

Laporan praktek UAS
Untuk Memenuhi Tugas Mata Kuliah Mikrokontroller



Oleh :

Caezarlov Nugraha (23081010182)

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL (VETERAN) JAWA TIMUR
SURABAYA
2025**

1. Eksperimen Konektivitas IoT (MQTT & Hotspot Gateway)

Deskripsi: Eksperimen ini bertujuan untuk menghubungkan ESP32 ke internet menggunakan *hotspot* dari Samsung Galaxy S24 Ultra guna menghindari protokol keamanan WiFi kampus yang kompleks. Data dikirim dan diterima melalui protokol MQTT menggunakan broker *broker.emqx.io*.

Foto Dokumentasi: (*Merujuk pada foto kedua yang menampilkan konfigurasi SSID "Laphone's S24 Ultra"*)

Analisis:

- **Efisiensi:** Penggunaan MQTT memungkinkan latensi rendah dalam pengiriman perintah sp (setpoint) dari ponsel ke motor.
- **Security:** Penggunaan *client ID* acak (random(0xffff)) mencegah konflik koneksi pada broker publik.

2. Eksperimen Interfacing Sensor & Pengukuran RPM

Deskripsi: Menggunakan sensor optocoupler yang membaca piringan *encoder* (2 lubang) pada motor DC. Program menggunakan teknik **Interrupt Service Routine (ISR)** untuk menghitung setiap pulsa yang masuk tanpa mengganggu proses utama dalam loop().

Foto Dokumentasi: (*Merujuk pada foto pertama yang memperlihatkan piringan encoder pada motor*)

Analisis:

- **Akurasi:** Kode menggunakan filter sederhana ($0.7 * \text{rpm_filtered} + 0.3 * \text{rpm}$) untuk menghaluskan fluktuasi pembacaan (Noise Reduction).
- **Logika Kalkulasi:** Dengan 2 lubang pada encoder, rumus yang digunakan adalah:

$$RPM = \frac{(Rev/2) \times 60000}{TimeElapsed}$$

Hal ini memastikan pembacaan tetap presisi meskipun motor berputar pada kecepatan tinggi.

3. Eksperimen Kontrol Kecepatan Tertutup (PID Control)

Deskripsi: Eksperimen ini merupakan inti dari sistem, di mana ESP32 bertugas menjaga kecepatan motor agar tetap stabil sesuai target (*setpoint*) meskipun ada beban atau fluktuasi tegangan.

Analisis Algoritma: Sistem menggunakan persamaan PID (Proportional-Integral-Derivative) untuk menghitung *output* PWM:

$$Output = K_p \cdot e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

- **Tuning:** Nilai $K_c = 0.007$ menunjukkan penguatan yang cukup kecil untuk menghindari *overshoot* yang besar pada motor DC kecil tersebut.
- **Windup Prevention:** Terdapat logika pengaman $ierr = ierr - KI * error * dt$ yang berfungsi mencegah akumulasi error berlebih (*integral windup*) saat motor mencapai batas maksimal PWM (255).

Kesimpulan

Eksperimen ini berhasil mengintegrasikan tiga pilar utama sistem tertanam: **Sensing** (RPM), **Actuating** (PWM Motor), dan **Communication** (MQTT). Sistem mampu merespons perubahan target RPM dari jarak jauh dengan stabilitas yang dijaga oleh algoritma PID.

Status Proyek: Berhasil dijalankan (Resetting via RTS pin... OK).

Foto Dokumentasi:

