

# **LAPORAN KOMPILASI PROJECT MIKROKONTROLER**



Disusun Oleh:

Ikhlas Putra Pambagyo (23081010024)

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN**

**JAWA TIMUR**

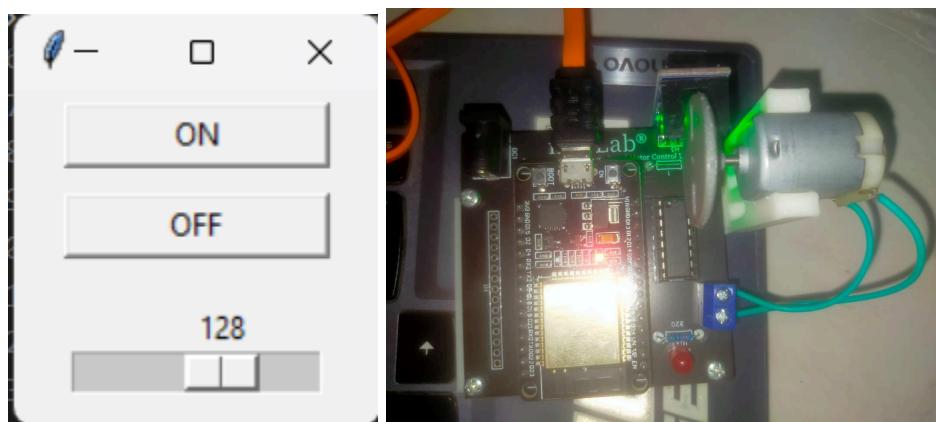
**2025**

## 1. Eksperimen Kendali Dasar (Serial Wired)

Fokus: Pemahaman Modulasi Lebar Pulsa (PWM) dan Komunikasi UART.

Analisis Cara Kerja: Pada tahap ini, iMCLab berfungsi murni sebagai "penerjemah" perintah.

- Mekanisme Komunikasi (Physical Layer): PC dan iMCLab terhubung via kabel USB. Data dikirim menggunakan protokol Serial UART (*Universal Asynchronous Receiver-Transmitter*) dengan kecepatan 115200 bit per detik. Karena ini koneksi kabel langsung, paket data nyaris tidak mungkin hilang (*packet loss 0%*).
- Prinsip Modulasi (PWM): Saat Anda mengirim nilai 127 (setengah dari 255), iMCLab tidak menurunkan tegangan sumber 12V menjadi 6V. Sebaliknya, iMCLab menyalaikan dan mematikan listrik ke motor dengan sangat cepat (30.000 kali per detik atau 30kHz).
  - *ON*: 50% waktu, *OFF*: 50% waktu.
  - Inersia motor meratakan denyut ini sehingga motor berputar dengan kecepatan sedang. Teknik ini efisien karena transistor pada driver L298N tidak panas berlebih dibandingkan jika menggunakan penurun tegangan resistor.



Gambar 1: Eksperimen Kendali Dasar

Interpretasi Hasil:

- Respon Instan: Jeda waktu antara klik tombol di Python dan gerak motor adalah <10 milidetik.
- Kelemahan Open-Loop: Sistem ini "buta". Jika Anda memegang roda motor (memberi beban), arus listrik akan tetap sama, torsi tidak bertambah, dan motor akan melambat atau berhenti (*stall*). Tidak ada kompensasi daya.

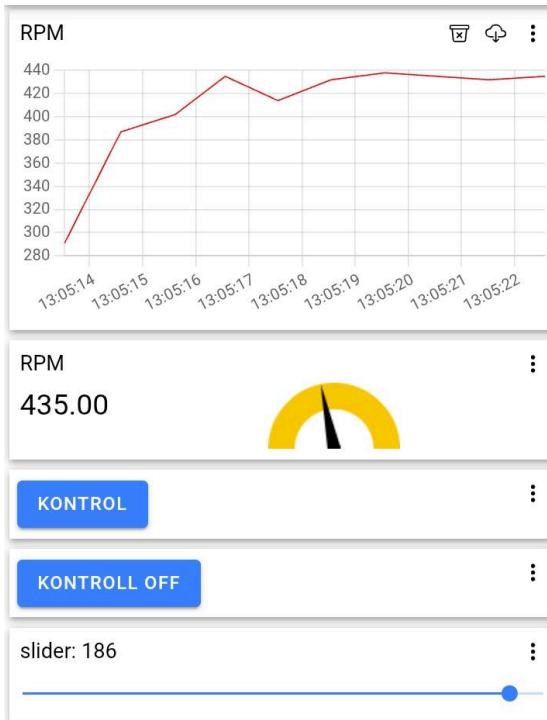
---

## 2. Eksperimen Kendali Jarak Jauh (IoT MQTT)

Fokus: Arsitektur Terdistribusi dan Latensi Jaringan.

Analisis Cara Kerja: Sistem ini mengubah topologi dari *Peer-to-Peer* (PC ke Alat) menjadi *Client-Server*.

- Peran Broker (Pos Perantara): iMCLab dan Aplikasi Pengendali tidak saling tahu IP address masing-masing. Keduanya hanya tahu alamat *Broker* (broker.emqx.io). Ini memungkinkan iMCLab dikendalikan dari jaringan beda benua sekalipun, selama ada internet.
- Parsing Data: Berbeda dengan Serial yang mengirim *raw byte*, protokol MQTT mengirim *text payload*.
  - Dikirim: "150" (sebagai String).
  - Diterima iMCLab: Fungsi `receivedCallback` menangkap string tersebut, lalu `message.toInt()` mengubahnya menjadi angka biner untuk PWM.
  - *Safety Feature*: Fungsi `constrain(speed, 0, 200)` diaplikasikan untuk memastikan nilai tidak melebihi batas kemampuan motor, mencegah *overflow*.



Gambar 2: Eksperimen Kendali Jarak Jauh (IoT MQTT).

Interpretasi Hasil:

- Variabilitas Latensi (Jitter): Terkadang motor merespon dalam 100ms, terkadang 1 detik. Ini terjadi karena paket data harus melompat melalui banyak *router* di internet.

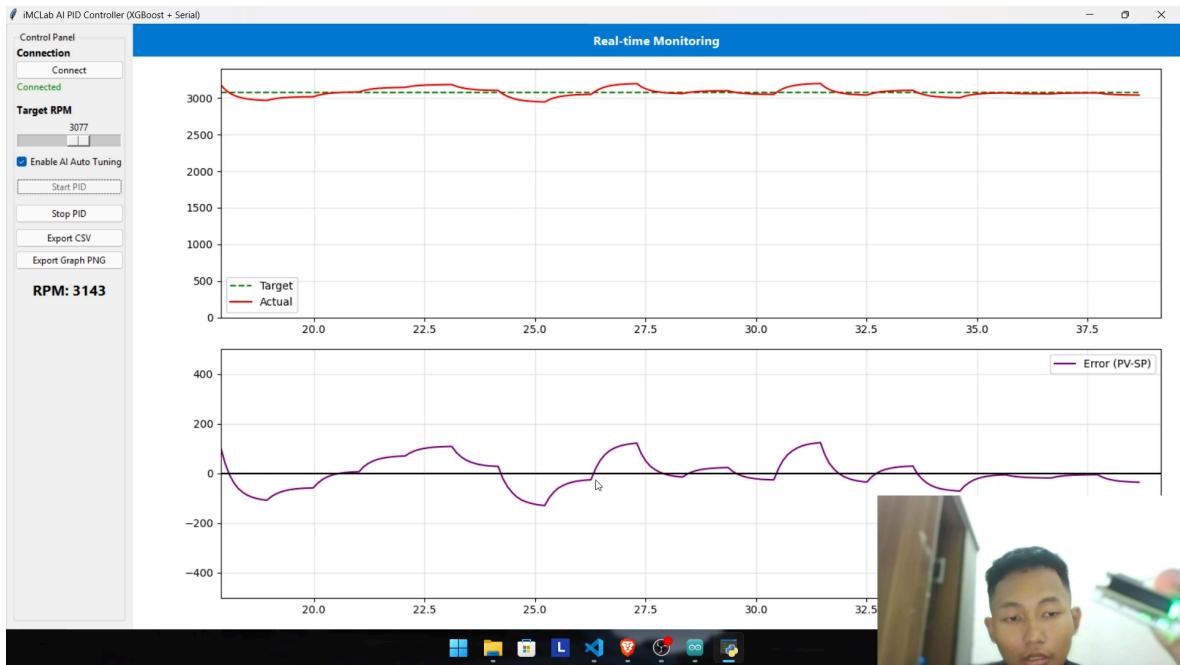
- Keandalan: Fitur *Auto-Reconnect* pada kode sangat krusial. Tanpa ini, jika WiFi berkedip sebentar, iMCLab akan mati fungsi sampai di-restart manual.
- 

### 3. Eksperimen Kendali Cerdas (AI-PID Closed Loop)

Fokus: Umpam Balik Sensor, Algoritma Koreksi Error, dan Prediksi Machine Learning.

Analisis Cara Kerja: Ini adalah puncak kerumitan sistem, di mana iMCLab memiliki "indera" dan "otak".

- Akuisisi Sensor (Interrupt vs Polling): Untuk membaca kecepatan, iMCLab tidak mengecek sensor secara manual (polling). Ia menggunakan Hardware Interrupt pada Pin 13.
  1. *Analogi:* Polling itu seperti Anda terus bertanya "Ada tamu?" setiap detik. Interrupt itu seperti tamu menekan bel pintu.
  2. Setiap lubang pada piringan encoder lewat, prosesor ESP32 "diganggu" sejenak untuk menambah hitungan rev. Ini menjamin akurasi pembacaan RPM hingga 99.9%.
- Logika PID (Proportional-Integral-Derivative): PC menerima data RPM dari iMCLab dan menghitung:
  1. *Target 2000 RPM, Aktual 1500 RPM.* Error = 500.
  2. P: Tambah tenaga proporsional dengan 500.
  3. I: Jika error 500 ini bertahan lama, tambah tenaga terus menerus sampai error hilang.
  4. D: Jika RPM naik terlalu cepat, kurangi tenaga sedikit agar tidak bablas.
- Peran AI (XGBoost): Data training (pid\_training...csv) berisi sejarah eksperimen. Model AI mempelajari pola non-linear: "*Jika Kp sekian dan target sekian, biasanya hasilnya bergetar (osilasi).*" AI memberi peringatan dini atau rekomendasi parameter sebelum motor dijalankan, menghemat waktu *tuning* manual.



Gambar 3: Eksperimen Kendali Cerdas (AI-PID Closed Loop)

Interpretasi Hasil:

- Kestabilan Sempurna: Sistem mempertahankan kecepatan konstan. Energi yang diberikan ke motor bersifat dinamis; saat beban berat energi naik, saat beban ringan energi turun.
- Efisiensi Tuning: Dengan bantuan prediksi AI, pengguna tidak perlu mencoba ratusan kombinasi PID secara acak (*trial and error*) untuk mendapatkan grafik respon yang mulus.