

MODUL 3

APLIKASI INTERNET OF THINGS MENGGUNAKAN PROTOKOL MQTT



CAPAIAN PEMBELAJARAN

1. Mampu mempraktekkan konsep Publish dan Subscribe
2. Mampu membuat mempraktekkan program Publish dan Subscribe di Arduino



KEBUTUHAN ALAT/BAHAN/SOFTWARE

Hardware:

1. Modul Dev ESP32
2. Software MQTT.Fix (<https://web.archive.org/web/20210514230412/https://www.jensd.de/apps/mqttx/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:

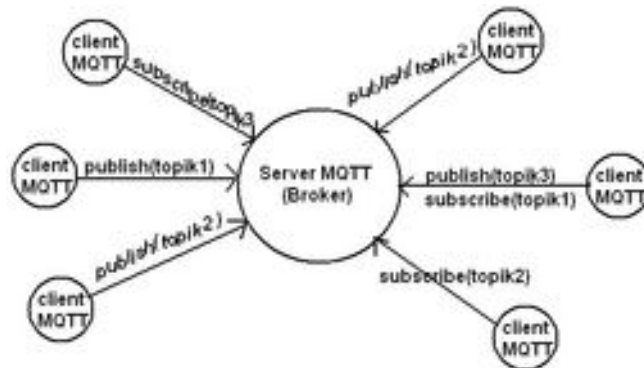
1. koneksi internet via Wifi



DASAR TEORI

Salah satu protokol yang digunakan sebagai platform Internet of Things adalah MQTT. Protokol *Message Queue Telemetry Transport* (MQTT) adalah protokol

pesan yang sangat sederhana dan ringan. Protokol MQTT menggunakan arsitektur *publish/subscribe* yang dirancang secara terbuka dan mudah untuk diimplementasikan, yang mampu menangani ribuan *client* jarak jauh dengan hanya satu server. MQTT meminimalkan bandwidth jaringan dan kebutuhan sumber daya perangkat ketika mencoba untuk menjamin kehandalan dan pengiriman. Pendekatan ini membuat protokol MQTT sangat cocok untuk menghubungkan mesin ke mesin (M2M), merupakan aspek penting dari konsep *Internet of things* [2].



Gambar Arsitektur MQTT

PRAKTIK

A. Persiapan

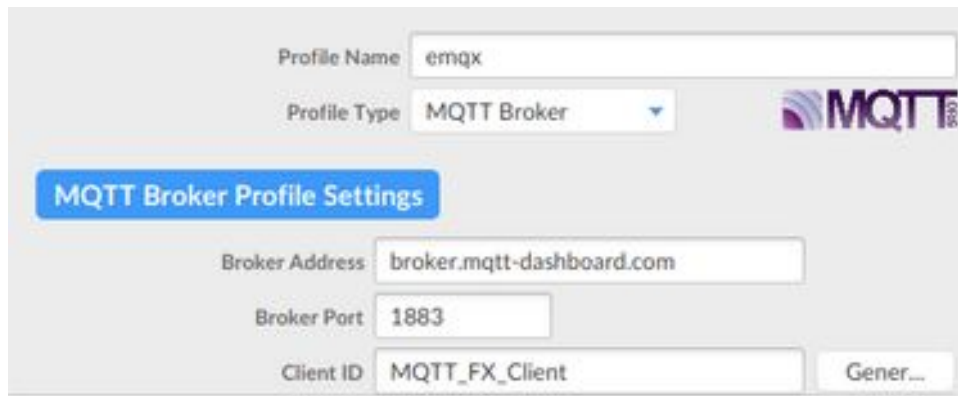
1. Download aplikasi MQTT.Fix (<https://web.archive.org/web/20210514230412/https://www.jensd.de/apps/mqttfx/1.7.1/>)
2. Install MQTT.Fix
3. Tambahkan library Arduino PubSubClient melalui Manager Library

B. Publish dan Subcribe Menggunakan MQTT.Fix

1. Jalankan MQTT.Fix
2. Atur Server MQTT (Broker) yang akan digunakan dengan menekan gambar gear.



3. Isikan server broker dll seperti pada Gambar berikut ini.



Profile Name: emqx

Profile Type: MQTT Broker

MQTT Broker Profile Settings

Broker Address: broker.mqtt-dashboard.com

Broker Port: 1883

Client ID: MQTT_FX_Client

Gener...

4. Tekan tombol OK
5. Tekan Tombol Connect agar terkoneksi dengan server MQTT
6. Bekerjalah berpasangan dengan teman.
7. Tekan menu Publish dan isikan di kotak Text edit sebuah topic seperti gambar berikut ini.

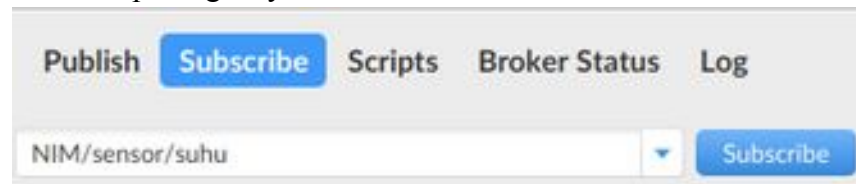


Publish Subscribe Scripts Broker Status Log

Switch Fountain » NIM/sensor/suhu

Publish

8. Untuk teman pasangannya tekan menu Subscribe.



Publish **Subscribe** Scripts Broker Status Log

NIM/sensor/suhu

Subscribe

9. Untuk yang Publish, ketik angka atau kalimat di text area kemudian tekan tombol Publish.
10. Untuk yang Subscribe perhatikan tampilan console di bawah.
11. Anda uji coba dengan melakukan Publish topic temannya yang lain dan Subscribe temannya yang lain.

C. Modul Dev ESP32 Publish ke Server MQTT

1. Siapkan modul Dev ESP32
2. Buat rangkaian seperti program berikut ini
3. Copy dan upload program berikut ini.

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

#define pinAnalog 4;

const char* ssid = "ssidnya";
```

```

const char* password = "passwordnya";
const char* mqtt_server = "test.mosquitto.org";

WiFiClient espClient;
PubSubClient client(espClient);

char msg[50];
long wktYll;
int dataAnalogAsal;

void setup_wifi() {
  Serial.println();
  Serial.print("Koneksi ke ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  delay(1000);
  if (WiFi.status() != WL_CONNECTED) {
    delay(500);
    WiFi.begin(ssid,password);
  }else if (WiFi.status()==WL_CONNECTED) {
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
  }
}

void reconnectMQTT() {
  if(WiFi.status()==WL_CONNECTED){
    while (!client.connected()) {
      Serial.print("Attempting MQTT connection...");
      String clientId = "ESP8266Client-";
      clientId += String(random(0xffff), HEX);
      if (client.connect(clientId.c_str())) {
        Serial.println("connected");
      } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" Coba 5 detik lagi");
        delay(5000);
      }
    }
  }
}

```

```

    }
}

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
}

void loop() {
    long now = millis();
    while(WiFi.status() != WL_CONNECTED) {
        WiFi.begin(ssid, password);
        delay(1000);
        if(WiFi.status() == WL_CONNECTED) {
            Serial.println("Wifi tersambung");
        }
    }

    if (!client.connected()) {
        reconnectMQTT();
    }

    if (now - wktYll > 2000) {
        wktYll = now;
        dataAnalogAsal = analogRead(pinAnalog);
        snprintf (msg, 50, "Nilai: %f", dataAnalogAsal);

        if(WiFi.status() == WL_CONNECTED) {
            Serial.print("Publish message: ");
            Serial.println(msg);
            client.publish("utdi/NIM981106", msg);
        } else {
            Serial.println("Wifi terputus");
        }
    }
}

```

4. Sesuaikan MQTT.Fix subscribenya dengan Publish dari nodeESP32
5. Dan amati data yang di terima.
6. Coba dengan server/broker yang lainnya.

D. Modul ESP32 Subscribe ke Server MQTT

1. Copy dan upload program berikut ini.

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

#define LED1 22

const char* ssid = "harjolukito";
const char* password = "ami1971da";
const char* mqtt_server = "test.mosquitto.org";

WiFiClient espClient;
PubSubClient client(espClient);

char msg[50];
long wktYll;
int dataAnalogAsal;

void setup_wifi() {
  Serial.println();
  Serial.print("Koneksi ke ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  delay(1000);
  if (WiFi.status() != WL_CONNECTED) {
    delay(500);
    WiFi.begin(ssid,password);
  }else if (WiFi.status()==WL_CONNECTED) {
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
  }
}

void reconnectMQTT() {
  if(WiFi.status()==WL_CONNECTED){
    while (!client.connected()) {
      Serial.print("Attempting MQTT connection...");
      String clientId = "ESP8266Client-";
      clientId += String(random(0xffff), HEX);
```

```

        if (client.connect(clientId.c_str())) {
            Serial.println("connected");
            client.subscribe("UTDI/SUB981106");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" Coba 5 detik lagi");
            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();

    if ((char)payload[0] == '1') {
        Serial.println("Data satu");
        digitalWrite(LED1, HIGH);
    } else if((char)payload[0] == '0'){
        Serial.println("Data no Satu");
        digitalWrite(LED2, LOW); // Turn the LED off by
making the voltage HIGH
    }

}

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

```

```

void loop() {
    client.loop();
    while(WiFi.status() != WL_CONNECTED) {
        WiFi.begin(ssid,password);
        delay(1000);
        if(WiFi.status() == WL_CONNECTED) {
            Serial.println("Wifi tersambung");
        }
    }

    if (!client.connected()) {
        reconnectMQTT();
    }
}

```

2. Lakukan publish via MQTTFx untuk mengendalikan LED1.



LATIHAN

1. Dikerjakan berdua. Buat aplikasi nodeSensorCahaya dan nodeWarning. NodeSensor membaca nilai dari masukan analog yang dipublish menggunakan protokol MQTT. NodeWarning akan menerima data dari nodeSensor melalui protokol MQTT. LED akan berkedip jika data yang diterima melebihi 1000.
2. Gambarkan diagram sistem dari soal 1.

TUGAS

1. Buat diagram alir nodeSensor dan nodeWarning

REFERENSI

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/>