtime Berbasis Internet Of Things," pp. 581–586.

[8] G. Priyandoko, "Rancang Bangun Sistem Portable Monitoring Infus Berbasis Internet of Things," vol. 3, pp. 56–61, 2021.

[9] R. A. S. Imami, A. Ma'arif, and S. Sunardi, "Position Control of DC Motor using Proportional-Integral-Derivative (PID) Controller Based on Arduino," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 524–528, 2021, doi: 10.18196/jrc.26131.

[10] M. F. Wicaksono, "Implementasi Modul WiFi ESP32-CAM dan ESP32 untuk Sistem Monitoring dan Otomasi pada Industri Logistik," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 2, pp. 85–92, 2021, doi: 10.14710/jtsiskom.2021.13944.

[11] N. S. Sholahuddin, M. Ramdhani, and E. Susanto, "Perancangan Sistem Pengendalian Posisi Motor Servo Menggunakan Kontroler PID-Ziegler Nichols berbasis Mikrokontroler," *eProceeding of Engineering*, vol. 8, no. 5, pp. 4812–4819, 2021.

[12] H. S. Lioe and J. Al Rasyid, "Sistem Monitoring Real-Time Suhu dan Getaran Motor Induksi Lini Produksi Berbasis Protokol MQTT dan Node-RED," *Jurnal Ilmiah Teknik Elektro*, vol. 12, no. 1, pp. 34–42, 2024, doi: 10.30871/jise.v12i1.4121.

[13] K. Ogata, *Modern Control Engineering*, 5th ed. Boston: Prentice Hall, 2010. *(Catatan: Sangat baik digunakan sebagai referensi buku teks utama pada bagian rumus matematis PID di Metode Penelitian).*

[14] A. Fahrurozi, Y. Yudhistira, and B. Suprianto, "Analisis Performa Kontrol Posisi Motor Servo Menggunakan Metode Tuning Trial-Error dan Ziegler-Nichols," *Jurnal Teknik Elektro*, vol. 11, no. 2, pp. 115–123, 2022.

[15] I. G. P. M. E. Putra, I. N. S. Kumara, and I. M. O. Widyantara, "Implementasi Dashboard IoT Menggunakan Platform Blynk untuk Monitoring Parameter Kelistrikan Mesin Industri," *Majalah Ilmiah Teknologi Elektro*, vol. 21, no. 2, pp. 245–252, 2022, doi: 10.24843/MITE.2022.v21i02.P11.

[16] M. S. Prasad, K. R. Kumar, and B. V. S. Rao, "IoT Based Remote Tuning and Monitoring of PID Controller for Industrial Actuators," *International Journal of Automation and Control*, vol. 17, no. 4, pp. 389–405, 2023, doi: 10.1504/IJAAC.2023.131200.