

MODUL 4

APLIKASI INTERNET OF THINGS MENGGUNAKAN PROTOKOL MQTT (Lanjutan)



CAPAIAN PEMBELAJARAN

1. Mampu melakukan komunikasi antar arduino dengan protokol MQTT



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Hardware:

1. Modul Dev ESP32
2. Software MQTT.Fix (<https://web.archive.org/web/20210514230412/https://www.jensd.de/apps/mqttfx/1.7.1/>)
3. LED x 4
4. resistor 100 Ohm x 4
5. potensiometer x1

Software:

1. Arduino IDE (<http://arduino.cc>)

Lain-lain:

1. koneksi internet via Wifi



DASAR TEORI

Client dapat melakukan subscribe ke beberapa topik dengan melakukan subscribe satu persatu atau menggunakan karakter Wildcard. Karakter Wildcard ada dua macam yaitu

Wildcard satu tingkat dan Wildcard banyak tingkat. Wildcard satu tingkat menggunakan karakter + (tambah) dan Wildcard banyak tingkat menggunakan karakter * (bintang). Karakter Wildcard hanya boleh digunakan untuk client yang subscribe dan tidak diperbolehkan untuk client yang publish. Contoh karakter Wildcard :

Wildcard satu tingkat:

topik= yogya/sensor/+

Client yang subscribe ke topik di atas akan subscribe di topik berikut ini.

yogya/sensor/suhu

yogya/sensor/kelembaban

yogya/sensor/co2

yogya/sensor/gempa

Wildcard banyak tingkat:

topik= yogya/*

Client yang subscribe ke topik di atas akan subscribe di topik berikut ini.

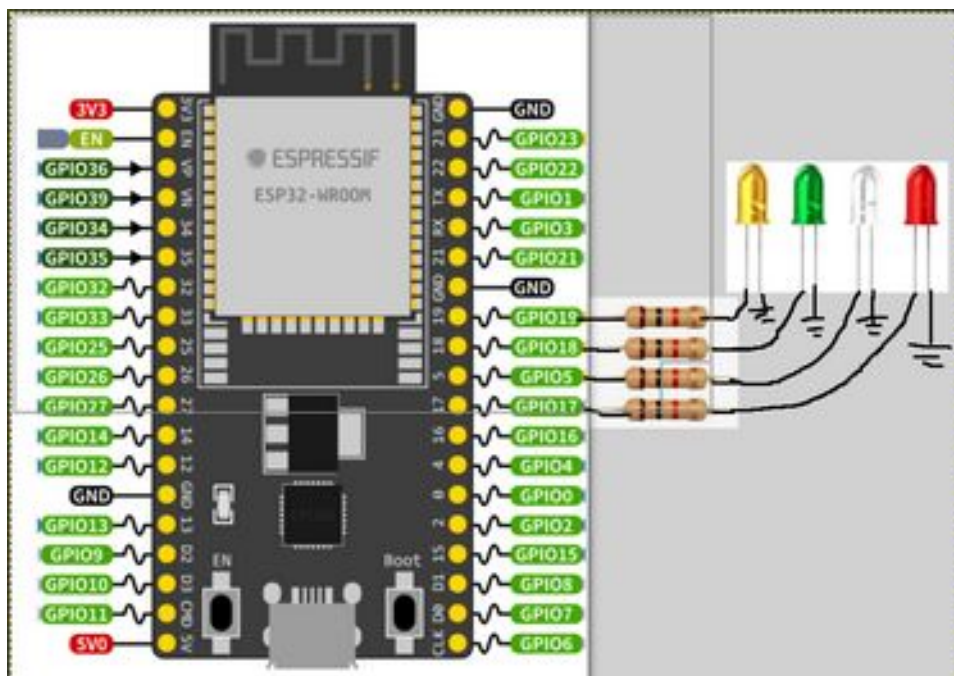
yogya/sensor/suhu

yogya/sensor/kelembaban

yogya/triffict/janti

yogya/triffict/demangan

PRAKTIK



Gambar 2.

A. Memproses Lebih dari Satu Subscribe

1. Buat rangkaian seperti Gambar 2.

2. Tulis program berikut ini dan upload terus dievaluasi. Coba dari Mqtt.Fix untuk publish sesuai Subscribe pada method reconnect();

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

#define LED1 19
#define LED2 18
#define LED3 5
#define LED4 17

const char* ssid = "harjolukito";
const char* password = "ami1971da";
const char* mqtt_server = "broker.mqtt-dashboard.com";

WiFiClient espClient;
PubSubClient client(espClient);
char msg[50];

String tpk=String(50);

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

```

void callback(char* topic, byte* payload, unsigned int
length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.println("] ");
    tpk=topic;
    for(int i=0;i<length;i++){
        Serial.print(payload[i]);
    }
    if(tpk=="yogya/utara/lampu"){
        switch (char(payload[0])) {
            case '1': if ((char)payload[1] == '1') {
                digitalWrite(LED1, HIGH);
            } else if ((char)payload[1] == '0'){
                digitalWrite(LED1, LOW);
            }
            break;
            case '2': if ((char)payload[1] == '1') {
                digitalWrite(LED2, HIGH);
            } else if ((char)payload[1] == '0'){
                digitalWrite(LED2, LOW);
            }
            break;
        }
    }
    if(tpk=="yogya/timur/lampu"){
        switch (char(payload[0])) {
            case '1': if ((char)payload[1] == '1') {
                digitalWrite(LED3, HIGH);
            } else if ((char)payload[1] == '0'){
                digitalWrite(LED3, LOW);
            }
            break;
            case '2': if ((char)payload[1] == '1') {
                digitalWrite(LED4, HIGH);
            } else if ((char)payload[1] == '0'){
                digitalWrite(LED4, LOW);
            }
        }
    }
}

```

```

        break;
    }
}

}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        String clientId = "ESP32Client-";
        clientId += String(random(0xffff), HEX);
        if (client.connect(clientId.c_str())) {
            Serial.println("connected");
            client.subscribe("yogya/utara/lampu");
            client.subscribe("yogya/timur/lampu");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

void setup() {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

void loop() {

    if (!client.connected()) {

```

```

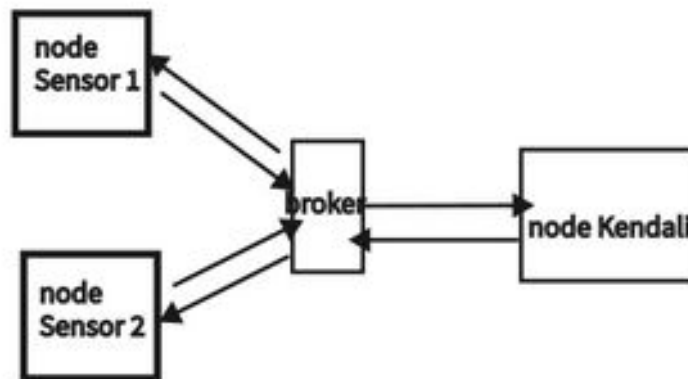
    reconnect();
}
client.loop();
}

```



LATIHAN

Kerjakan berdua.



Gambar 2.

Buat sistem seperti pada Gambar 2. Rangkaian nodeSensor1 dan nodeSensor2 seperti pada Gambar 1. nodeSensor1 dan nodeSensor2 diatur nyala dan padamnya LED dari node Kendali. Setiap diberi perintah dari nodeKendali nodeSensor memberikan respon balik.



TUGAS

1. Buat diagram alir program di client

1. Lanjutkan latihannya.



REFERENSI

1. 1. Neil Kolban, Kolban's Book on ESP8266, 2016

2. <https://github.com/knolleary/pubsubclient>

3. <https://web.archive.org/web/20210514230412/https://www.jensd.de/apps/mqttfx/1.7.1/>