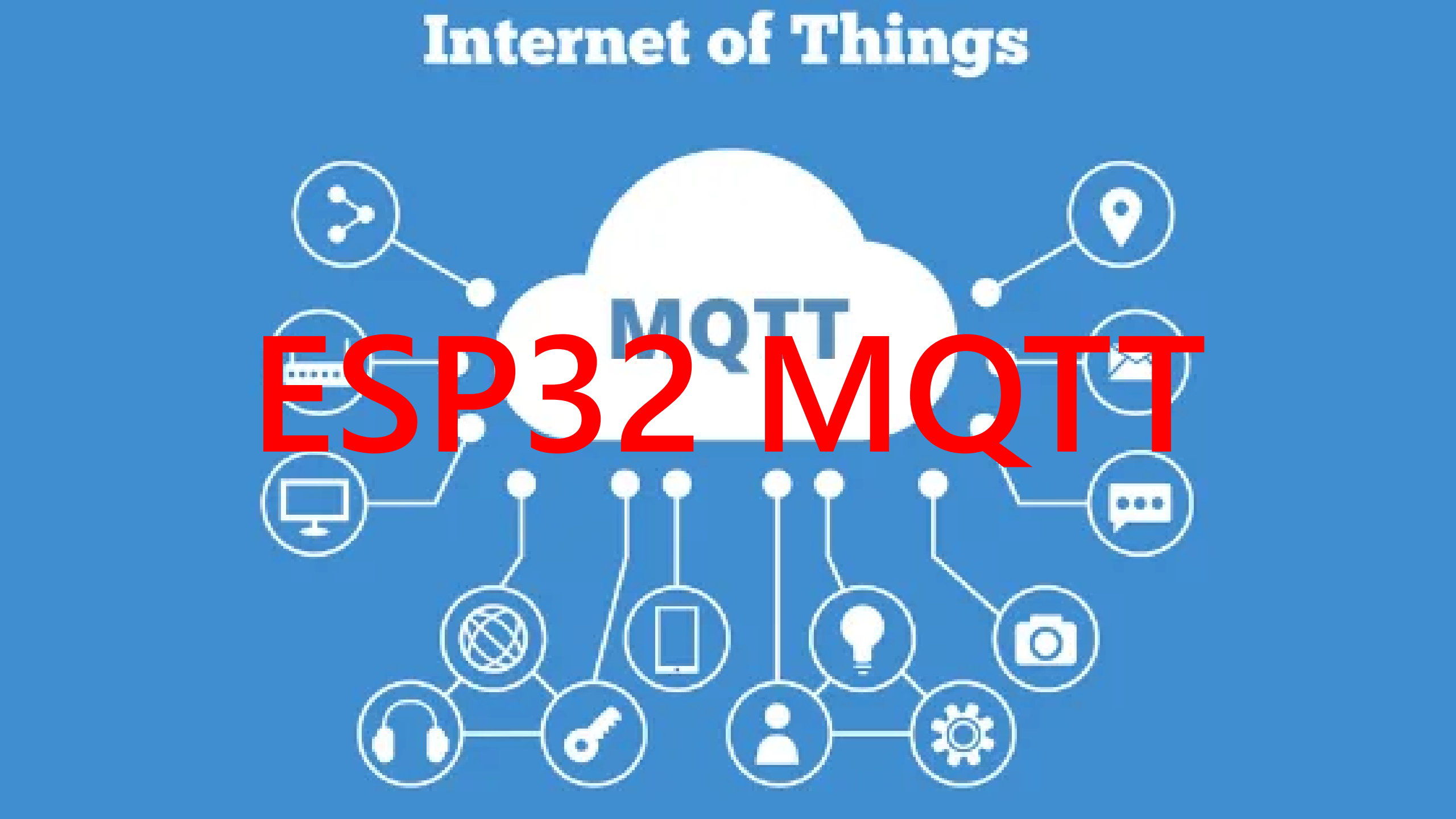

Internet of Things
MQTT
ESP32 MQTT

Intro

- ESP32 : small WiFi MQTT node
- Realtime data, full control
- MQTT Publish Subscribe via https://github.com/plapointe6/EspMQTTClient :
  - Connecting to a WiFi network.
  - Connecting to a MQTT broker.
  - Automatically detecting connection lost either from the - WiFi client or the MQTT broker and it will retry a connection automatically.
  - Subscribing/unsubscribing to/from MQTT topics by a friendly callback system.

Intro

- MQTT Publish Subscribe via https://github.com/plapointe6/EspMQTTClient :
  - Supports wildcards (+, #) in subscriptions
  - Provide a callback handling to advise once everything is connected (Wifi and MQTT).
  - Provide a function to enable printing of useful debug information related to MQTT and Wifi connections.
  - Provide some other useful utilities for MQTT and Wifi management.
  - Provide a function to enable an HTTP Update server secured by a password to allow remote update.
  - Provide a function to enable OTA secured by a password to allow remote update.
  - Based on PubSubClient : https://github.com/knolleary/pubsubclient

Tujuan

- Menginstalasi Library ESP MQTT Client ESP32
- Melakukan proses transfer data via MQTT dengan ESP32

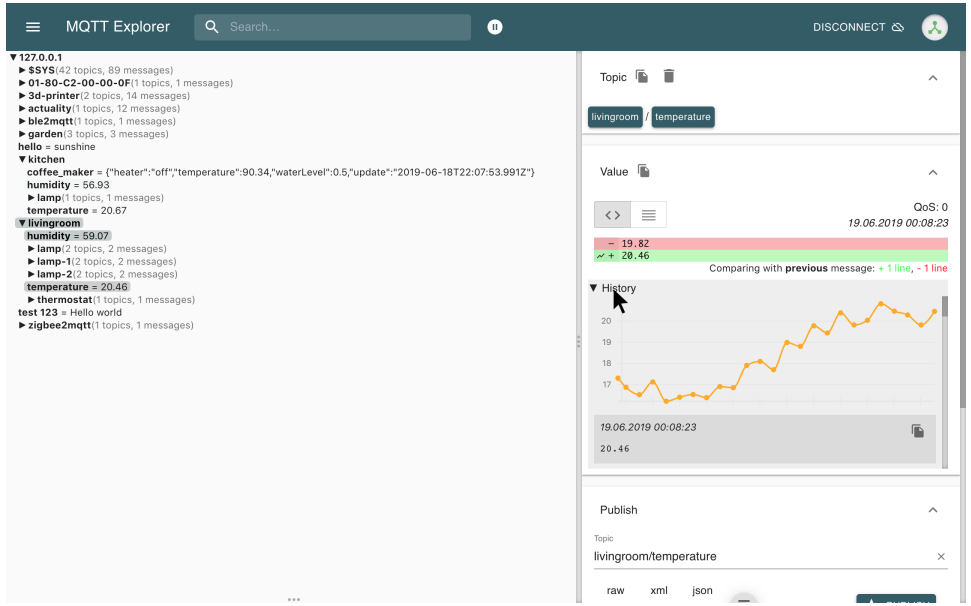
Alat dan bahan

- Komputer (desktop, laptop)
- Modul ESP32
- Software :
    - MQTT Explorer (http://mqtt-explorer.com/)

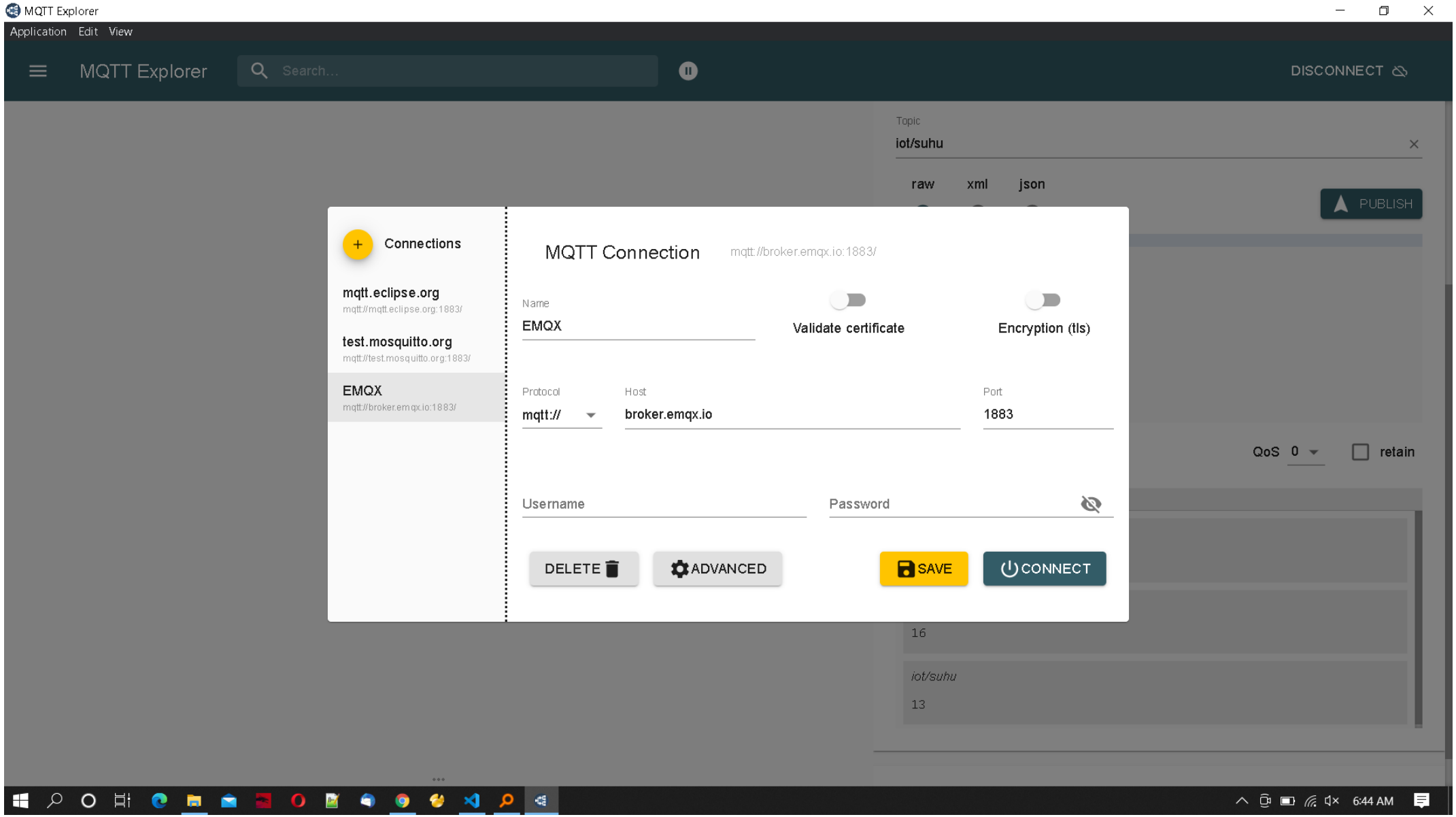
Percobaan 1

- **Tujuan** :
  - Menginstall MQTT Explorer
  - Subscribe dan Publish via MQTT Explorer
- **Langkah** :

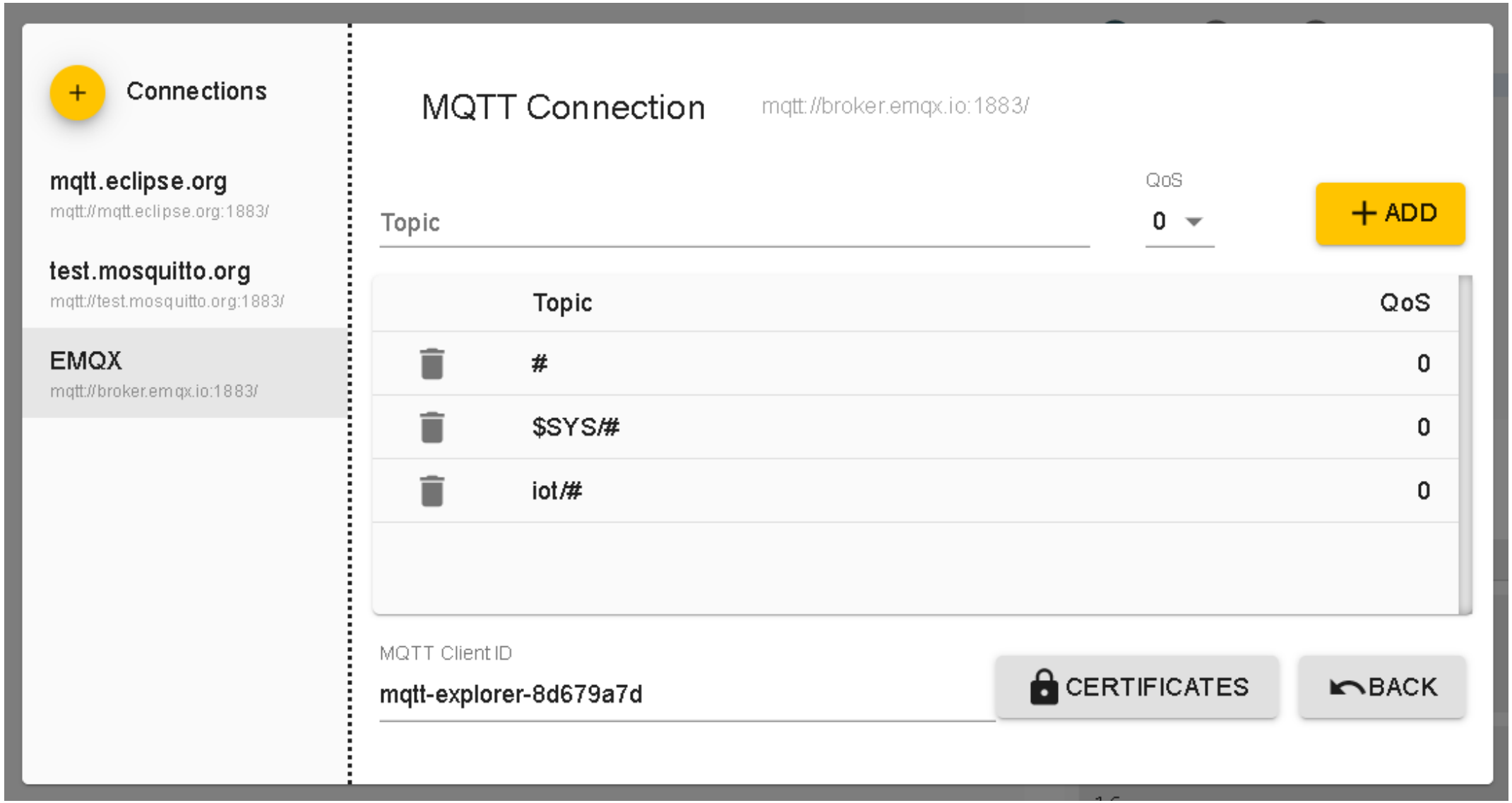
1. Download MQTT Explorer dari http://mqtt-explorer.com/

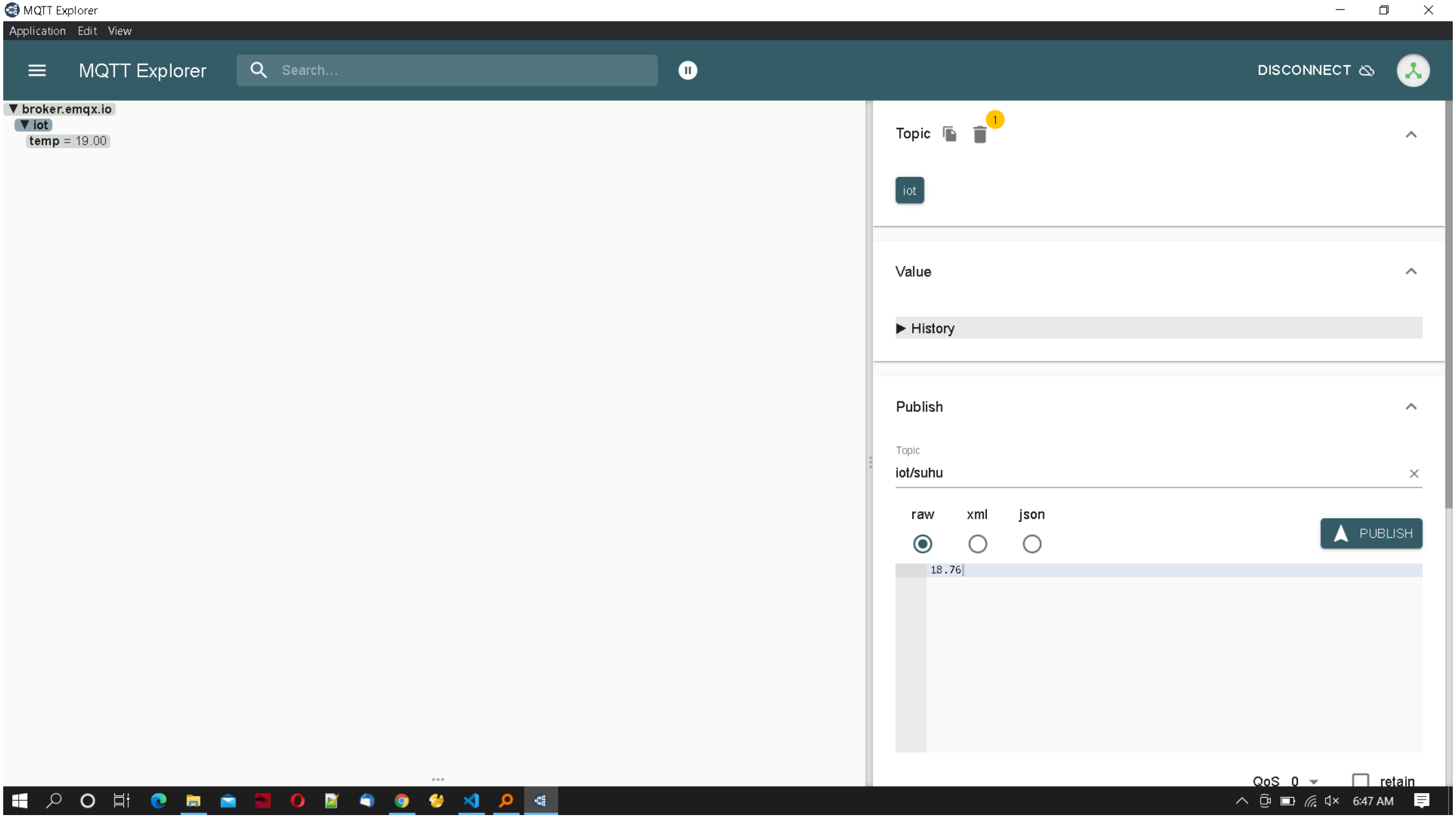


2. Run As Administrator file yang sudah di-download dan koneksikan ke EMQX

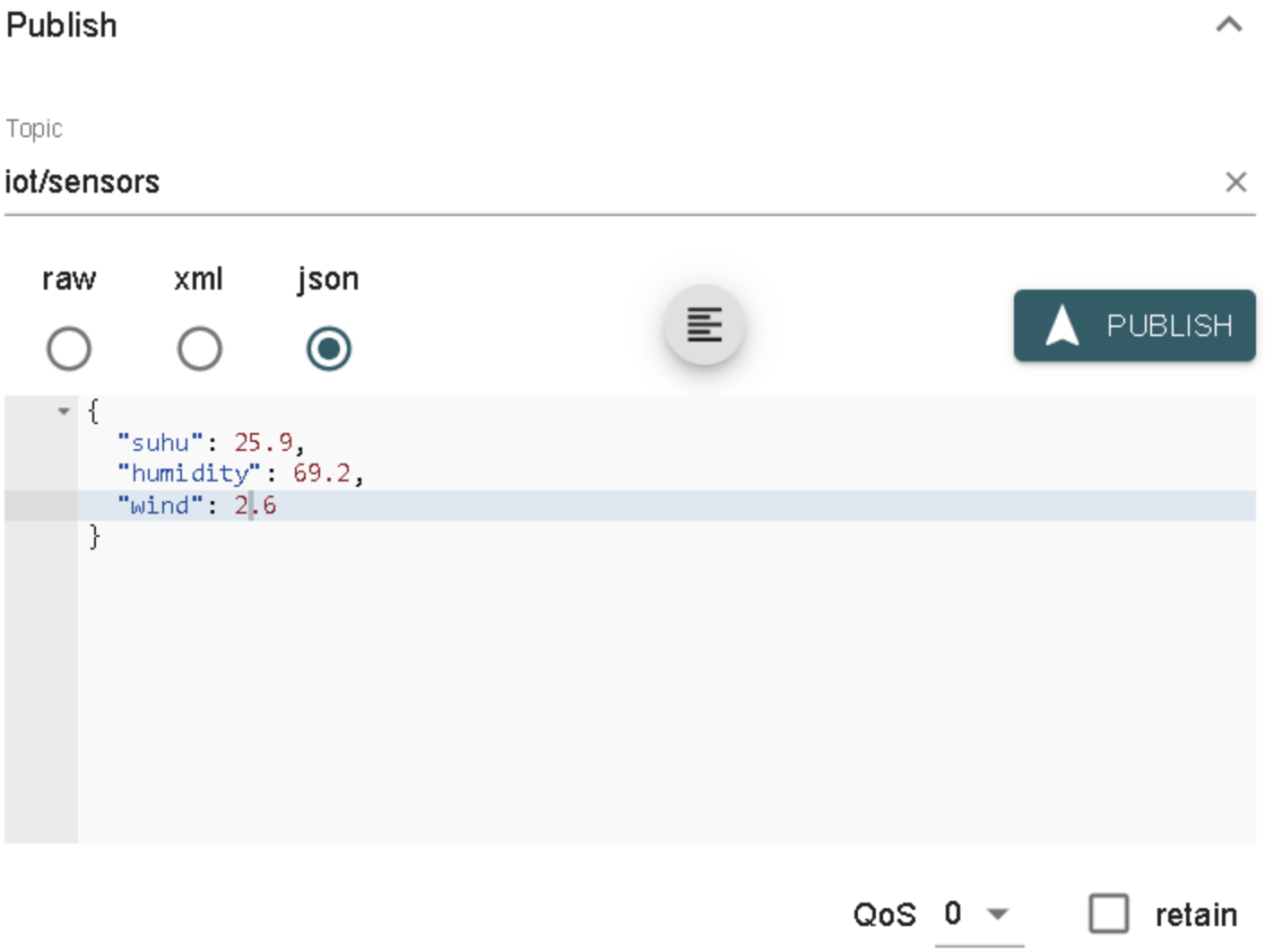


3. Tambahkan topik **iot/#** pada bagian **Advance**. Klik **Back** lalu klin **Connect**

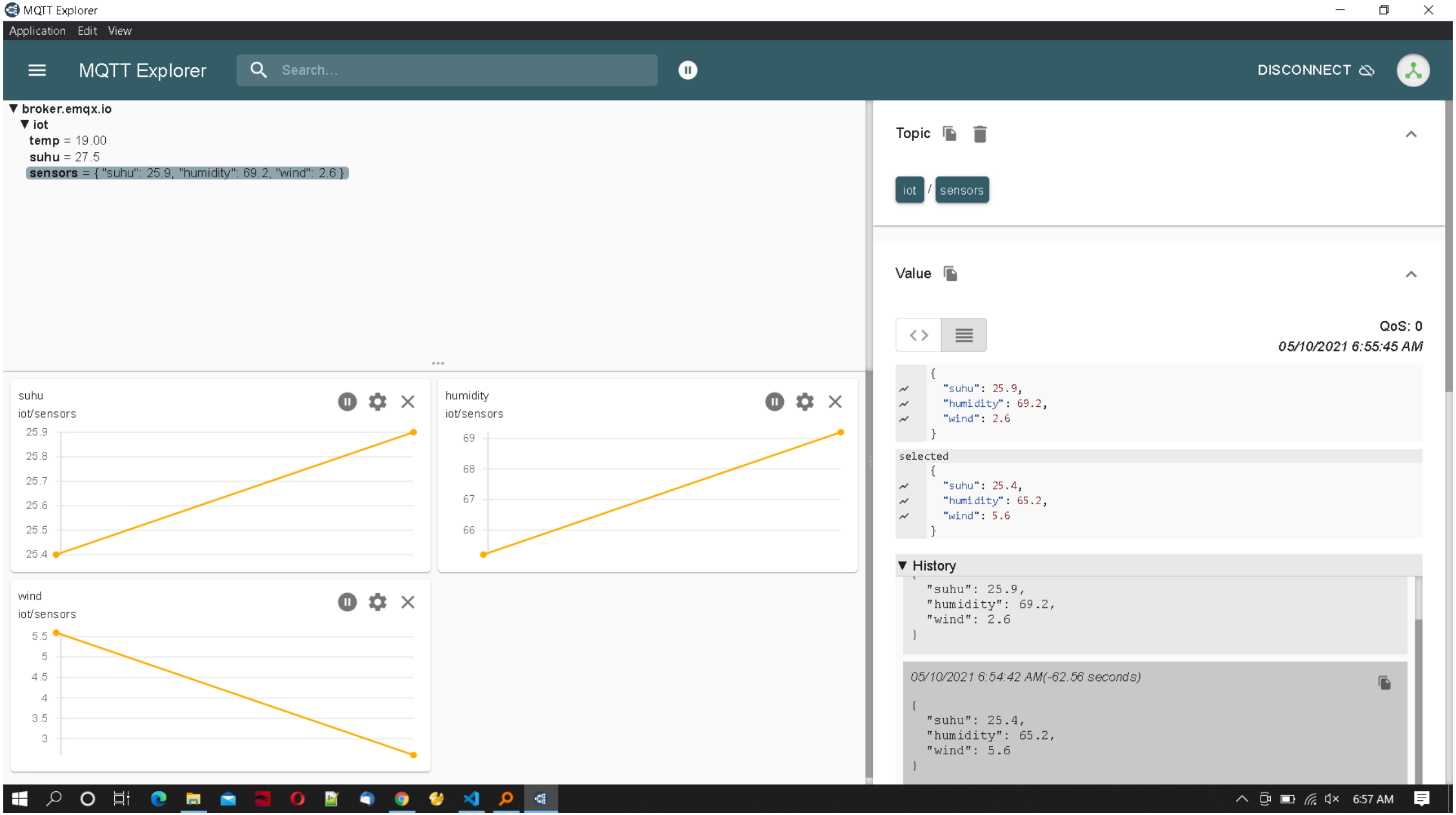



MQTT Explorer
Application Edit View
MQTT Explorer
Search...
DISCONNECT
broker.emqx.io
iot
temp = 19.00
Topic
1
iot
Value
History
Publish
Topic
iot/suhu
raw
xml
json
PUBLISH
18.76
QoS 0
retain
6:47 AM

3. Publish pesan ke topik **iot/sensors**



4. Lihat pada topik Subscribe dan plot sebagai berikut :



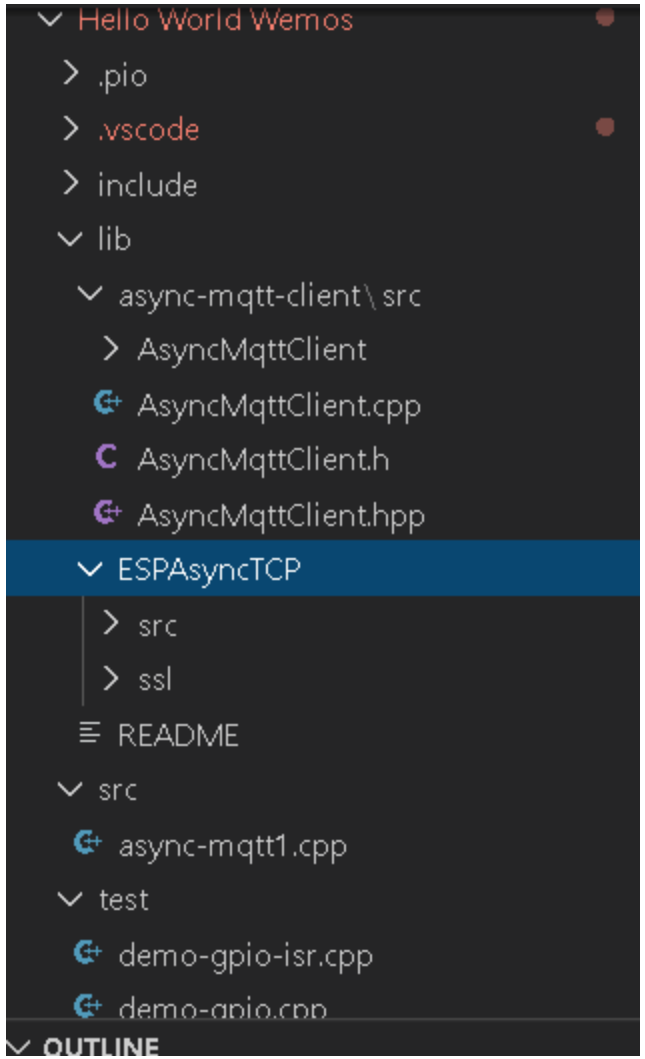
5. Publish data lainnya pada topik atau subtopik yang berbeda dan monitor pada layar aplikasinya.

Percobaan 2

- **Tujuan** :
  - Menginstall library ESP MQTT C;ient
  - Subscribe dan Publish via ESP32
- **Langkah** :

1. Download file librarynya dari :
   - https://github.com/knolleary/pubsubclient
   - https://github.com/plapointe6/EspMQTTClient

2. Ekstrak filenya dan kopi ke folder lib project PlatformIO. Buat juga file program untuk tes misal **esp-mqtt1.cpp** di folder **src**



3. Isi file tersebut dengan kode berikut ini dan amati apa yang terjadi di MQTT Explorer

```
#include <Arduino.h>
#include <Esp.h>
#include <EspMQTTClient.h>

#define DEVICE_ID "mqtt_ex4_esp32_13072023"
EspMQTTClient client("iotx", "iot12345678",
                     "broker.emqx.io", // MQTT Broker ip atau hostname
                     "",               // Username MQTT Broker. Bisa dikosongkan
                     "",               // Password MQTT Broker. Bisa dikosongkan
                     DEVICE_ID,        // ID device. Harus unik!!!
                     1883              // Port MQTT Broker. Umumnya 1883
                     );

// variabel global
unsigned long tmr0 = 0;

// prototipe void
void onConnectionEstablished();
```

```
void setup() {
  Serial.begin(115200);

  // Fungsionalitas MQTT client library.
  client.enableDebuggingMessages(); // Pesan debugging ke serial port
  client.enableHTTPWebUpdater();    // Web update. Username password sama dengan
                                    // MQTT di atas
  // enableHTTPWebUpdater("user", "password").  // Custom username password
  client.enableOTA(); // Onkan OTA (Over The Air) update. Password sama dengan
                      // MQTT
  // Override dengan
  //  enableOTA("password", port).

  // pesan ketika device ini terputus
  client.enableLastWillMessage(
      DEVICE_ID "/lastwill", "I am going offline",
      true); // You can activate the retain flag by setting the
             // third parameter to true

  tmr0 = millis();
}
```

```
// FUNGSI INI HARUS DIIMPLEMENTASIKAN SECARA TERPISAH. HANYA SEKALI!!!
void onConnectionEstablished() {
  // Subscribe ke topik dan keluarkan ke serial port
  client.subscribe(DEVICE_ID "/command",
                   [](const String &payload) { Serial.println(payload); });

  // Subscribe ke topik sembarang di bawah sebuah subtopik
  client.subscribe(DEVICE_ID "/general/#", [](const String &topic,
                                              const String &payload) {
    Serial.println("(From wildcard) topic: " + topic + ", payload: " + payload);
  });

  // Publish pesan ke sebuah topik
  client.publish(DEVICE_ID "/status", DEVICE_ID " ON");

  // // Execute delayed instructions
  // client.executeDelayed(5 * 1000, []() {
  //   client.publish("mqtt_ex4_esp32/wildcardtest/test123",
  //                  "This is a message sent 5 seconds later");
  // });
}
```

```
void loop() {
  client.loop();

  if (millis() - tmr0 >= 5000) {
    tmr0 = millis();
    if (client.isMqttConnected()) {
      Serial.print(F("Publish..."));
      //   Serial.println(client.publish("mqtt_ex4_esp32/status",
      //                                 String(millis()) + String(":") +
      //                                     String(ESP.getFreeHeap())));
      Serial.println(
          client.publish(DEVICE_ID "/ram", String(ESP.getFreeHeap())));
    }
  }

  yield();
}
```

Referensi

- https://www.emqx.io/mqtt/public-mqtt5-broker
- http://mqtt-explorer.com/
- https://github.com/plapointe6/EspMQTTClient
- https://github.com/knolleary/pubsubclient
- https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod
- https://github.com/srevinsaju/guiscrcpy


THANK YOU VERY MUCH