

Sistem Kontrol Kecepatan Motor Berbasis ESP32 dengan Feedback Encoder, Kontrol PID, dan Komunikasi

MQTT*Adiel Fatich Ar Rafi, Dr Basuki Rahmat, Penulis 3 (Penomoran Penulis Ketika

Beda Afiliasi)

Universitas Pembangunan Nasional "Veteran" Jawa Timur, Informatika, Fakultas Ilmu
Komputer, Indonesia

Diterima: 26 Desember, 2025 | Revisi: 11 Mei, 2024 | Diterbitkan: 11 Juni 2025

DOI:

ABSTRAK

Penelitian ini mengimplementasikan sistem kontrol kecepatan motor berbasis ESP32 dengan integrasi sensor encoder, algoritma PID anti-windup, dan komunikasi MQTT. Kebaruan penelitian terletak pada kombinasi kontrol klasik dengan infrastruktur IoT yang reliable. Metode penelitian meliputi akuisisi feedback real-time melalui interrupt-driven pulse counting, filtering exponential moving average, serta pengujian komunikasi MQTT dengan broker HiveMQ. Hasil menunjukkan sistem mencapai stabilisasi kecepatan dengan settling time <30 detik, overshoot terkontrol 12%, dan latency komunikasi <2 detik. Kesimpulan penelitian menegaskan bahwa ESP32 merupakan solusi cost-effective untuk embedded control systems dengan kemampuan integrasi jaringan yang baik.

Kata Kunci: PID, encoder, MQTT, IoT, ESP32

Title Written in Initial Capital Only, Maximum 15 Words

How to Cite:

Nasrulloh, M. F., Faidah, R. S. N., Maksum, M. J. S., & Arifin, M. Z. (2023). Pengembangan LKPD Menggunakan Pembelajaran Core pada Materi Statistika SMP. *Scan : Jurnal Teknologi Informasi dan Komunikasi*, 1(1), 1-5.

*Corresponding Author:

Email : xxxx@gmail.com
Alamat : Alamat Afiliasi



This article is published under a Creative Commons Attribution 4.0 International License.

PENDAHULUAN

Kontrol kecepatan motor merupakan salah satu aspek penting dalam sistem otomasi industri, robotika, dan perangkat rumah pintar. Permintaan terhadap sistem yang akurat, responsif, dan dapat dioperasikan jarak jauh semakin meningkat seiring perkembangan teknologi Internet of Things (IoT). Penelitian sebelumnya menunjukkan bahwa algoritma PID mampu memberikan performa yang baik dalam menjaga kestabilan kecepatan motor, namun masih terdapat tantangan berupa overshoot tinggi dan keterbatasan komunikasi real-time.

ESP32 sebagai platform embedded menawarkan prosesor dual-core, konektivitas WiFi, serta GPIO fleksibel yang menjadikannya solusi cost-effective untuk implementasi sistem kontrol modern. Beberapa studi terdahulu telah menguji PID pada motor DC dengan hasil memuaskan, tetapi integrasi dengan protokol komunikasi MQTT dan mekanisme anti-windup masih jarang dibahas secara komprehensif. Celaah penelitian ini menjadi motivasi utama untuk mengembangkan sistem kontrol motor berbasis ESP32 yang tidak hanya stabil secara lokal, tetapi juga dapat dipantau dan dikendalikan secara remote melalui infrastruktur IoT.

Rumusan masalah dalam penelitian ini mencakup: (1) bagaimana mengimplementasikan akuisisi feedback encoder secara real-time dengan overhead CPU minimal; (2) bagaimana algoritma PID dengan anti-windup dapat mengurangi overshoot dan steady-state error; (3) bagaimana komunikasi MQTT dapat mendukung kontrol jarak jauh dengan latency rendah; dan (4) bagaimana sistem ini dapat diaplikasikan pada berbagai domain seperti smart home, industri kecil, dan robotika.

Tujuan penelitian adalah mendemonstrasikan integrasi sensor encoder, kontrol PID dengan anti-windup, dan komunikasi MQTT pada platform ESP32, serta menganalisis performa sistem dari aspek stabilisasi kecepatan, reduksi noise, dan reliabilitas komunikasi. Dengan demikian, penelitian ini diharapkan memberikan kontribusi praktis dalam penerapan teori kontrol klasik pada embedded systems modern sekaligus memperluas literatur mengenai integrasi IoT dalam sistem kontrol motor.

METODE PENELITIAN

Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental berbasis sistem embedded. ESP32 dipilih sebagai platform utama karena memiliki prosesor dual-core, konektivitas WiFi, serta peripheral PWM yang fleksibel. Pemilihan ESP32 juga didasarkan pada pertimbangan biaya rendah dan ketersediaan luas, sehingga relevan untuk aplikasi industri kecil maupun otomasi rumah. Desain penelitian ini bertujuan untuk menguji integrasi tiga komponen utama: akuisisi feedback encoder, algoritma PID dengan anti-windup, dan komunikasi MQTT.

Pengambilan Sampel dan Konteks

Unit analisis berupa motor DC dengan encoder optik 1 PPR. Motor ini dipilih karena karakteristiknya sederhana namun cukup representatif untuk aplikasi kontrol kecepatan. Konteks penelitian diarahkan pada simulasi aplikasi industri ringan, seperti conveyor

atau pompa, serta otomasi rumah seperti kipas atau sistem irigasi. Sampel berupa satu motor DC diuji dalam empat mode operasi (OFF, LOW, MEDIUM, HIGH) untuk menilai konsistensi performa sistem.

Pengumpulan Data

Data kecepatan motor dikumpulkan melalui metode interrupt-driven pulse counting. Setiap pulsa dari encoder memicu interrupt service routine (ISR) yang menambah counter global. Interval sampling ditetapkan 2 detik, sehingga jumlah pulsa per interval dapat dikonversi menjadi RPM. Untuk mengurangi noise akibat jitter mekanis, diterapkan filter exponential moving average (EMA) dengan smoothing constant 0.5. Data hasil pengukuran kemudian dikirim melalui protokol MQTT ke broker HiveMQ untuk monitoring jarak jauh.

Pengukuran Variabel

Variabel utama yang diukur meliputi:

- **Kecepatan motor (RPM):** dihitung dengan persamaan

$$RPM = \frac{N \times 60}{T}$$

di mana N adalah jumlah pulsa dalam interval waktu T .

- **Overshoot:** dihitung sebagai persentase selisih maksimum terhadap setpoint.
- **Settling time:** waktu yang dibutuhkan sistem untuk mencapai $\pm 5\%$ dari nilai target.
- **Steady-state error:** selisih antara nilai setpoint dan kondisi stabil.
- **Latency komunikasi MQTT:** selisih waktu antara pengiriman perintah kontrol dan penerimaan data feedback.

Analisis Data

Analisis dilakukan secara deskriptif kuantitatif. Nilai rata-rata, deviasi standar, dan persentase error digunakan untuk menilai performa sistem. Perbandingan dilakukan antara mode operasi untuk melihat konsistensi sistem, serta antara kondisi dengan dan tanpa anti-windup untuk menilai efektivitas algoritma. Analisis juga mencakup evaluasi filter EMA terhadap noise pengukuran, serta reliabilitas komunikasi MQTT dalam jangka panjang.

Alasan Pemilihan Metode

Metode interrupt-driven dipilih karena lebih efisien dibanding polling, dengan overhead CPU rendah. PID dipilih karena merupakan algoritma kontrol klasik yang terbukti efektif, sementara mekanisme anti-windup ditambahkan untuk mengatasi masalah saturasi aktuator. Filter EMA dipilih karena sederhana namun efektif dalam mereduksi noise tanpa menambah kompleksitas komputasi. MQTT dipilih karena protokol ini ringan, efisien, dan sesuai untuk perangkat embedded dengan keterbatasan sumber daya.

HASIL DAN PEMBAHASAN

Performa Sistem Kontrol

Pengujian sistem menunjukkan bahwa motor mampu mencapai kecepatan target pada setiap mode operasi. Mode LOW (800 RPM) dan MEDIUM (900 RPM) relatif cepat mencapai kestabilan, dengan settling time kurang dari 20 detik. Mode HIGH (2000 RPM) membutuhkan waktu lebih lama, sekitar 30 detik, karena beban transien lebih besar saat motor berakselerasi menuju kecepatan tinggi. Hal ini sesuai dengan teori kontrol klasik, di mana semakin besar setpoint, semakin besar pula energi yang dibutuhkan untuk mencapai kondisi stabil.

Efektivitas PID Anti-Windup

Implementasi PID dengan mekanisme anti-windup terbukti menekan overshoot secara signifikan. Tanpa anti-windup, overshoot dapat mencapai lebih dari 25%, sedangkan dengan anti-windup overshoot hanya berkisar 10–15%. Temuan ini konsisten dengan penelitian [11][12][13] yang menekankan pentingnya anti-windup dalam sistem dengan saturasi aktuator. Steady-state error juga berhasil diminimalkan hingga <2%, menunjukkan bahwa integrasi PID dengan anti-windup memberikan kontribusi nyata terhadap kualitas kontrol.

Pengukuran Encoder dan Filtering

Data mentah dari encoder menunjukkan fluktuasi ± 50 RPM akibat noise mekanis dan jitter sinyal. Penerapan filter exponential moving average (EMA) dengan smoothing constant 0.5 mampu mengurangi noise hingga 60% tanpa menimbulkan lag signifikan. Respon sistem tetap cepat, dengan keterlambatan kontrol kurang dari 4 detik. Hasil ini sejalan dengan literatur [14][15][16] yang menyatakan bahwa filtering digital sederhana dapat meningkatkan akurasi pengukuran tanpa membebani komputasi.

Komunikasi MQTT

Pengujian komunikasi MQTT menunjukkan latency rata-rata 500–800 ms, dengan kondisi terburuk mencapai 2 detik. Latency ini masih dapat diterima untuk aplikasi non-kritis seperti smart home atau monitoring industri ringan, tetapi kurang sesuai untuk aplikasi real-time kritis. Pengujian jangka panjang selama lebih dari 24 jam menunjukkan sistem mampu mempertahankan koneksi stabil, dengan mekanisme auto-reconnect yang efektif. Temuan ini mendukung penelitian [19][20][21] yang menekankan keandalan MQTT dalam aplikasi IoT dengan perangkat terbatas.

Efisiensi Sumber Daya

Loop kontrol real-time, yang mencakup akuisisi data encoder, perhitungan PID, dan output PWM, dapat diselesaikan dalam 50–100 ms. Hal ini menghasilkan sampling rate lebih dari 10 Hz, cukup untuk aplikasi kontrol motor sederhana. Utilisasi CPU tercatat kurang dari 40% pada core utama, sehingga masih tersedia kapasitas untuk ekspansi sistem. Kondisi ini menunjukkan bahwa ESP32 merupakan platform yang efisien dan fleksibel untuk pengembangan lebih lanjut.

Diskusi dan Analisis

Secara keseluruhan, hasil penelitian ini menegaskan bahwa sistem kontrol berbasis ESP32 dengan PID anti-windup dan komunikasi MQTT mampu memberikan performa yang stabil, akurat, dan dapat diandalkan. Dibandingkan dengan metode polling tradisional, pendekatan interrupt-driven lebih efisien dalam penggunaan sumber daya. Integrasi filter EMA juga memberikan peningkatan kualitas data tanpa mengorbankan kecepatan respon. Dengan demikian, sistem ini layak diaplikasikan pada berbagai bidang seperti otomasi rumah, industri kecil, robotika, maupun pertanian berbasis IoT.

Tabel dan Gambar

Penulis harus memberikan karya seni berkualitas tinggi untuk semua ilustrasi. Resolusi atau definisi yang buruk tidak dapat diterima. Semua tabel dan gambar (grafik) harus disertakan langsung di bagian artikel yang sesuai (tidak dipisahkan). Setiap tabel atau gambar harus diberi nomor secara terpisah (Tabel 1, Tabel 2; Gambar 1, Gambar 2) dan diberi judul lengkap, yang mengacu pada isi tabel atau gambar. Di bawah setiap tabel dan gambar, sebutkan sumber tepercaya. Jangan Menyajikan hasil output alat analisis secara langsung, melainkan sajikan tabel atau gambar yang akan dibahas[4].

Tabel 1
Nama Tabel....

Kolom 1	Kolom 2	Kolom 2	Kolom 2
Abcde1	0.xxx	0.xxx1	0.xxx2
Abcde2	0.yyyy	0.yyyy1	0.yyyy2
Abcde3	0.zzz	0.zzz1	0.zzz2
Abcde4	0.aaaa	0.aaaa1	0.aaaa2

Sumber:



Gambar 1. Nama Gambar.....

Sumber:

SIMPULAN

Penelitian ini berhasil mengimplementasikan sistem kontrol kecepatan motor berbasis ESP32 dengan integrasi encoder feedback, algoritma PID anti-windup, dan komunikasi MQTT. Hasil pengujian menunjukkan bahwa sistem mampu mencapai target RPM dengan settling time optimal, overshoot terkontrol, serta steady-state error yang rendah. Filtering exponential moving average terbukti efektif dalam mereduksi noise pengukuran, sementara protokol MQTT memberikan komunikasi jarak jauh yang stabil dengan latency rata-rata di bawah 1 detik.

Secara empiris, penelitian ini memberikan bukti bahwa ESP32 dapat digunakan sebagai solusi cost-effective untuk embedded control systems dengan kemampuan integrasi IoT. Secara teoritis, penelitian ini memperkuat literatur mengenai penerapan PID anti-windup dan filtering digital sederhana pada sistem kontrol motor. Dari sisi ekonomi, sistem ini menawarkan alternatif murah dan fleksibel untuk aplikasi otomasi rumah, industri kecil, dan robotika.

Keterbatasan penelitian terletak pada latency komunikasi MQTT yang masih mencapai 2 detik dalam kondisi terburuk, sehingga kurang sesuai untuk aplikasi real-time kritis. Selain itu, sistem hanya menggunakan feedback RPM tanpa sensor tambahan untuk deteksi kondisi abnormal seperti stall. Penelitian lanjutan dapat diarahkan pada pengembangan self-tuning PID, sinkronisasi multi-motor, serta peningkatan keamanan komunikasi dengan TLS.

DAFTAR PUSTAKA

- [1] IEEE Explore, *Design and Implementation of a Verilog Based PID Controller for DC Motor Speed Control*, 2025.
- [2] IEEE Explore, *Three-Motor Synchronous Speed-Regulation System Based on RBF Neural Network Control*, 2010.
- [3] IEEE Explore, *BLDC Motor Speed Control with Digital Adaptive PID-Fuzzy Controller and Reduced Harmonic Content*, 2024.
- [4] MDPI Electronics, *Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller*, 2022.
- [5] Last Minute Engineers, *In-Depth: Generating a PWM Signal on the ESP32*, 2025.
- [6] Journal of Food Science and Culinary Technology, *DC Motor Speed Control with Proportional Integral Derivative (PID) Control on the Prototype of a Mini-Submarine*, 2023.
- [7] MDPI Sensors, *Optimal PID Control of a Brushed DC Motor with an Embedded Low-Cost Magnetic Quadrature Encoder for Improved Step Overshoot and Undershoot Responses*, 2022.
- [8] IEEE Explore, *Comparative Analysis of PID, Self-Tuning PID, and Adaptive Neuro-Fuzzy Logic Inference System Controllers for BLDC Motor Speed Control*, 2024.
- [9] IEEE Explore, *Adaptive PID Control for Autonomous Vehicle Motor Regulation: Design and Simulation Using MATLAB*, 2024.
- [10] Journal of International Science and Technology Education, *DC Motor Angular Speed Controller Using an Embedded Microcontroller-Based PID Controller*, 2025.
- [11] Dergi Park, *Anti-Windup PID Control for Globe Valve Internal Disturbances: Design and Simulation Approach*, 2024.
- [12] Springer Link, *Gain-scheduled anti-windup PID control for LPV systems under actuator saturation and its application to aircraft*, 2022.
- [13] Telkom University, *A Method of Anti-Windup PID Controller for a BLDC-Drive*, 2024.
- [14] Deep Blue Embedded, *Arduino RPM Sensor (RPM Meter/Counter With Encoder)*, 2025.
- [15] Motion Control Tips, *How are encoders used for speed measurement?*, 2022.
- [16] Global Tech, *What type of encoder can measure speed?*, 2021.

[17] All PCB - AllElectroHub, *Concurrency and Interrupts in Microcontrollers and Embedded Systems*, 2025.

[18] Arduino Kit Project, *Exploring ESP32 PWM: Your Complete Starter's Guide*, 2024.

[19] IEEE Explore, *MQTT2EdgePeer: a Robust and Scalable Brokerless Peer-to-Peer Edge Middleware for Topic-Based Publish/Subscribe*, 2024.

[20] MDPI Applied Sciences, *A Reliable Publish–Subscribe Mechanism for Internet of Things-Enabled Smart Greenhouses*, 2024.

[21] MDPI Sensors, *Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions*, 2023.

[22] Steve's Internet Guide, *Using the Arduino PubSub MQTT Client*, 2024.

[23] HiveMQ Blog, *Arduino PubSubClient - MQTT Client Library Encyclopedia*, 2015.

[24] Random Nerd Tutorials, *ESP32 MQTT Publish Subscribe with Arduino IDE*, 2020.

[25] Random Nerd Tutorials, *ESP32 PWM with Arduino IDE (Analog Output)*, 2024.

[26] Arduino Kit Project, *Exploring ESP32 PWM: Your Complete Starter's Guide to PWM on the ESP32*, 2024.

[27] Last Minute Engineers, *In-Depth: Generating a PWM Signal on the ESP32*, 2025.