Nama : Aditya Mahardika Pratama

Nrp : 152022227

Kelas : AA

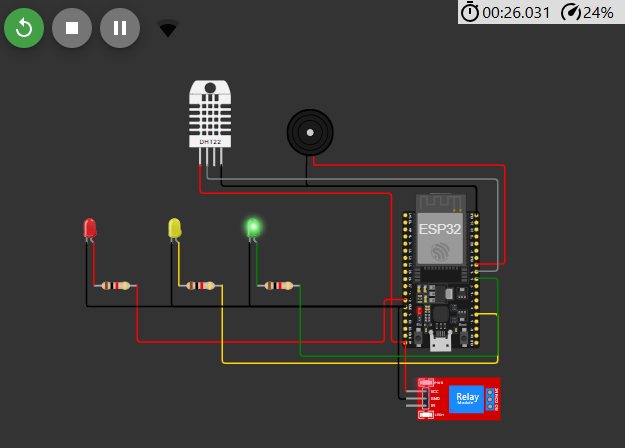
Mata Kuliah : Pemrograman iot

**1A.**

1. Akuisisi Data dengan Sensor
   * Sensor mengumpulkan data seperti suhu, kelembaban, dan pergerakan dari lingkungan dan mengubahnya menjadi sinyal digital yang dapat dibaca oleh mikrokontroler (ESP32, Arduino, Raspberry Pi, dll).
2. Mengirim Data ke Server Menggunakan MQTT
   * MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi yang dirancang untuk transfer data ringan antara perangkat yang menggunakan jaringan yang terbatas. Protokol ini cocok untuk komunikasi IoT yang real-time dan hemat daya.
   * Mikrokontroler yang telah mengambil data sensor mengirimkan data tersebut ke server melalui MQTT broker.
   * MQTT menggunakan konsep publisher, subscriber, dan broker. Mikrokontroler diatur sebagai publisher yang mengirimkan data sensor ke topik tertentu di broker MQTT. Server kemudian dapat berlangganan (subscribe) ke topik yang sama untuk menerima data yang dikirim oleh sensor.
   * Dengan mekanisme publish-subscribe, server dapat menerima data dari beberapa perangkat sensor secara real-time dengan latensi rendah.
3. Menyimpan Data dalam
   * Database Setelah server menerima data dari broker MQTT, server dapat menyimpannya dalam database yang sesuai (misalnya MySQL, PostgreSQL, atau MongoDB). Data ini disimpan untuk diambil oleh aplikasi atau untuk analisis nanti.
4. Aplikasi Mobile (Frontend - Flutter) Mengambil Data dari Server
   * Aplikasi Flutter dapat mengakses data dari server menggunakan API atau WebSockets yang disediakan server. Hal ini memungkinkan data dari sensor ditampilkan sesuai permintaan atau secara real-time di aplikasi Anda..
5. Menampilkan Data di Antarmuka Pengguna (UI)
   * Setelah menerima data dari server, aplikasi Flutter menampilkan data ini dalam bagan, tabel, atau komponen visual lainnya untuk memudahkan interpretasi data pengguna.

**1B.**

1. Wokwi



1. Code

1. #include "DHTesp.h"

2. #include "PubSubClient.h"

3. #include "WiFi.h"

4.

5. #define DHTPIN 18           // Pin DHT

6. #define DHTTYPE DHT22       // Tipe sensor DHT (updated to DHT22)

7. #define LED\_GREEN 5         // Pin LED Hijau

8. #define LED\_YELLOW 2        // Pin LED Kuning

9. #define LED\_RED 12          // Pin LED Merah

10. #define RELAY\_PIN 17        // Pin Relay Pompa

11. #define BUZZER\_PIN 19       // Pin Buzzer

12.

13. DHTesp dhtSensor;

14.

15. const char\* ssid = "Wokwi-GUEST";

16. const char\* pass = "";

17. const char\* mqtt\_server = "broker.hivemq.com"; // Alamat Broker MQTT

18.

19. WiFiClient wifiClient;

20. PubSubClient mqtt(wifiClient);

21.

22. void setup() {

23.   // Inisialisasi Serial Monitor

24.   Serial.begin(115200);

25.

26.   // Inisialisasi DHT sensor

27.   dhtSensor.setup(DHTPIN, DHTesp::DHT22);

28.

29.   // Konfigurasi pin untuk LED, Relay, dan Buzzer

30.   pinMode(LED\_GREEN, OUTPUT);

31.   pinMode(LED\_YELLOW, OUTPUT);

32.   pinMode(LED\_RED, OUTPUT);

33.   pinMode(RELAY\_PIN, OUTPUT);

34.   pinMode(BUZZER\_PIN, OUTPUT);

35.

36.   // Koneksi ke WiFi

37.   setup\_wifi();

38.

39.   // Set server MQTT

40.   mqtt.setServer(mqtt\_server, 1883);

41. }

42.

43. void setup\_wifi() {

44.   delay(10);

45.   Serial.println("Connecting to WiFi...");

46.   WiFi.begin(ssid, pass);

47.

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

49.     delay(500);

50.     Serial.print(".");

51.   }

52.   Serial.println("Connected to WiFi!");

53. }

54.

55. void reconnect() {

56.   // Loop sampai berhasil terkoneksi ke MQTT

57.   while (!mqtt.connected()) {

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

59.     if (mqtt.connect("ESP32Client")) {

60.       Serial.println("Connected to MQTT broker");

61.     } else {

62.       Serial.print("Failed, rc=");

63.       Serial.print(mqtt.state());

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

65.       delay(5000);

66.     }

67.   }

68. }

69.

70. void loop() {

71.   if (!mqtt.connected()) {

72.     reconnect();

73.   }

74.   mqtt.loop();

75.

76.   // Membaca data suhu dan kelembapan

77.   TempAndHumidity data = dhtSensor.getTempAndHumidity();

78.   float temperature = data.temperature;

79.   float humidity = data.humidity;

80.

81.   // Mengontrol pompa berdasarkan kelembapan

82.   bool pumpStatus = false;

83.   if (humidity < 50) {

84.     digitalWrite(RELAY\_PIN, HIGH);  // Menyalakan Pompa

85.     pumpStatus = true;

86.   } else {

87.     digitalWrite(RELAY\_PIN, LOW);   // Mematikan Pompa

88.     pumpStatus = false;

89.   }

90.

91.   // Mengontrol LED dan Buzzer berdasarkan suhu

92.   if (temperature > 35) {

93.     digitalWrite(LED\_RED, HIGH);    // Menyalakan LED Merah

94.     digitalWrite(LED\_YELLOW, LOW);

95.     digitalWrite(LED\_GREEN, LOW);

96.     digitalWrite(BUZZER\_PIN, HIGH); // Menyalakan Buzzer

97.   } else if (temperature >= 30 && temperature <= 35) {

98.     digitalWrite(LED\_YELLOW, HIGH); // Menyalakan LED Kuning

99.     digitalWrite(LED\_RED, LOW);

100.     digitalWrite(LED\_GREEN, LOW);

101.     digitalWrite(BUZZER\_PIN, LOW);  // Mematikan Buzzer

102.   } else {

103.     digitalWrite(LED\_GREEN, HIGH);  // Menyalakan LED Hijau

104.     digitalWrite(LED\_RED, LOW);

105.     digitalWrite(LED\_YELLOW, LOW);

106.     digitalWrite(BUZZER\_PIN, LOW);  // Mematikan Buzzer

107.   }

108.

109.   // Mengirim data ke MQTT jika data valid

110.   if (!isnan(temperature) && !isnan(humidity)) {

111.     char payload[100]; // Sesuaikan ukuran buffer sesuai kebutuhan

112.     snprintf(payload, sizeof(payload), "{\"temperature\":%.2f,\"humidity\":%.2f,\"pump\":\"%s\"}", temperature, humidity, pumpStatus ? "ON" : "OFF");

113.

114.     mqtt.publish("blacksun", payload);

115.

116.     Serial.print("Data sent to MQTT: ");

117.     Serial.println(payload);

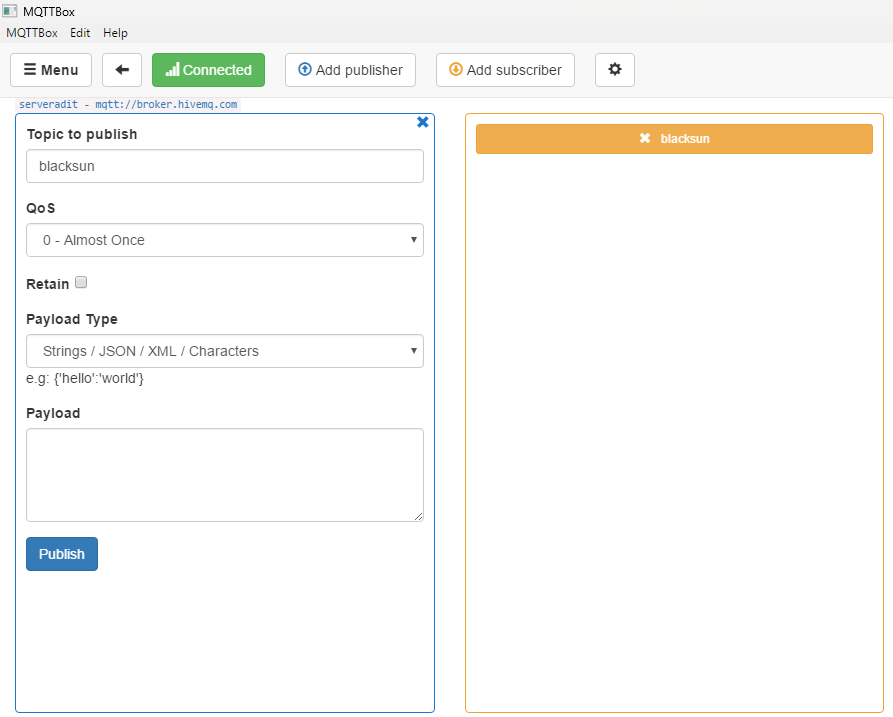
118.   }

119.

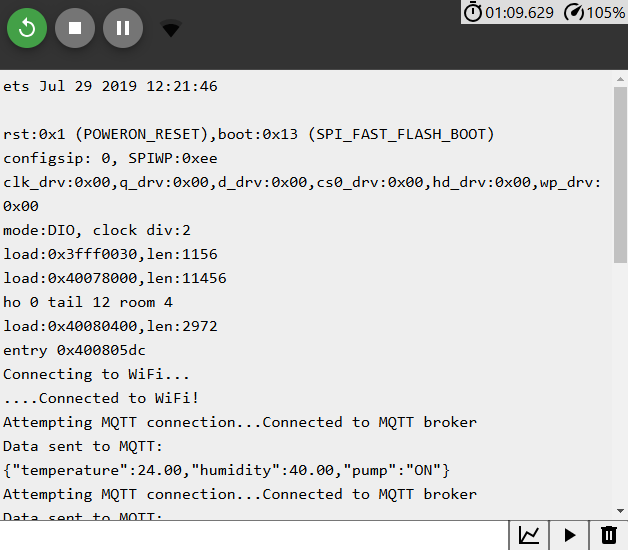
120.   delay(2000); // Delay 2 detik sebelum loop berikutnya

121. }

1. MQTT



1. Output



**2A.**

1. Akuisisi Data dengan Sensor