

LAPORAN PRAKTIKUM

**Pengendalian Motor DC Berbasis IoT Menggunakan ESP32, Kit iMCLab, dan
Aplikasi IoT MQTT Panel**



Disusun Oleh:

Deva Arya Pradana (23081010067)

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR

2025

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi Internet of Things (IoT) memungkinkan berbagai perangkat fisik untuk saling terhubung dan dikendalikan melalui jaringan internet. Salah satu implementasi IoT yang banyak digunakan dalam bidang otomasi adalah pengendalian aktuator seperti motor DC secara jarak jauh. Dengan memanfaatkan mikrokontroler ESP32 yang memiliki modul WiFi terintegrasi, sistem kendali dapat dihubungkan dengan broker MQTT sebagai media komunikasi data.

Pada praktikum ini digunakan kit iMCLab yang dipadukan dengan aplikasi IoT MQTT Panel pada smartphone. Aplikasi tersebut berfungsi sebagai antarmuka pengguna (user interface) untuk mengirimkan perintah ON dan OFF ke perangkat ESP32 melalui protokol MQTT. Sistem ini diharapkan dapat memberikan pemahaman mengenai konsep dasar IoT, komunikasi MQTT, serta pengendalian perangkat keras secara real-time.

1.2 Rumusan Masalah

1. Bagaimana cara menghubungkan ESP32 dengan jaringan WiFi dan broker MQTT?
2. Bagaimana cara mengendalikan motor DC menggunakan perintah MQTT?
3. Bagaimana implementasi aplikasi IoT MQTT Panel sebagai pengendali jarak jauh?

1.3 Tujuan Praktikum

1. Mengimplementasikan komunikasi MQTT pada ESP32.
2. Mengendalikan motor DC menggunakan ESP32 dan driver motor pada kit iMCLab.
3. Menggunakan aplikasi IoT MQTT Panel sebagai media kontrol ON/OFF.

1.4 Manfaat Praktikum

Manfaat dari praktikum ini adalah menambah pemahaman mahasiswa mengenai konsep IoT, penggunaan protokol MQTT, serta integrasi antara perangkat lunak dan perangkat keras.

BAB II

DASAR TEORI

2.1 Internet of Things (IoT)

Internet of Things (IoT) merupakan konsep di mana objek fisik dapat saling berkomunikasi dan bertukar data melalui jaringan internet. Dalam implementasinya, IoT memanfaatkan sensor, aktuator, serta mikrokontroler sebagai pengendali utama.

2.2 ESP32

ESP32 adalah mikrokontroler berbasis WiFi dan Bluetooth yang banyak digunakan dalam proyek IoT. ESP32 memiliki keunggulan berupa kecepatan pemrosesan yang tinggi, konsumsi daya rendah, serta dukungan berbagai protokol komunikasi.

2.3 Protokol MQTT

MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi berbasis publish/subscribe yang ringan dan efisien. Protokol ini sangat cocok digunakan pada sistem IoT karena membutuhkan bandwidth yang kecil.

2.4 Motor DC dan PWM

Motor DC merupakan aktuator yang mengubah energi listrik menjadi energi mekanik. Kecepatan motor DC dapat dikontrol menggunakan teknik Pulse Width Modulation (PWM), yaitu dengan mengatur duty cycle sinyal.

2.5 Aplikasi IoT MQTT Panel

IoT MQTT Panel adalah aplikasi Android yang digunakan untuk mengirim dan menerima pesan MQTT. Aplikasi ini memungkinkan pengguna membuat tombol, switch, atau panel kontrol untuk mengendalikan perangkat IoT.

BAB III

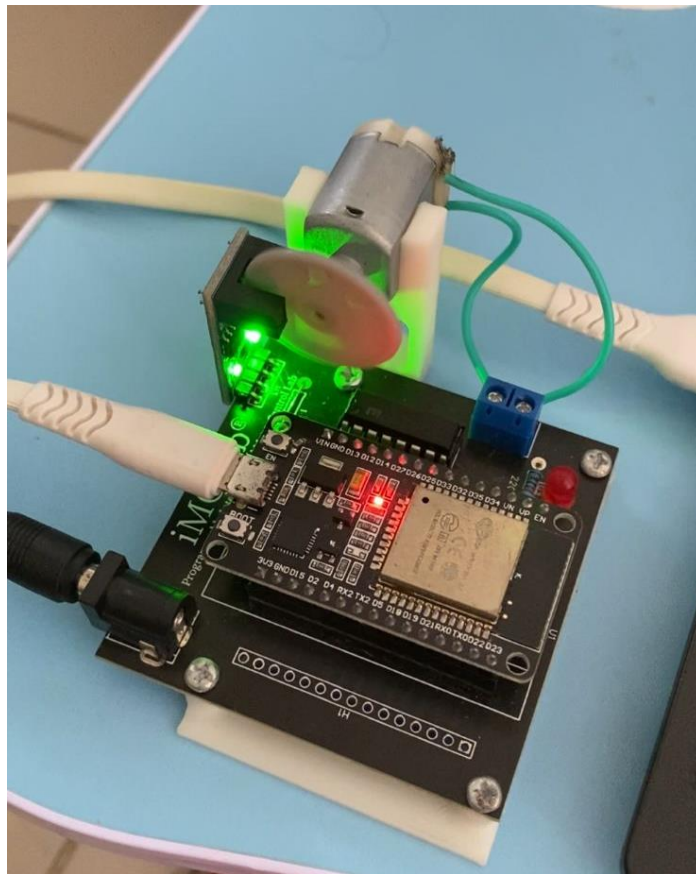
PERANCANGAN SISTEM

3.1 Diagram Blok Sistem

Sistem terdiri dari smartphone dengan aplikasi IoT MQTT Panel sebagai publisher, broker MQTT (broker.emqx.io) sebagai perantara komunikasi, dan ESP32 sebagai subscriber yang mengendalikan motor DC.

3.2 Perangkat Keras

- ESP32
- Kit iMCLab



Gambar 1. Kit iMCLab

- Motor DC
- Driver motor
- Smartphone.

3.3 Perangkat Lunak

- Arduino IDE
- Library WiFi.h
- Library PubSubClient.h
- IoT MQTT Panel.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Program

Program pada ESP32 diawali dengan konfigurasi pin, koneksi WiFi, serta koneksi ke broker MQTT. ESP32 melakukan subscribe pada topik arypadana untuk menerima perintah dari aplikasi.

Jika payload yang diterima bernilai '1', maka fungsi MotorOn() akan dijalankan sehingga motor DC berputar. Sebaliknya, jika payload bernilai '0', maka fungsi MotorOff() dijalankan dan motor akan berhenti.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>

int ledPin = 2;
char myData = 0;

int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 12;

// Setting PWM properties
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;

const byte pin_rpm = 13;
int volatile rev = 0;
//int rpm = 0;

const char* ssid = "DANERA 5G"; // Enter your WiFi name
const char* password = "danera1234"; // Enter WiFi password

#define mqttServer "broker.emqx.io"
#define mqttPort 1883

WiFiServer server(80);
WiFiClient espClient;
PubSubClient client(espClient);
```

```

String Topic;
String Payload;

// constants
const int baud = 115200;          // serial baud rate

void setup()
{
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);
    // sets the pins
    pinMode(motor1Pin1, OUTPUT);
    pinMode(motor1Pin2, OUTPUT);
    pinMode(enable1Pin, OUTPUT);
    pinMode(pin_rpm, INPUT_PULLUP);
    // pinMode(pin_rpm, INPUT);
    //attachInterrupt(digitalPinToInterrupt(pin_rpm), isr, RISING);

    // configure LED PWM functionalitites
    ledcSetup(pwmChannel, freq, resolution);

    // attach the channel to the GPIO to be controlled
    ledcAttachPin(enable1Pin, pwmChannel);

    // testing
    Serial.print("Testing DC Motor...");

    // Connect to WiFi network
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    // Connect to Server IoT (CloudMQTT)
    client.setServer(mqttServer, mqttPort);

```

```

client.setCallback(receivedCallback);

while (!client.connected()) {
    Serial.println("Connecting to Cloud IoT ...");

    // if (client.connect("ESP32Client", mqttUser, mqttPassword )) {
    if (client.connect("iMCLab On/Off")) {

        Serial.println("connected");
        Serial.print("Message received: ");

        } else {
            Serial.print("failed with state ");
            Serial.print(client.state());
            delay(2000);
        }
        client.subscribe("aryapradana");
    }
}

void MotorOn()
{
    // Move DC motor forward with increasing speed
    dutyCycle = 205;
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    ledcWrite(pwmChannel, dutyCycle);
}

void MotorOff()
{
    dutyCycle = 0;
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    ledcWrite(pwmChannel, dutyCycle);
}

void receivedCallback(char* topic, byte* payload, unsigned int length) {

    /* we got '1' -> MotorOn() */
    if ((char)payload[0] == '1') {
        MotorOn();
        Serial.println("Motor On");
    }
}

```



```

/* we got '0' -> Motoroff */
  if ((char)payload[0] == '0') {
    MotorOff();
    Serial.println("Motor Off");
  }
}

void loop()
{
  client.loop();

  myData = int(Serial.read());

  if (myData == '1'){
    Serial.println("LED is on !!!");
    digitalWrite(ledPin, HIGH);
    MotorOn();
  }
  else if (myData == '0'){
    Serial.println("LED is off !!!");
    digitalWrite(ledPin, LOW);
    MotorOff();
  }
}

```

Tabel 1. Kode program

4.2 Pengujian Sistem

Pengujian dilakukan dengan menekan tombol ON/OFF pada aplikasi IoT MQTT Panel. Hasil pengujian menunjukkan bahwa motor DC dapat menyala dan mati sesuai dengan perintah yang dikirimkan melalui MQTT secara real-time.

11.17

← Edit Panel

Panel name *
On Off

☐ Disable dashboard prefix topic

Topic *
aryapradana

Subscribe Topic
aryapradana

Payload on *
1

Payload off *
0

Switch color

☐ Use icon switch

☐ Payload is JSON Data

☐ Show received timestamp

☐ Show sent timestamp

☐ Confirm before publish

☐ Retain QoS 0

CANCEL SAVE

Gambar 2. Komponen buat kontrol on off



Gambar 3. Tampilan control aplikasi

4.3 Analisis Hasil

Berdasarkan hasil pengujian, sistem berjalan dengan baik selama koneksi WiFi dan broker MQTT stabil. Keterlambatan (delay) yang terjadi relatif kecil sehingga sistem layak digunakan sebagai media pembelajaran IoT.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa ESP32 berhasil dihubungkan dengan broker MQTT dan dapat mengendalikan motor DC menggunakan aplikasi IoT MQTT Panel. Sistem ini menunjukkan implementasi dasar IoT yang efektif dan mudah dipahami.

5.2 Saran

Pengembangan selanjutnya dapat dilakukan dengan menambahkan sensor, fitur monitoring kecepatan motor, serta sistem keamanan pada koneksi MQTT.

DAFTAR PUSTAKA

Kho, D. (2020). Pengantar Internet of Things.

MQTT.org. MQTT Protocol Specification.

Espressif Systems. ESP32 Technical Reference Manual.

EMQX. MQTT Broker Documentation.