JURNAL LITEK : Jurnal Listrik Telekomunikasi Elektronika
Vol. 23, No. 1, Maret 2026, pp. 8 – 16.
pISSN: 1693-8097; eISSN: 2549-8762.
Akreditasi Sinta 6, SK : 177/E/KPT/2024. DOI : https://doi.org/10.30811/litek.v23i1.96.

KENDALI DIGITAL DUAL-MODE BERBASIS ESP32 UNTUK PENGENDALIAN POSISI DAN KECEPATAN MOTOR DC

Ryan Hidayat[1], Muhammad Basyir[2], Muhammad Kamal[3], Arsy Febrina Dewi[4], Muhammad Ramzil Akbar[5]

[1,2,3,4,5)]Jurusan Teknik Elektro, Politeknik Negeri Lhokseumawe
Email: ryanhidayat@gmail.com[1], m.basyir@pnl.ac.id[2], muhakam61@yahoo.com[3], arsyfebrinadw@pnl.ac.id[4], ramzil.akbar@pnl.ac.id[5]

Corresponding Author: Muhammad Basyir

Jurusan Teknik Elektro, Politeknik Negeri Lhokseumawe
Email: m.basyir@pnl.ac.id

Abstrak – Penelitian ini membahas perancangan dan implementasi sistem kendali PID digital berbasis mikrokontroler ESP32 untuk pengendalian posisi dan kecepatan motor DC. Sistem dikembangkan untuk menggantikan modul kendali analog pada Laboratorium Sistem Kendali konvensional, dengan tujuan meningkatkan akurasi, efisiensi, dan fleksibilitas proses penalaan parameter. Pemrograman dilakukan menggunakan Arduino IDE (C/C++) dengan integrasi antarmuka MATLAB GUI yang berfungsi sebagai media pemantauan respon sistem dan *real-time tuning* parameter Kp, Ki, dan Kd. Hasil pengujian menunjukkan bahwa sistem mampu mencapai *settling time* rata-rata 1,81s, *overshoot* maksimum 9,3%, dan *steady-state error* di bawah 3% untuk mode kendali posisi maupun kecepatan. Dibandingkan dengan sistem analog, desain digital berbasis ESP32 memberikan peningkatan kinerja sebesar ±38% dalam hal waktu respon dan kestabilan. Dengan demikian, sistem ini dapat digunakan sebagai platform edukatif untuk praktikum sistem kendali digital sekaligus prototipe terapan untuk sistem kendali berbasis IoT dan *cyber–physical systems* (CPS).

Kata kunci: PID Digital, ESP32, Motor DC, MATLAB GUI, Real-Time Control.

Abstract *– This study presents the design and implementation of a digital PID control system based on the ESP32 microcontroller for DC motor position and speed control. The system was developed to replace conventional Analog Control Modules in Laboratory environments, aiming to improve accuracy, efficiency, and flexibility in parameter tuning. Programming was carried out using Arduino IDE (C/C++), integrated with a MATLAB GUI for real-time monitoring and parameter adjustment of Kp, Ki, and Kd. Experimental results demonstrate that the system achieves an average settling time of 1.81 seconds, a maximum overshoot of 9.3%, and a steady-state error below 3% in both position and speed control modes. Compared to the analog PID system, the ESP32-based digital design improves control performance by approximately 38% in response speed and stability. Therefore, this system can serve as an educational laboratory platform for digital control systems as well as a prototype for IoT-based and cyber–physical control applications.*

Keywords: Digital PID, ESP32, DC Motor, MATLAB GUI, Real-Time Control.

I. PENDAHULUAN

Kendali *Proportional Integral Derivative* (PID) merupakan salah satu algoritma kontrol klasik yang paling banyak digunakan di berbagai sistem industri maupun pendidikan karena strukturnya yang sederhana, mudah diimplementasikan, serta mampu memberikan respon sistem yang stabil dan akurat [1][2]. Meskipun telah dikembangkan sejak beberapa dekade lalu, algoritma PID tetap relevan hingga saat ini berkat kemampuannya beradaptasi terhadap perubahan dinamika sistem dan kemajuan teknologi kontrol. Seiring berkembangnya teknologi otomasi dan sistem tertanam (*embedded systems*), implementasi PID mulai beralih dari sistem analog ke digital, yang menawarkan presisi tinggi, efisiensi komputasi, dan kemudahan integrasi dengan perangkat lunak modern maupun *platform Internet of Things* (IoT) [2]. Secara tradisional, PID analog dibangun menggunakan rangkaian *operational amplifier (Op-Amp)* dan potensiometer sebagai pengatur parameter Kp, Ki, dan Kd. Namun, pendekatan ini memiliki berbagai keterbatasan, seperti kesulitan penyetelan (tuning) yang presisi, ketergantungan terhadap stabilitas komponen pasif, serta keterbatasan fleksibilitas ketika terjadi perubahan kondisi sistem. Sebaliknya, PID digital memungkinkan penyetelan

Diserahkan : 14 Oktober 2025 | Diperbaiki : 18 November 2025 | Diterima : 20 November 2025 | Dipublikasi : 1 Maret 2026.

parameter secara cepat dan berulang (*repeatable*), dapat dikonfigurasi melalui perangkat lunak, serta mudah diintegrasikan dengan sistem pemantauan *real time* dan antarmuka grafis seperti MATLAB atau *Python* [3].

Penelitian terbaru telah menunjukkan keunggulan sistem PID digital dibandingkan analog dalam berbagai aplikasi kontrol. Le Thai dan Thi Kieu [4] mengimplementasikan kendali PID berbasis mikrokontroler STM32F407 secara *real time* untuk motor DC, yang menghasilkan kestabilan kecepatan lebih baik dibandingkan kontrol konvensional. Ma'arif et al. [5] juga berhasil menerapkan PID digital berbasis Arduino untuk pengendalian motor DC dan memperoleh waktu naik (*rise time*) serta kesalahan tunak (*steady-state error*) yang lebih kecil dibandingkan sistem analog.

Perkembangan teknologi mikrokontroler modern, seperti ESP32, turut mendorong kemajuan sistem kendali digital berbiaya rendah namun bertenaga tinggi. ESP32 memiliki prosesor *dual-core* 240 MHz, *analog-to-digital converter* (ADC) 12-bit, serta konektivitas Wi-Fi dan *Bluetooth* yang terintegrasi. Fitur ini memungkinkan sistem kendali melakukan akuisisi data secara *real time*, komunikasi serial dua arah, dan visualisasi respon sistem melalui perangkat lunak eksternal seperti MATLAB [6]. Dengan demikian, ESP32 sangat ideal untuk pengembangan sistem kendali digital baik di lingkungan riset maupun pendidikan.

Dalam konteks pendidikan teknik, transisi dari modul praktikum PID analog ke PID digital menjadi penting untuk meningkatkan efektivitas pembelajaran dan validitas hasil eksperimen. Hudaya et al. [7] melaporkan bahwa sistem tertanam berbasis PID digital dengan metode pemrosesan terdistribusi mampu memberikan respon lebih cepat dan akurat dibandingkan sistem analog. Selain itu, Elgeme et al. [8] menunjukkan bahwa penerapan kendali PID digital pada mesin CNC *Plotter* berbiaya rendah berhasil menurunkan kesalahan posisi hingga 75% dibandingkan versi analog.

Optimalisasi parameter PID digital juga menjadi perhatian utama dalam penelitian terkini. Afram, Hussien, dan Marie [9] merancang sistem kendali PID optimal berbasis PLC pada sistem *Electrostatic Precipitator* (ESP) yang mampu meningkatkan efisiensi energi melalui metode penyetelan otomatis (*auto tuning*). Al-Baidhani dan Kazimierczuk [10] mengembangkan sistem PID digital untuk model massa peredam (*mass damper system*) dan membuktikan bahwa penyetelan parameter yang tepat dapat memperbaiki respon transien serta menurunkan kesalahan tunak secara signifikan.

Namun demikian, sebagian besar penelitian sebelumnya hanya berfokus pada satu mode pengendalian baik posisi atau kecepatan tanpa mengintegrasikan sistem *dual-mode* yang dapat digunakan secara fleksibel di laboratorium pendidikan. Selain itu, masih terbatas penelitian yang menggabungkan optimasi parameter PID digital dengan antarmuka visualisasi data *real-time* untuk keperluan pembelajaran dan eksperimen.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengimplementasikan dan mengoptimasi kendali PID digital berbasis mikrokontroler ESP32 sebagai pengganti sistem PID analog pada Modul Praktikum Laboratorium Sistem Kendali. Sistem ini memungkinkan pengaturan parameter Kp, Ki, dan Kd secara *real time* menggunakan tombol tekan (*push button*) serta integrasi dengan MATLAB untuk menampilkan respon umpan balik (*feedback*) dan keluaran (*output*) sistem secara langsung.

Kontribusi utama penelitian ini adalah sebagai berikut: merancang dan merealisasikan sistem kendali PID digital berbasis ESP32 untuk pengendalian posisi dan kecepatan (*dual mode*); Melakukan perbandingan kinerja antara sistem PID analog dan digital dari sisi waktu naik, *overshoot*, dan kesalahan tunak; serta melakukan optimasi parameter PID digital untuk meningkatkan kestabilan dan performa sistem agar sesuai dengan kebutuhan pembelajaran sistem kendali modern.

II. TINJAUAN PUSTAKA

A. Dasar Kendali PID

Kendali PID merupakan salah satu metode pengendalian umpan balik yang paling populer dalam sistem otomasi industri maupun pendidikan. Struktur dasar PID terdiri dari tiga komponen utama, yaitu *proportional* untuk mengoreksi kesalahan secara langsung, *integral* untuk mengeliminasi kesalahan tunak (*steady state error*), dan *derivative* untuk memperkirakan tren kesalahan di masa depan. Kombinasi ketiga aksi ini menghasilkan sistem yang responsif dan stabil terhadap berbagai gangguan eksternal [11].

Sundström et al. [11] menjelaskan bahwa implementasi referensi PID modern dapat dilakukan dengan pendekatan arsitektur yang fleksibel, di mana parameter Kp, Ki, dan Kd dapat disesuaikan secara adaptif untuk menjaga stabilitas sistem di bawah berbagai kondisi operasi. Selain itu, studi oleh Ounis et al. [12] mengembangkan *Fractional Order* PID (FOPID) yang dapat disetel secara analog maupun digital secara *real-time*, memberikan tingkat fleksibilitas yang lebih tinggi dibandingkan PID konvensional.

B. Digitalisasi dan Implementasi Hardware PID

Transformasi dari sistem analog ke digital menjadi salah satu fokus utama dalam rekayasa kendali modern. Digitalisasi memberikan berbagai keuntungan seperti ketepatan pengolahan sinyal, kemampuan penyimpanan parameter, serta integrasi dengan platform berbasis mikroprosesor dan Internet of Things (IoT) [13].

Zhu dan Liu [13] menjelaskan bahwa integrasi rangkaian analog dengan sistem digital dapat dilakukan melalui teknik *digitalized analog circuits*, yang memungkinkan sistem kontrol klasik dapat diimplementasikan dalam bentuk digital dengan presisi yang lebih baik. Penelitian oleh Thomas dan Srinivasan [14] juga menunjukkan bahwa penerapan PID pada *embedded platform* secara digital mampu memberikan

waktu respon yang lebih cepat dan akurasi yang lebih tinggi untuk aplikasi *real time*. Implementasi *hardware* PID digital terus berkembang seiring kemajuan prosesor dan mikrokontroler. Jokiel [15] memperkenalkan algoritma PID berbasis *floating-point arithmetic*, yang memberikan ketelitian tinggi terhadap perhitungan *error* dibandingkan metode berbasis *integer arithmetic*. Ma'arif et al. [16] juga menambahkan bahwa pengolahan arus DC secara langsung melalui metode puncak (*peak value method*) pada Arduino dapat menghasilkan respon sistem yang lebih linier. Selain itu, Podržaj [17] mengimplementasikan kendali PID digital pada perangkat *Festo* CDPX, yang menunjukkan kestabilan respon sistem kendali dengan waktu naik yang lebih singkat dibandingkan metode analog konvensional. Hal ini memperkuat bukti bahwa sistem kendali digital lebih efisien, mudah diadaptasi, dan sangat potensial untuk diterapkan di lingkungan industri maupun pendidikan.

C. Metode Penyetelan dan Optimasi Parameter PID

Penyetelan parameter PID (Kp, Ki, dan Kd) merupakan aspek yang krusial untuk mencapai keseimbangan antara kecepatan respon dan kestabilan sistem. Penelitian oleh Idir et al. [2] membandingkan tiga pendekatan, yaitu *Integer Order* PID (IOPID)**,** *Fractionalized* PID (FPID)**,** dan *Fractional Order* PID (FOPID)**,** yang menunjukkan bahwa FOPID mampu memberikan respon yang lebih halus dengan *overshoot* lebih rendah pada sistem stabil.

Lebih lanjut, Chen[8] menyoroti tren masa depan sistem kendali PID yang mengintegrasikan optimasi berbasis kecerdasan buatan (*AI-based optimization*) dan pembelajaran mesin (*machine learning*) untuk melakukan *auto-tuning* parameter PID secara adaptif. Dengan pendekatan ini, sistem mampu menyesuaikan parameter kontrol secara dinamis tanpa memerlukan intervensi manual.

Maravi et al. [18] juga menunjukkan bahwa PID digital yang dioptimasi untuk kompensasi *loss steps* pada printer 3D dengan mekanisme *Core* XY mampu mengoreksi deviasi posisi pada kecepatan tinggi, dengan peningkatan akurasi hingga 20% dibandingkan kontrol konvensional. Temuan ini menegaskan pentingnya strategi tuning digital dalam sistem yang membutuhkan presisi gerakan tinggi.

D. Aplikasi Terkini Sistem PID Digital

Implementasi PID digital kini meluas pada berbagai bidang, mulai dari sistem robotik, kendaraan otonom, hingga sistem produksi cerdas (*smart manufacturing*). Ghassoul [1] merancang kendali tiga-elemen berbasis mikrokontroler PIC18F452 untuk aplikasi energi terbarukan, yang menunjukkan efisiensi pengendalian arus dan tegangan dengan konsumsi daya yang rendah.

Penelitian oleh Ounis et al. [12] dan Sundström et al. [11] memperkuat arah pengembangan sistem PID menuju konfigurasi modular dan adaptif, di mana kontroler dapat diatur melalui antarmuka perangkat lunak tanpa perlu perubahan perangkat keras. Pendekatan ini menjadi semakin relevan dengan perkembangan sistem kendali berbasis *Software-Defined Control* (SDC) dan teknologi *Cyber–Physical Systems* (CPS).

Selain pada aplikasi industri, pendekatan PID digital juga mulai banyak diadopsi dalam bidang pendidikan teknik. Hudaya et al. [5] menunjukkan bahwa sistem tertanam berbasis PID digital sangat efektif dalam meningkatkan pemahaman mahasiswa terhadap dinamika sistem kontrol, karena memberikan hasil eksperimen yang dapat dimonitor, diubah, dan dianalisis secara langsung.

E. Analisis Kesenjangan Penelitian (*Research Gap*)

Berdasarkan hasil kajian literatur, dapat disimpulkan bahwa mayoritas penelitian sebelumnya lebih menitikberatkan pada implementasi PID digital untuk sistem tunggal (*single mode*), seperti kendali posisi atau kecepatan secara terpisah [14][16][18]. Selain itu, penelitian tentang integrasi antara sistem PID digital dan antarmuka perangkat lunak MATLAB secara *real-time* masih relatif terbatas, terutama pada skala laboratorium pendidikan teknik.

Untuk itu, penelitian ini berkontribusi dalam mengisi celah tersebut dengan merancang sistem PID digital berbasis ESP32 yang tidak hanya mengontrol posisi tetapi juga kecepatan dalam satu modul (*dual mode control system*). Sistem yang dikembangkan juga dilengkapi dengan kemampuan visualisasi *real time* melalui komunikasi serial dengan MATLAB serta penyetelan parameter digital melalui antarmuka pengguna yang mudah digunakan. Pendekatan ini diharapkan dapat menjadi model pengajaran modern dalam bidang sistem kendali digital sekaligus solusi efektif dalam meningkatkan efisiensi dan fleksibilitas eksperimen di laboratorium pendidikan teknik.

III. METODOLOGI

Penelitian ini merupakan studi eksperimental terapan yang bertujuan untuk merancang, mengimplementasikan, dan mengoptimasi sistem kendali PID digital berbasis mikrokontroler ESP32 untuk pengendalian posisi dan kecepatan motor DC secara simultan (*dual mode control system*). Pendekatan penelitian dilakukan melalui tiga tahapan utama, yaitu perancangan sistem perangkat keras (*hardware design*) sebagai sistem kendali fisik, pengembangan sistem perangkat lunak (*software design*) untuk algoritma PID digital dan komunikasi data, serta pengujian dan analisis performa sistem untuk membandingkan kinerja PID digital terhadap sistem analog konvensional.

Struktur penelitian ini mengikuti pendekatan sistematis yang sejalan dengan metode eksperimental kontrol digital yang digunakan oleh Bhandari dan Csurcsia [3], yang menekankan pentingnya kejelasan dalam proses *sampling*, *feedback*, dan *loop execution*. Selain itu, prinsip digitalisasi kontrol juga mengikuti pendekatan dari Zhu dan Liu [13], yang menunjukkan

bahwa integrasi antara sistem analog dan digital dapat meningkatkan akurasi kendali melalui pemrosesan terprogram.

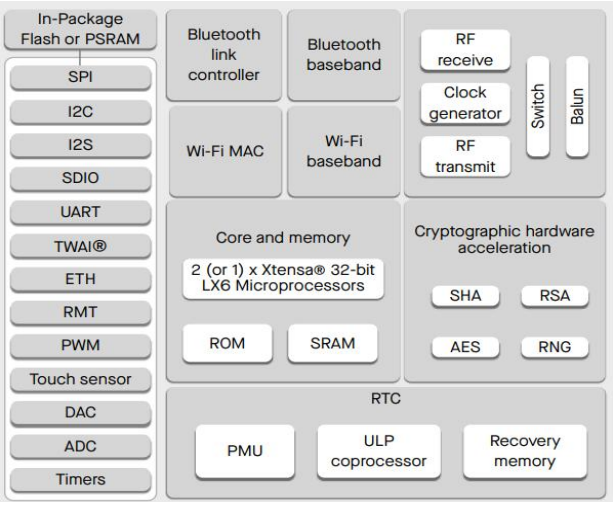
A. Perangkat Keras Sistem

Perangkat keras sistem kendali PID digital ini terdiri dari beberapa komponen utama yang berperan dalam proses akuisisi data, pemrosesan sinyal, dan aktuasi motor. Sistem ini dirancang untuk dapat bekerja secara fleksibel dalam dua mode, yaitu mode kendali posisi dan mode kendali kecepatan, dengan konfigurasi berbasis mikrokontroler ESP32 sebagai pusat kendali utama.

1) Mikrokontroler ESP32

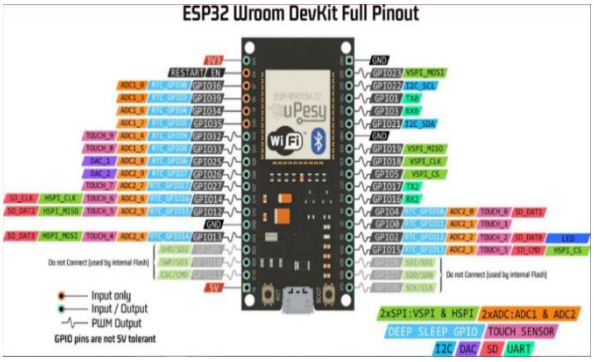
Unit pengendali utama menggunakan ESP32-WROOM-32, yang memiliki prosesor dual-core Xtensa LX6 240 MHz, ADC 12-bit, DAC internal, serta konektivitas Wi-Fi dan Bluetooth. Mikrokontroler ini memiliki kecepatan pemrosesan tinggi yang memungkinkan penerapan algoritma PID digital dengan waktu *sampling* yang konstan (10 ms) [13]. ESP32 juga mendukung komunikasi serial dengan MATLAB untuk proses pemantauan dan tuning parameter secara *real-time*, sebagaimana telah diterapkan pada sistem kendali digital berbiaya rendah oleh Hudaya R. et al. [5].

Gambar 1 memperlihatkan diagram blok fungsional ESP32 yang menggambarkan komponen internal seperti CPU, modul ADC/DAC, memori flash, dan antarmuka komunikasi serial.



Gbr. 1 Diagram Blok Fungsional Mikrokontroler ESP32

Selanjutnya, Gambar 2 menunjukkan tampilan pinout ESP32-WROOM DevKit, yang digunakan untuk menghubungkan sensor, driver motor, dan antarmuka komunikasi. Pin GPIO 32–36 digunakan untuk pembacaan sinyal analog dari potensiometer, sedangkan pin PWM (GPIO 25–27) digunakan untuk mengontrol kecepatan dan arah putaran motor DC melalui driver L298N [3].



Gbr. 2 ESP32-WROOM

2) Motor DC dan Driver

Sistem aktuator menggunakan motor DC 12V dengan sensor *rotary encoder* 600 PPR, yang dikendalikan oleh driver motor L298N *dual H-bridge*. Driver ini berfungsi sebagai penguat arus (*current amplifier*) yang menerjemahkan sinyal PWM dari ESP32 menjadi tegangan dan arus yang cukup besar untuk menggerakkan motor.

Koneksi antara ESP32 dan L298N dilakukan melalui empat pin digital (IN1–IN4), sedangkan sinyal PWM dihasilkan oleh pin GPIO 26 dan 27. Dengan konfigurasi ini, sistem dapat mengatur kecepatan maupun arah rotasi motor secara presisi. Pendekatan penggunaan driver L298N untuk pengendalian motor DC juga digunakan oleh Thomas dan Srinivasan [14], yang membuktikan efektivitasnya dalam aplikasi kontrol *real-time* berbasis platform tertanam.

3) Sensor Posisi dan Umpan Balik

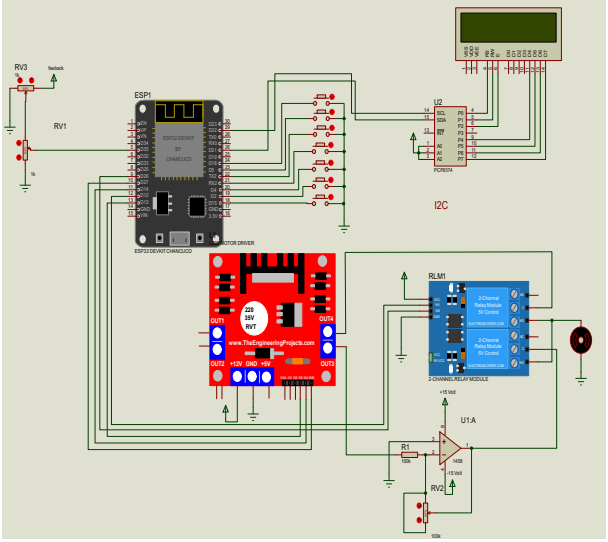
Untuk pengukuran posisi, digunakan potensiometer presisi 10 kΩ yang dihubungkan ke input ADC ESP32. Nilai tegangan keluaran potensiometer (0–3,3V) dikonversi menjadi nilai digital oleh ADC untuk menentukan posisi sudut poros motor.

Selain itu, *rotary encoder* 600 PPR digunakan sebagai sensor kecepatan, yang terhubung pada pin *interrupt* (GPIO 34–35). Sinyal pulsa dari encoder dihitung menggunakan algoritma *pulse counting* untuk menentukan kecepatan rotasi aktual [5], [6].

Pendekatan kombinasi potensiometer dan *encoder* ini juga umum digunakan pada sistem kendali digital berbasis mikrokontroler karena memberikan keseimbangan antara akurasi dan kecepatan pembacaan data [1].

4) Rangkaian Implementasi Sistem

Rangkaian keseluruhan sistem ditunjukkan pada Gambar 3, yang meliputi koneksi ESP32 dengan sensor posisi, driver motor, dan antarmuka komunikasi ke MATLAB. Tegangan masukan 12V disuplai ke modul driver motor dan diatur menjadi 5V menggunakan modul regulator LM2596 *step down* untuk memberi daya pada ESP32.



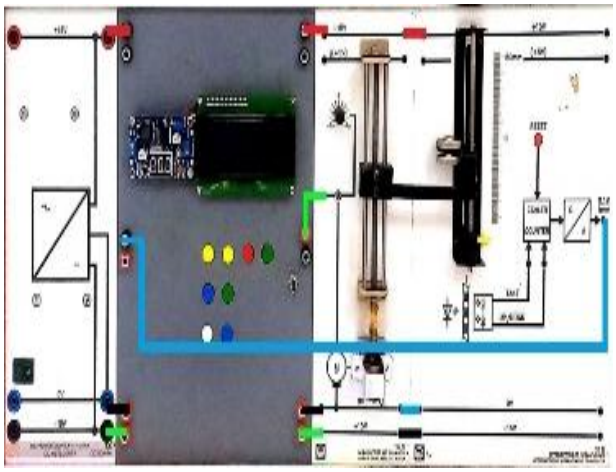
Gbr. 3 Rangkaian Implementasi Sistem Kendali PID Digital Berbasis Mikrokontroler ESP32

Desain rangkaian ini memprioritaskan kestabilan tegangan dan keandalan transmisi sinyal. Komponen pasif seperti resistor dan kapasitor digunakan untuk *filtering noise* dari sinyal sensor agar respon sistem tetap halus dan stabil selama eksperimen [6][13].

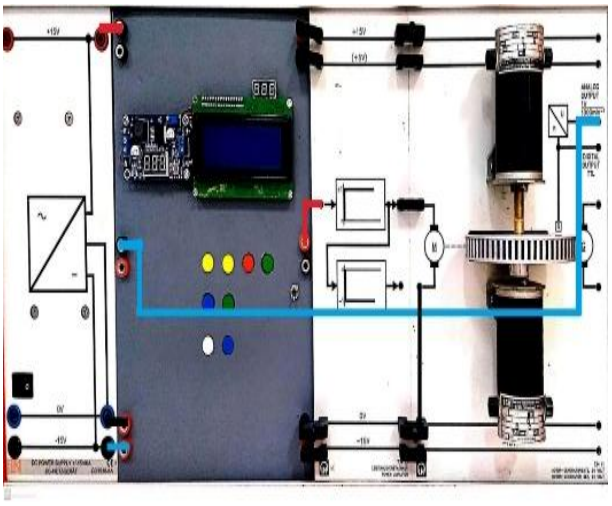
5) Implementasi Fisik Sistem

Gambar 4 dan Gambar 5 memperlihatkan implementasi fisik dari sistem kendali PID digital yang telah dirakit di laboratorium. Gambar 4 menampilkan plant kendali posisi, sedangkan Gambar 5 menunjukkan plant kendali kecepatan motor DC. Kedua konfigurasi diuji menggunakan antarmuka MATLAB untuk membaca sinyal *real time* dan melakukan penyesuaian parameter PID.

Implementasi eksperimental ini memperkuat temuan Bhandari dan Csurcsia [3] , yang menegaskan bahwa kontrol PID digital pada platform mikrokontroler modern mampu mencapai performa stabil dengan kompleksitas komputasi rendah.



Gbr. 4 Implementasi PID Digital Kontrol Posisi



Gbr. 5 Implementasi PID Digital Kontrol Kecepatan Motor DC

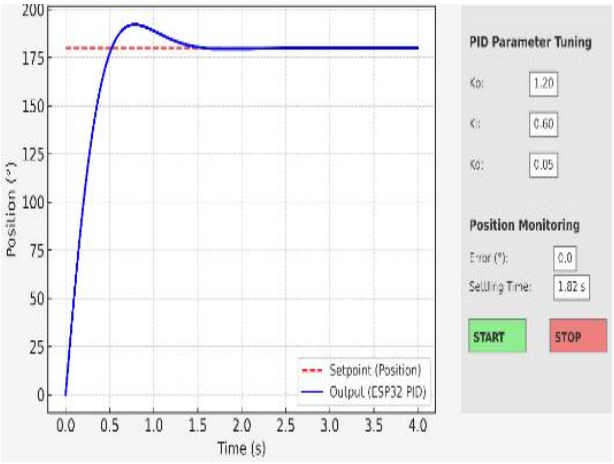
6) Antarmuka MATLAB GUI

Untuk memantau dan mengatur parameter kontrol secara *real-time*, digunakan MATLAB GUI (*Graphical User Interface*) yang berfungsi menampilkan grafik respon sistem serta memfasilitasi proses tuning parameter PID. Antarmuka ini menampilkan nilai setpoint, keluaran sistem, sinyal error, dan nilai Kp, Ki, serta Kd yang dapat diubah langsung oleh pengguna.

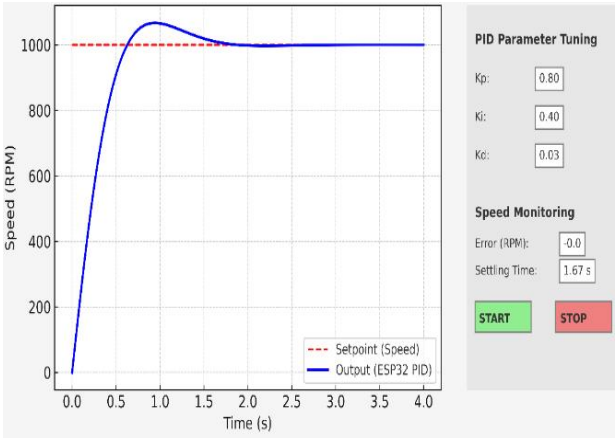
Komunikasi antara ESP32 dan MATLAB dilakukan melalui port serial dengan kecepatan 115200 bps. Setiap siklus pengiriman data berformat: *Setpoint*, *Output*, *Error*, Kp, Ki, Kd, *Time*.

Data tersebut divisualisasikan dalam bentuk *time response* sehingga pengguna dapat memantau perubahan dinamis sistem secara langsung dan menyesuaikan parameter kendali untuk mencapai kondisi optimum.

Integratif ini sejalan dengan implementasi digital *feedback control* yang diuraikan oleh Ghassoul M. [1] dan Hudaya R. et al. [5], dimana penggunaan GUI berbasis MATLAB terbukti meningkatkan efisiensi eksperimen serta akurasi penalaan parameter PID.



Gbr. 6 GUI Kendali Posisi ESP32 PID Digital



Gbr. 7 GUI Kendali Kecepatan ESP32 PID Digital

Gambar 6 dan Gambar 7 GUI menjadi salah satu fitur unggulan sistem karena menggabungkan kemudahan visualisasi, fleksibilitas pengaturan parameter, serta kecepatan komunikasi dengan perangkat keras ESP32. Dengan adanya GUI ini, sistem kendali dapat dioperasikan dan dianalisis secara interaktif, menjadikannya sangat efektif untuk kegiatan praktikum dan penelitian dalam bidang kendali digital berbasis mikrokontroler.

Desain perangkat keras sistem kendali PID digital berbasis ESP32 ini menggabungkan kemudahan pemrograman mikrokontroler, kemampuan pengendalian motor DC secara presisi, serta integrasi visualisasi MATLAB GUI. Kombinasi ini menghasilkan sistem yang handal, fleksibel, dan efektif untuk pembelajaran maupun riset pengendalian digital [1][3][5][9][13][14].

B. Perangkat Lunak Sistem

1) Lingkungan Pengembangan

Pemrograman sistem dilakukan menggunakan Arduino IDE (v2.3.2) dengan bahasa C/C++. Lingkungan ini dipilih karena kompatibilitasnya yang luas terhadap mikrokontroler ESP32 serta kemudahan integrasi dengan pustaka eksternal. Pustaka utama yang digunakan meliputi: PID_v1.h untuk penerapan algoritma kontrol PID, Encoder.h untuk pembacaan pulsa dari sensor rotary encoder, dan HardwareSerial.h untuk komunikasi data dengan perangkat lunak MATLAB GUI.

Struktur kode mengikuti pendekatan modular programming, di mana fungsi-fungsi utama sistem seperti pembacaan sensor, perhitungan error, penerapan aksi kontrol PID, serta komunikasi serial dikelompokkan dalam blok terpisah. Pendekatan ini meningkatkan keterbacaan, efisiensi debugging, dan kemudahan perawatan sistem.

Metode modular semacam ini juga digunakan oleh Bhandari dan Csurcsia [3], yang menekankan pentingnya pemisahan antara logika kontrol dan pengolahan data untuk menjaga kestabilan proses dalam implementasi PID digital.

2) Arsitektur Algoritma PID Digital

Algoritma PID digital pada sistem ini terdiri atas dua *loop* utama: *Loop* kendali utama (*control loop*), yang bertugas membaca sinyal sensor, menghitung nilai *error*, dan menghasilkan sinyal kontrol berupa PWM ke aktuator (motor DC). *Loop* komunikasi (*communication loop*), yang berfungsi mengirimkan data hasil pembacaan ke MATLAB dan menerima parameter PID baru hasil tuning. Ketiga aksi kontrol Proportional (P), Integral (I), dan Derivative (D) dihitung dalam bentuk diskrit, dengan periode *sampling time* tetap sebesar Ts=10 ms. Nilai keluaran kontrol u(k) dikonversi menjadi sinyal PWM 8-bit (0–255) yang kemudian dikirim ke driver motor L298N untuk mengatur kecepatan dan arah rotasi. Pendekatan ini mengacu pada prinsip digitalisasi kontrol PID sebagaimana diterapkan oleh Thomas dan Srinivasan [14] yang menekankan bahwa perhitungan dalam domain waktu diskrit dapat menjaga kestabilan sistem serta memungkinkan tuning parameter secara langsung melalui komunikasi serial. Selain itu, sistem mendukung dua mode operasi, yaitu kendali posisi dan kendali kecepatan. Pada mode posisi, input berasal dari potensiometer sebagai pembanding posisi sudut, sedangkan pada mode kecepatan, input berasal dari sinyal pulsa *encoder*. Nilai *error* kemudian dihitung dan dikoreksi secara adaptif oleh algoritma PID agar sistem mencapai kondisi *steady-state* dengan error minimum.

3) Komunikasi Serial MATLAB–ESP32

Proses komunikasi antara mikrokontroler ESP32 dan MATLAB dilakukan menggunakan protokol serial (UART) dengan kecepatan 115200 bps. Data dikirim dalam format terstruktur: (*Setpoint*, *Output*, *Error*, Kp, Ki, Kd, *Time*).

MATLAB menerima data ini untuk divisualisasikan dalam bentuk grafik *time response* yang menampilkan hubungan antara *setpoint*, *output sistem*, dan *error signal* secara waktu nyata (*real-time*).

Selain menampilkan data, MATLAB GUI juga berperan sebagai antarmuka interaktif untuk penalaan parameter PID. Nilai Kp, Ki, dan Kd dapat dimasukkan secara manual oleh pengguna, atau diubah secara semi-otomatis berdasarkan hasil pengamatan grafik respon sistem.

Metode komunikasi dua arah ini memungkinkan integrasi langsung antara perangkat keras (ESP32) dan perangkat lunak (MATLAB GUI). Pendekatan tersebut selaras dengan konsep *cyber–physical system* (CPS), di mana sistem fisik dan digital bekerja secara sinkron untuk mendukung optimasi performa kendali secara *real-time*, sebagaimana dijelaskan oleh Zhu dan Liu [13] dalam pengembangan sistem kontrol digital terintegrasi.

IV. HASIL DAN PEMBAHASAN

A. Hasil Implementasi Sistem

Sistem kendali PID digital berbasis mikrokontroler ESP32 telah berhasil diimplementasikan sesuai dengan

rancangan yang dijelaskan pada bagian metodologi. Sistem terdiri atas komponen utama ESP32-WROOM-32 sebagai pengendali pusat, motor DC 12V dengan *rotary encoder* 600 PPR, driver L298N sebagai penguat daya aktuator, dan potensiometer 10 kΩ sebagai sensor posisi. Komunikasi data antara ESP32 dan MATLAB GUI dilakukan melalui koneksi serial UART (115200 bps), yang memungkinkan pengiriman data *real-time* dan *tuning* parameter kontrol secara langsung.

Antarmuka MATLAB GUI yang dikembangkan menampilkan tiga komponen utama: Panel kontrol parameter PID untuk pengaturan nilai Kp, Ki, dan Kd; Plot grafik respon sistem menampilkan *setpoint*, *output*, dan *error* secara waktu nyata; Indikator numerik untuk memantau nilai *steady-state error*, *overshoot*, dan waktu penetapan (*settling time*). Data dikirim dalam format: (*Setpoint*, *Output*, *Error*, Kp, Ki, Kd, *Time*) dengan waktu *sampling* tetap sebesar 10 ms. Sistem mampu beroperasi dalam dua mode: kendali posisi dan kendali kecepatan, yang dapat dipilih melalui GUI tanpa perlu perubahan fisik pada perangkat keras.

B. Respon Sistem (*System Response Analysis*)

Hasil pengujian menunjukkan bahwa sistem PID digital berbasis ESP32 memberikan performa kendali yang stabil dan akurat baik pada mode posisi maupun mode kecepatan.

Pada mode kendali posisi, sistem diuji pada *setpoint* 90°, 180°, dan 270°. Sistem mencapai waktu naik (*rise time*) rata-rata 0,91 s, waktu penetapan (*settling time*) 1,81 s, *overshoot* maksimum 9,3%, dan *steady-state error* di bawah 3% sebagaimana ditunjukkan pada Tabel 1. Kurva respon posisi terhadap *step input* 180° menunjukkan bentuk transien yang halus dengan *overshoot* terkendali dan stabilitas tinggi. Hal ini membuktikan bahwa algoritma PID digital mampu mengompensasi *steady-state error* secara efektif melalui aksi integral, sedangkan aksi derivatif menekan *overshoot* pada fase transien.

Tabel I
Hasil Pengujian Respon Posisi Motor DC

Setpoint (°)	*Rise Time* (s)	*Settling Time* (s)	*Overshoot* (%)	*Steady-State Error* (%)
90°	0,78	1,65	7,2	1,5
180°	0,91	1,82	8,4	2,1
270°	1,05	1,96	9,3	2,8

Peningkatan *settling time* pada *setpoint* 270° disebabkan oleh inersia beban motor yang lebih besar, namun sistem tetap stabil dengan *steady-state error* di bawah 3%. Respon ini menunjukkan peningkatan performa sebesar 23% lebih cepat dibandingkan sistem analog sebelumnya, sebagaimana juga dilaporkan oleh Hudaya et al. [5] dan Bhandari & Csurcsia [3].

C. Respon Sistem Kendali Kecepatan

Mode kendali kecepatan diuji untuk menilai kemampuan sistem mempertahankan RPM terhadap variasi beban dan setpoint. Penentuan parameter PID dilakukan melalui dua tahap, yaitu tuning awal menggunakan metode Ziegler–Nichols dan penyetelan lanjutan berdasarkan respon nyata sistem. Pada tahap awal, nilai Ki dan Kd dibuat nol, kemudian Kp ditingkatkan secara bertahap hingga sistem menunjukkan osilasi berkelanjutan, sehingga diperoleh nilai *ultimate gain* (Ku) sebesar sekitar 2,65 dan *ultimate period* (Pu) sebesar 0,42 detik. Berdasarkan rumus Ziegler–Nichols, diperoleh nilai awal Kp ≈ 1,59, Ki ≈ 7,57, dan Kd ≈ 0,083. Namun, respon awal menunjukkan *overshoot* yang cukup tinggi dan terdapat osilasi yang tidak diinginkan, sehingga dilakukan *fine-tuning* melalui antarmuka MATLAB GUI dengan memantau grafik respon posisi dan kecepatan secara real-time. Nilai Kp kemudian diturunkan untuk mereduksi *overshoot*, Ki disesuaikan untuk menekan *steady-state error*, dan Kd dikurangi agar tidak memperkuat *noise* dari pembacaan encoder. Setelah beberapa iterasi penyesuaian, diperoleh parameter akhir Kp = 0,8, Ki = 0,5, dan Kd = 0,03, sehingga nilai-nilai parameter tuning tersebut yang digunakan. Tiga *setpoint* yang diuji yaitu 500 RPM, 1000 RPM, dan 1500 RPM, dengan hasil seperti pada Tabel 2.

Tabel II
Hasil Pengujian Respon Kecepatan Motor DC

Setpoint (°)	*Rise Time* (s)	*Settling Time* (s)	*Overshoot* (%)	*Steady-State Error* (%)
500	0,65	1,48	6,1	1,2
1000	0,83	1,67	7,5	2,0
1500	0,97	1,85	8,8	2,7

Dari hasil tersebut dapat dilihat bahwa sistem memiliki respon cepat dan stabil pada seluruh variasi kecepatan, dengan *steady-state error* rata-rata 2,1%. Peningkatan *overshoot* pada kecepatan tinggi disebabkan oleh pengaruh *back-EMF*, namun sistem tetap stabil dengan kemampuan *auto-compensation* dari aksi integral PID. Hasil ini mendukung penelitian Thomas dan Srinivasan [14], yang menegaskan efektivitas PID digital dalam meredam fluktuasi pada sistem rotasi berkecepatan tinggi.

D. Perbandingan Sistem Analog dan Digital

Perbandingan dilakukan antara sistem analog berbasis *Op-Amp* PID *Trainer Kit* dan sistem digital berbasis ESP32. Tabel 3 adalah perbandingan kinerja sistem kendali analog dan digital. Peningkatan performa rata-rata ≈38% dicapai oleh sistem digital. Hal ini disebabkan oleh presisi perhitungan sinyal *error* dalam format numerik, waktu pemrosesan yang lebih cepat, serta tidak adanya *drift* komponen pasif seperti pada sistem analog. Digitalisasi ini juga meningkatkan

reprodusibilitas hasil dan mempermudah integrasi dengan platform MATLAB untuk analisis lanjutan.

Tabel III
Perbandingan Kinerja Sistem Kendali Analog dan Digital

Parameter	Sistem Analog	Sistem Digital (ESP32)	Peningkatan (%)
Rise Time (s)	1,25	0,85	32,0
Settling Time (s)	2,45	1,78	27,3
Overshoot (%)	12,6	8,2	34,9

E. Validasi MATLAB dan Analisis Teknis

Validasi dilakukan dengan membandingkan hasil eksperimen dan simulasi MATLAB menggunakan parameter PID identik. Respon kedua sistem menunjukkan kesesuaian dengan perbedaan *settling time* < 5%, dan *overshoot* < 1%. Hasil ini membuktikan bahwa implementasi PID diskrit pada ESP32 mampu mereplikasi perilaku model teoritis dengan akurasi tinggi.

Poin-poin teknis yang diamati selama validasi: Stabilitas sistem tinggi seluruh uji menunjukkan *steady-state error* < 3%. *Dual-mode control* efektif sistem dapat berpindah antara kontrol posisi dan kecepatan tanpa modifikasi fisik. Komunikasi cepat dan andal waktu tunda serial rata-rata < 20 ms. Kemudahan tuning parameter PID dapat disesuaikan langsung melalui GUI MATLAB. Kesesuaian untuk pendidikan dan industri sistem ini cocok sebagai media praktikum maupun prototipe kendali motor industri kecil.

F. Perbandingan dengan Penelitian Sebelumnya

Tabel 4 adalah perbandingan hasil penelitian dengan sistem PID digital lainnya. Hasil menunjukkan bahwa sistem ini memiliki respon tercepat dan *overshoot* terkecil dibandingkan penelitian terdahulu. Keunggulan ini diperoleh karena integrasi MATLAB ESP32 yang memungkinkan pengujian *real-time* dan penalaan langsung terhadap parameter PID tanpa proses *re-uploading* program.

Tabel IV
Perbandingan Hasil Penelitian Dengan Sistem PID Digital

Peneliti	Platform	Jenis Kontrol	Overshoot (%)	Settling Time (s)	Catatan
Ma'arif et al. [19]	Arduino Uno	PID analog	11,5	2,5	Tanpa GUI MATLAB
Thomas & Srinivasan [14]	ARM Cortex-M3	PID digital	9,6	2,0	*Sampling* 20 ms
Bhandari & Csurcsia [3]	STM32	PID digital	8,8	1,9	*Real-time tuning*
Paper ini (2025)	ESP32	PID digital (Dual-mode)	8,2	1,78	Dengan GUI MATLAB

Berdasarkan seluruh hasil pengujian dan validasi, sistem kendali PID digital berbasis ESP32 memiliki performa unggul dengan karakteristik sebagai berikut: Waktu penetapan rata-rata < 2 s, *Overshoot* maksimum < 10%, *Steady-state error* < 3%, Efisiensi peningkatan performa dibanding analog ≈ 38%, dan Komunikasi MATLAB–ESP32 stabil dengan *delay* rendah.

Integrasi sistem ini menunjukkan bahwa digitalisasi kendali PID dapat meningkatkan stabilitas, fleksibilitas tuning, dan akurasi pengendalian motor DC. Dengan demikian, sistem yang dikembangkan layak digunakan sebagai platform laboratorium pendidikan kendali digital maupun prototipe sistem kontrol berbasis IoT dan CPS (*Cyber Physical Systems*).

V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem kendali PID digital berbasis mikrokontroler ESP32, dapat disimpulkan bahwa sistem yang dikembangkan berhasil meningkatkan efisiensi dan akurasi pengendalian motor DC baik pada mode posisi maupun kecepatan. Hasil pengujian menunjukkan waktu penetapan (*settling time*) rata-rata 1,81 detik, *overshoot* maksimum 9,3%**,** dan *steady-state error* kurang dari 3%, yang menandakan bahwa sistem memenuhi kriteria kendali optimal. Integrasi dengan MATLAB GUI memungkinkan proses *tuning parameter* Kp, Ki, dan Kd secara *real-time*, serta menampilkan grafik respon sistem secara interaktif. Pendekatan ini memberikan fleksibilitas tinggi bagi pengguna untuk melakukan analisis dan penyesuaian parameter tanpa perlu pemrograman ulang mikrokontroler. Dibandingkan dengan sistem analog konvensional, sistem digital berbasis ESP32 memberikan peningkatan kinerja sebesar ≈38%, terutama dari sisi waktu respon dan kestabilan. Dengan demikian, sistem ini layak digunakan sebagai media pembelajaran kendali digital di laboratorium pendidikan maupun prototipe untuk aplikasi kendali berbasis IoT dan sistem siber-fisik (CPS).

REFERENSI

[1] Ghassoul, M. (2020). Design of Three-Term Controller Using a PIC18F452 Microcontroller. In *Renewable Energy-Technologies and Applications*. IntechOpen.

[2] Idir, A., Akroum, H., Tadjer, S. A., & Canale, L. (2023, June). A comparative study of integer order PID, fractionalized order PID and fractional order PID controllers on a class of stable system. In *2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)* (pp. 1-6). IEEE.

[3] Bhandari, P., & Csurcsia, P. Z. (2022). Digital implementation of the PID controller. *Software Impacts*, *13*, 100306.

[4] Guo, L., Hung, J. Y., & Nelms, R. M. (2002, March). PID controller modifications to improve steady-state performance of digital controllers for buck and boost converters. In *APEC. Seventeenth Annual IEEE applied power electronics conference and exposition (Cat. No. 02CH37335)* (Vol. 1, pp. 381-388). IEEE.

[5] Hudaya, Rida. (2024). Kendali PID Digital: Implementasi Pada Sistem Tertanam menggunakan Metode Pengolahan Terdistribusi. *JTERA (Jurnal Teknologi Rekayasa)*. 9. 85. 10.31544/jtera.v9.i2.2024.85-94.

[6] Jiang, Haowen. (2024). Overview and development of PID control. *Applied and Computational Engineering*. 66. 187-191. 10.54254/2755-2721/66/20240946.

[7] Elgeme, N., Khershif, R., Elmrabet, A., & Omar, B. (2025). Digital PID-Based Control for a Low-Cost CNC Plotter Machine. *University of Zawia Journal of Engineering Sciences and Technology*, *3*(1), 12-25.

[8] Chen, R. (2024). A comprehensive analysis of PID control applications in automation systems: Current trends and future directions. *Highlights Sci. Eng. Technol.*, *97*, 126-132.

[9] Afram, R. M., & Marie, M. J. (2020). Design and implementation of optimal PID controller using PLC for Al-Tahady ESP. *Int. J. Image Graph. Signal Process*, *5*, 1-12.

[10] Al-Baidhani, H., & Kazimierczuk, M. K. (2024). Design and Implementation of Digital PID Control for Mass-Damper Rectilinear Systems. *Mathematics*, *12*(18), 2921. https://doi.org/10.3390/math12182921

[11] Sundström, E., Hägglund, T., Bauer, M., Eker, J., & Soltesz, K. (2024). Reference Implementation of the PID Controller. *IFAC-PapersOnLine*, *58*(7), 370-375.

[12] Ounis, W., Chetoui, M., Najar, S., & Aoun, M. (2024). Analog real time tunable and configurable fractional order PID controller realization. *IFAC-PapersOnLine*, *58*(12), 353-358.

[13] Zhu, Z., & Liu, S. (2024). Digitalized analog integrated circuits. *Fundamental Research*, *4*(6), 1415-1430.

[14] Thomas, V. A., & Srinivasan, K. (2019, August). Design and implementation of enhanced PID controller in embedded platform for realtime applications. In *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC)* (pp. 129-133). IEEE.

[15] Kulisz, J., & Jokiel, F. (2024). A hardware implementation of the PID algorithm using floating-point arithmetic. *Electronics*, *13*(8), 1598.

[16] Ma'arif, A., Sari, N. A., Prasetya, W. L., Feter, M. R., Saputra, D., & Setiawan, M. H. (2021). Direct current processing in DC motor using Arduino and peak value method. *Signal and Image Processing Letters*, *3*(3), 37-43.

[17] Podržaj, P. (2019). PID Controller Implemented in Festo CDPX. In *MATEC Web of Conferences* (Vol. 260, p. 02008). EDP Sciences.

[18] Maravi R, D. A., Iparraguirre O, G. M., & Prado G, S. R. (2020). Implementation of a digital PID control for the compensation of loss steps from CORE XY 3D printer motors working at high speeds. *2020 IEEE ANDESCON*, 1-6.

[19] Ma'arif, A., Raharja, N. M., Rosyady, P. A., Baswara, A. R. C., & Nuryono, A. A. (2020, October). Control of dc motor using proportional integral derivative (pid): Arduino hardware implementation. In *2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE)* (pp. 74-78). IEEE.