**Tugas 1 IF4051 Pengembangan Sistem IoT**

**Semester II Tahun 2023/2024**

Tugas Mandiri : LED dan MQTT

A blue and white logo

Description automatically generated

**Disusun oleh:**

Roby Purnomo (13520106)

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung**

**Jl. Ganesha 10, Bandung 40132**

**2024**

# Kode Program

/\*

 \* Nama : Roby Purnomo

 \* NIM  : 13520106

 \*

 \* Blink

 \*/

#include <WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "Galaxy S919f1";

const char\* password = "123456789";

int FREQ = 10;

void setup\_wifi() {

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL\_CONNECTED) {

        delay(500);

        Serial.print(".");

    }

    Serial.println("");

    Serial.println("WiFi connected");

    Serial.println("IP address: ");

    Serial.println(WiFi.localIP());

}

const char\* mqtt\_server = "0.tcp.ap.ngrok.io";

const int mqtt\_port = 12471;

WiFiClient espClient;

PubSubClient client(espClient);

void reconnect() {

    while (!client.connected()) {

        Serial.print("Attempting MQTT connection...");

        if (client.connect("ESP32Client")) {

            Serial.println("connected");

            client.subscribe("esp32/act");

            String tempString = "13520106 >> FREQUENCY = " + String(FREQ);

            client.publish("esp32/freq", tempString.c\_str());

        } else {

            Serial.print("failed, rc=");

            Serial.print(client.state());

            Serial.println(" try again in 5 seconds");

            delay(5000);

        }

    }

}

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

    Serial.print("Message arrived [");

    Serial.print(topic);

    Serial.print("] ");

    for (int i = 0; i < length; i++) {

        Serial.print((char)payload[i]);

    }

    Serial.println();

}

#define LED 2

#define BUTTON\_PIN 0

// Variables

int lastState = 1;     // last state from the input button

int currentState;      // current state from the input button

void setup() {

    Serial.begin(9600);

    setup\_wifi();

    client.setServer(mqtt\_server, mqtt\_port);

    client.setCallback(callback);

    reconnect();

    // Set pin mode

    pinMode(LED,OUTPUT);

    // initialize the push button pin as an input

    pinMode(BUTTON\_PIN, INPUT\_PULLUP);

}

void loop() {

    if (!client.connected()) {

        reconnect();

    }

    client.loop();

    currentState = digitalRead(BUTTON\_PIN);

    if (lastState == LOW && currentState == HIGH) {

      FREQ = FREQ + 1;

      String tempString = "13520106 >> FREQUENCY = " + String(FREQ);

      client.publish("esp32/freq", tempString.c\_str());

    }

    // save the the last state

    lastState = currentState;

    Serial.print("FREQUENCY : ");

    Serial.println(FREQ);

    delay(500/FREQ);

    digitalWrite(LED,HIGH);

    delay(500/FREQ);

    digitalWrite(LED,LOW);

}

Penjelasan :

Pada program tersebut saya menggunakan localhost sebagai broker tetapi karena terkendala berkali-kali, oleh karena itu saya menggunakan [**ngrok**](https://ngrok.com/) sebagai bridge yang menghubungkannya dengan localhost saya, yakni dengan **ip "0.tcp.ap.ngrok.io"** dan **port 12471** seperti yang tertera pada command line di bawah ini.

A screenshot of a computer

Description automatically generated

# Hasil

Hasil yang didapatkan pada subscriber “esp32/freq” adalah sebagai berikut.

A screenshot of a computer

Description automatically generated

# Video Demonstrasi

Untuk lebih detail demonstrasinya dapat dilihat pada link video berikut ini.

<https://youtu.be/faYMnZXmiNk>