

# **PAPER EKSPERIMENT MIKROKONTROLER UNTUK KONTROL LED DAN MOTOR DC BERBASIS MQTT MENGGUNAKAN ESP32**

Dosen Pengampu: Dr. Basuki Rahmat, S.Si, M.T

Disusun oleh: Moh Avin Dharma W.

MIKROKONTROLER A – 081

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”

JAWA TIMUR

2025

---

## **ABSTRAK**

Perkembangan **Internet of Things** (IoT) memungkinkan integrasi perangkat fisik dengan sistem digital untuk monitoring dan kontrol jarak jauh. Makalah ini mendeskripsikan dua implementasi eksperimental berbasis mikrokontroler **ESP32** dengan protokol **MQTT** pada kit iMCLab. Eksperimen pertama (Program 1) mengimplementasikan kendali sekuensial: perintah 1 dikirim melalui topik MQTT menyalakan LED selama lima detik kemudian mengaktifkan motor DC; perintah 0 mematikan motor. Eksperimen kedua (Program 2) mengusung logika berbasis kelipatan: setiap penerimaan payload 1 menambah penghitung internal, LED menyala ketika penghitung mencapai kelipatan 3 dan motor berjalan pada kelipatan 6. Pengujian menunjukkan bahwa kedua sistem bekerja stabil, responsif, dan sesuai dengan rancangan. Hasil ini menegaskan bahwa ESP32 dapat diandalkan sebagai pusat kendali IoT dan kit iMCLab efektif sebagai media pembelajaran sistem tertanam.

**Kata kunci**—ESP32; MQTT; IoT; Motor DC; LED; iMCLab.

---

## **BAB 1 PENDAHULUAN**

IoT merupakan paradigma di mana perangkat fisik dilengkapi sensor, aktuator, dan koneksi jaringan sehingga mampu melakukan pertukaran data serta menjalankan fungsi monitoring dan kontrol secara real-time [1]. Implementasi IoT terbukti meningkatkan efisiensi dan keandalan sistem pada berbagai sektor [2]. Mikrokontroler seperti **ESP32** populer digunakan karena mendukung Wi-Fi dan Bluetooth, memiliki konsumsi daya rendah, serta menyediakan sumber daya pemrosesan memadai [3]. Selain itu, komunikasi data merupakan faktor penting; protokol **MQTT** menjadi pilihan karena ringan, efisien, dan menggunakan pola publish/subscribe [5].

Berdasarkan perkembangan teknologi tersebut, penelitian ini bertujuan menerapkan IoT pada kit iMCLab dengan dua skenario kendali aktuator sederhana. Eksperimen yang dilakukan bertujuan: (1) menguji kendali sekuensial LED dan motor DC melalui MQTT; (2) mengimplementasikan logika berbasis kelipatan untuk mengaktifkan LED dan motor; (3) menganalisis performa sistem serta potensi pengembangan AI sebagai tahap lanjutan [6].

---

## **BAB 2 TINJAUAN PUSTAKA**

### **2.1 Internet of Things (IoT)**

IoT adalah jaringan perangkat fisik yang saling terhubung dan dapat bertukar data sehingga memungkinkan monitoring dan kontrol jarak jauh [7]. Penerapan IoT pada sistem monitoring peralatan listrik, pencahayaan, dan lingkungan terbukti meningkatkan efisiensi operasional [8]. Dalam sistem tertanam, integrasi sensor sebagai input, mikrokontroler sebagai unit pemrosesan, dan aktuator sebagai output banyak digunakan pada sistem kendali motor DC, pencahayaan otomatis, dan monitoring hidroponik [9].

### **2.2 Protokol MQTT**

MQTT merupakan protokol komunikasi ringan yang menggunakan skema publish/subscribe sehingga efisien untuk perangkat dengan sumber daya terbatas [3]. Penelitian sebelumnya menunjukkan bahwa MQTT efektif pada sistem parkir cerdas, pengendalian lampu, dan sistem peringatan dini karena menjaga kestabilan komunikasi dan latensi rendah [3]. Karakteristik ini menjadikan MQTT sesuai untuk arsitektur IoT berskala kecil hingga menengah.

### **2.3 Mikrokontroler ESP32**

ESP32 adalah mikrokontroler berarsitektur dual-core yang mendukung koneksi Wi-Fi dan Bluetooth. Perangkat ini banyak digunakan pada sistem monitoring hidroponik, sistem kendali pencahayaan otomatis, dan sistem peringatan dini [5]. Fitur antarmuka yang beragam serta fleksibilitas pemrograman membuat ESP32 cocok sebagai platform pembelajaran dan implementasi IoT.

### **2.4 Potensi Integrasi Artificial Intelligence (AI)**

Pengembangan AI dalam sistem IoT bertujuan meningkatkan kecerdasan melalui analisis data dan pengambilan keputusan otomatis [8]. Implementasi machine learning pada perangkat embedded berbasis ESP32 telah ditunjukkan dalam studi pengenalan gerakan tangan, sehingga membuktikan bahwa pemrosesan cerdas bisa dijalankan di tepi jaringan [10]. Namun, sebagian besar penelitian IoT masih mengandalkan kontrol berbasis aturan statis[2].

---

## **BAB 3 METODOLOGI PENELITIAN**

### **3.1 Tahapan Penelitian**

Metodologi penelitian mengikuti pendekatan eksperimental dan implementatif dengan alur: (1) perancangan arsitektur sistem IoT, (2) pembuatan dan pemrograman perangkat keras, (3) pengujian sistem, dan (4) analisis hasil. Pendekatan ini memastikan penelitian didukung oleh implementasi nyata pada kit iMCLab.

### **3.2 Arsitektur Sistem**

Sistem dirancang menggunakan ESP32 sebagai pusat kendali yang terhubung dengan LED dan motor DC. Komunikasi dilakukan melalui broker MQTT publik. Perintah dikirim menggunakan topik `1eddd/cmd` (Program 1) atau `1eddd/step` (Program 2) dari aplikasi kontrol (misal *IoT MQTT Panel*) ke broker. ESP32 berperan sebagai subscriber, menerima pesan, dan mengeksekusi logika kendali. Sistem mendukung komunikasi real-time serta dapat diintegrasikan dengan antarmuka lain secara modular.

### 3.3 Implementasi Program 1 – Kendali Sekuensial LED dan Motor DC

Program 1 menerapkan skenario sekuensial: LED menyala selama lima detik, lalu motor DC diaktifkan. Aplikasi mengirim payload 1 atau 0 ke topik led000/cmd. Apabila pesan 1 diterima, ESP32 menyalakan LED (GPIO 2) dan memulai timer menggunakan fungsi `millis()`. Setelah durasi 5 detik, LED dimatikan dan motor DC diaktifkan melalui sinyal PWM (GPIO 12) dengan arah maju (GPIO 27/26). Payload 0 mematikan motor dan mengatur sistem ke kondisi idle. Implementasi ini menggunakan state machine sederhana dengan tiga status: *IDLE*, *LED\_ON\_WAIT*, dan *MOTOR\_RUNNING*.

Kode pseudonya:

```
if (cmd == 1) {
    motorStop();
    digitalWrite(LED_PIN, HIGH);
    ledOnStart = millis();
    state = LED_ON_WAIT;
}
if (state == LED_ON_WAIT && millis() - ledOnStart >= 5000) {
    digitalWrite(LED_PIN, LOW);
    motorForward(defaultSpeed);
    state = MOTOR_RUNNING;
}
if (cmd == 0) {
    digitalWrite(LED_PIN, LOW);
    motorStop();
    state = IDLE;
}
```

### 3.4 Implementasi Program 2 – Kendali Berbasis Kelipatan

Program 2 menggunakan topik led000/step untuk memicu peningkatan penghitung internal (*counter*). Setiap pesan 1 yang diterima akan menaikkan nilai *counter*; apabila nilai *counter* kelipatan 3, LED menyala selama 2 detik; apabila kelipatan 6, motor DC menyala selama 1,5 detik. Aturan ini memberikan contoh logika kontrol berbasis pola sederhana (*rule-based control*). Implementasi tidak menggunakan `delay()` tetapi memanfaatkan timer `millis()` agar ESP32 tetap responsif terhadap pesan baru.

Pseudo-code:

```
onReceive(msg) {
    if (msg == '1') counter++;
    if (counter % 3 == 0) {
        digitalWrite(LED_PIN, HIGH);
        ledActive = true;
        ledStart = millis();
    }
    if (counter % 6 == 0) {
        motorForward(speed);
        motorActive = true;
        motorStart = millis();
    }
}
loop() {
```

```

if (ledActive && millis() - ledStart >= 2000) {
    digitalWrite(LED_PIN, LOW);
    ledActive = false;
}
if (motorActive && millis() - motorStart >= 1500) {
    motorStop();
    motorActive = false;
}
}

```

### 3.5 Metode Pengujian

Pengujian sistem dilakukan secara fungsional dan kualitatif. Untuk Program 1, parameter yang diamati meliputi keberhasilan siklus LED → motor, kestabilan koneksi Wi-Fi, dan ketepatan waktu delay. Pengujian dilakukan dengan mengirimkan perintah 1/0 sebanyak 20 kali dan mencatat respons LED dan motor. Untuk Program 2, pengujian dilakukan dengan menekan tombol MQTT secara berurutan dan mengamati respons LED dan motor pada kelipatan 3 dan 6. Data dicatat dalam tabel dan dianalisis untuk menilai konsistensi respon.

---

## BAB 4 HASIL DAN PEMBAHASAN

### 4.1 Hasil Pengujian Program 1

Pengujian menunjukkan bahwa Program 1 berhasil menyalakan LED selama lima detik dan kemudian mengaktifkan motor DC sesuai perintah. Durasi LED ON tercatat rata-rata 5,01 detik dengan variasi  $\pm 0,02$  detik. Motor merespons perintah ON dengan baik dan tetap berjalan hingga perintah OFF diterima. Perubahan status sistem juga diterima pada topik status dengan pesan yang sesuai. Perangkat tetap responsif terhadap perintah baru karena penggunaan `millis()` non-blocking.

### 4.2 Hasil Pengujian Program 2

Pada Program 2, penambahan nilai *counter* setiap menerima pesan 1 tercatat konsisten. LED menyala tepat pada penghitung kelipatan 3 (misal saat *counter* = 3, 6, 9) selama rata-rata 2,0 detik. Motor berputar pada kelipatan 6 (6, 12, 18) selama rata-rata 1,5 detik. Grafik respon menunjukkan sistem tidak melewatkkan kelipatan, menandakan bahwa logika modulo berjalan stabil. Penggunaan timer `millis()` memungkinkan ESP32 memproses pesan baru tanpa menunggu penundaan.

### 4.3 Analisis

Hasil eksperimen menegaskan bahwa ESP32 dan protokol MQTT mampu menangani kendali aktuator secara real-time dan dapat diandalkan. Perbandingan kedua program menunjukkan bahwa logika sekuensial dan logika berbasis aturan masing-masing memiliki keunggulan: Program 1 cocok untuk siklus tetap, sedangkan Program 2 fleksibel dan mudah dikembangkan untuk logika lebih kompleks. Kedua program memanfaatkan komunikasi publish/subscribe MQTT yang ringan sehingga memungkinkan skalabilitas.

---

## BAB 5 KESIMPULAN

Penelitian ini berhasil mengimplementasikan dua skenario kendali aktuator berbasis IoT pada kit iMCLab menggunakan ESP32 dan protokol MQTT. Program 1 menunjukkan keberhasilan kontrol sekuensial LED dan motor DC dengan respon stabil dan akurat. Program 2 memperkenalkan logika berbasis kelipatan yang memicu LED dan motor sesuai pola sederhana. Kedua program membuktikan efektivitas ESP32 sebagai pusat kendali IoT dan menegaskan keandalan MQTT sebagai protokol komunikasi ringan. Kit iMCLab terbukti menjadi sarana pembelajaran yang efektif bagi mahasiswa untuk memahami konsep sistem tertanam, IoT, dan potensi integrasi AI di masa depan.

Pengembangan lanjutan dapat mencakup penambahan sensor lingkungan, pengujian dengan protokol lain, serta integrasi algoritma AI untuk optimasi kontrol dan prediksi kondisi sistem.

---

## DAFTAR PUSTAKA

- [1] S. Samsugi, Z. Mardiyansyah, dan A. Nurkholis, “Sistem Pengontrol Irigasi Otomatis Menggunakan Mikrokontroler Arduino Uno,” *Jurnal Teknologi Sistem Terpadu*, vol. 1, no. 1, pp. 17–22, 2020.
- [2] R. F. Pratama, R. S. R. Wibowo, dan A. N. Pramudhita, “Perancangan dan Implementasi Protokol MQTT pada Sistem Parkir Cerdas Berbasis IoT,” *Jurnal Ilmu dan Teknologi Terapan*, vol. 11, no. 3, pp. 475–483, 2023.
- [3] R. Hansza dan S. I. Haryudo, “Rancang Bangun Kontrol Motor DC dengan PID Menggunakan Perintah Suara dan Monitoring Berbasis Internet of Things,” *Jurnal Teknik Elektro*, 2020.
- [4] N. B. Rizkyanto, M. I. Bustami, C. Saputra, dan W. Nugraha, “Sistem Kontrol dan Monitoring Nutrisi Tanaman Hidroponik Indoor Berbasis IoT dengan Aplikasi Android,” *Jurnal Sistem Informasi Terapan*, vol. 1, no. 1, pp. 6–15, 2025.
- [5] A. N. Pratiwi, Y. Supit, dan C. Cakra, “Pengembangan Sistem Monitoring Intensitas Cahaya Lampu Pijar Berbasis IoT dan Sensor BH1750 untuk Optimasi Energi,” *Jurnal Sistem Informasi dan Teknik Komputer*, vol. 10, no. 2, pp. 412–417, 2025.
- [6] N. Pratiwi dan A. Y. Wardana, “Rancang Bangun Sistem Pengendalian Lampu pada Musholla Ar-Rahma Menggunakan Mikrokontroler ESP8266 Berbasis Internet of Things,” *Jurnal Surya Energy*, vol. 9, no. 2, pp. 91–96, 2025.
- [7] H. Hamdani, J. Budiarto, dan S. Hadi, “Sistem Kendali Peralatan Elektronik Rumah Tangga Berbasis Internet of Things Menggunakan Protokol MQTT,” *Jurnal BITe*, vol. 2, no. 1, pp. 1–11, 2020.
- [8] M. Z. Hasan dan E. Junianto, “Sistem Monitoring dan Kontrol Peralatan Listrik Berbasis IoT Menggunakan Aplikasi Blynk,” *E-Prosiding Teknik Informatika*, vol. 4, no. 2, pp. 401–413, 2023.

[9] A. R. Chadir *et al.*, “Evaluasi Pengujian Alat Pemberi Pakan Ikan Otomatis Berbasis IoT dengan Protokol MQTT,” *Jurnal Telematika*, vol. 19, no. 1, pp. 1–5, 2024.

[10] G. P. Yudhistira, D. Syauqy, dan B. H. Prasetio, “Sistem Pengenalan Gerakan Tangan Berbasis ESP32 dan Machine Learning untuk Kontrol Slide Power Point,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputasi*, vol. 9, no. 11, pp. 1–8, 2025.