

## Topik 10

# Pengenalan dan Praktikum Open Source MQTT Broker Untuk Pendukung Open-Source IoT Platform

Digitalent Scholarship Professional Academy

**Isi dan elemen dari dokumen ini memiliki hak kekayaan intelektual yang dilindungi oleh undang-undang**

**Dilarang menggunakan, merubah, memperbanyak, dan mendistribusikan dokumen ini untuk tujuan komersil**

# Outline

- Pengenalan MQTT
- Apa yang Harus dipersiapkan
- Mosquitto
  - Pengenalan Mosquitto
  - Protokol Mosquitto
- Wokwi
  - Pengenalan Wokwi
  - Bagian pada Wokwi
- MQTT Explorer
- Mengirimkan Data dari Publisher ke MQTT Broker
- Subscribe Data yang Dikirimkan Publisher
- Mengirimkan Data Dalam Bentuk String dan Json

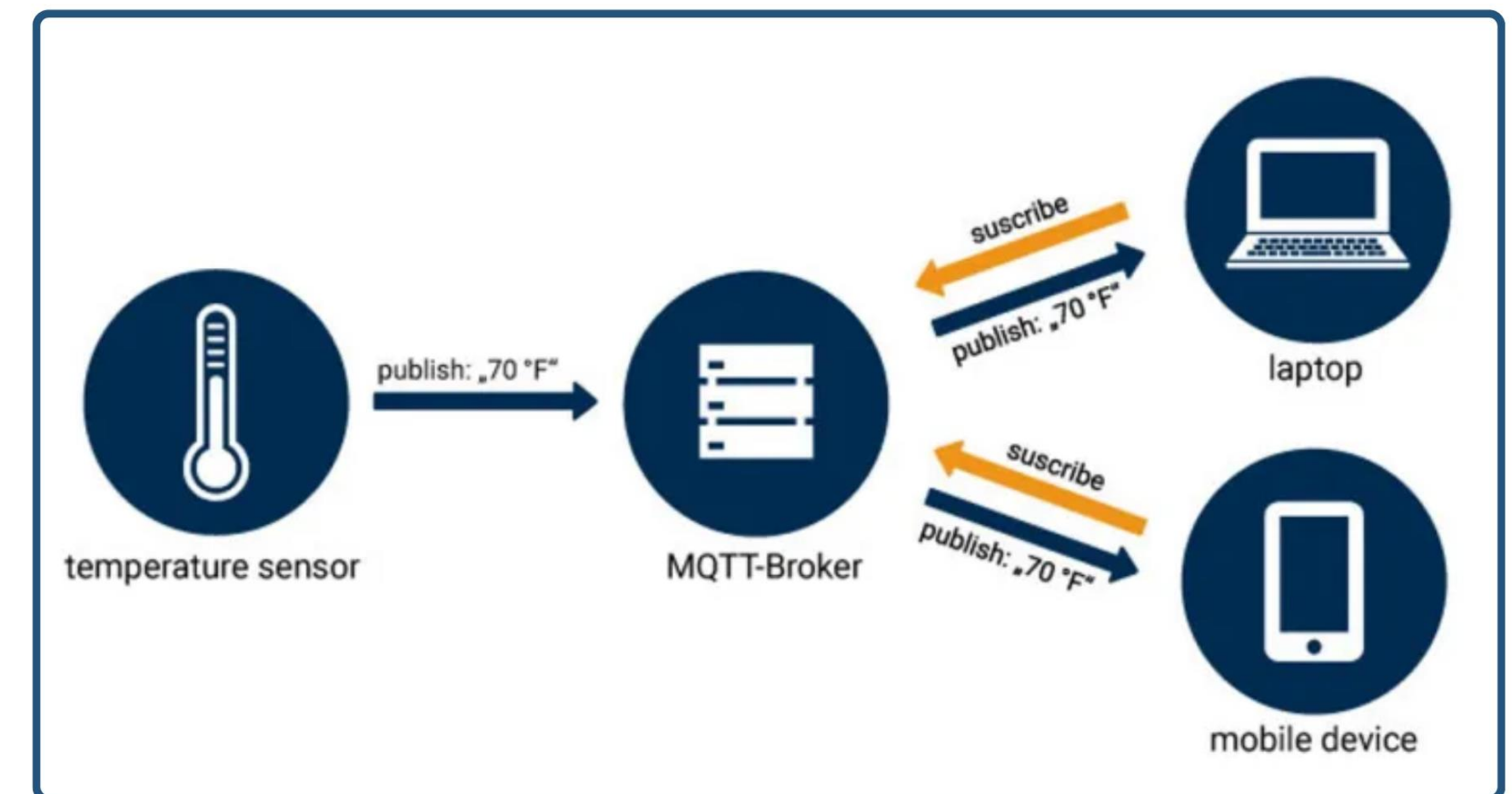


# A. Pengenalan MQTT

MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi pengiriman pesan yang dibentuk dengan TCP/IP berdasarkan model messaging publish-subscribe dan dirancang khusus untuk machine to machine yang tidak memiliki alamat khusus.

Publisher berfungsi sebagai penyedia data, yang dimana datanya nanti akan dikirimkan ke MQTT broker. Di bagian MQTT broker ini akan disimpan secara sementara. Kemudian subscriber akan mengambil data tersebut dari MQTT broker.

Nantinya subscriber bisa mengolah data yang sudah di subscribe atau data yang sudah diambil bisa ditampilkan pada sebuah dashboard pada platform tertentu atau disimpan dalam database.



*Ilustrasi Cara Kerja MQTT*



# B. Persiapan Praktikum

Berikut ini merupakan alat dan bahan yang digunakan dalam praktikum :

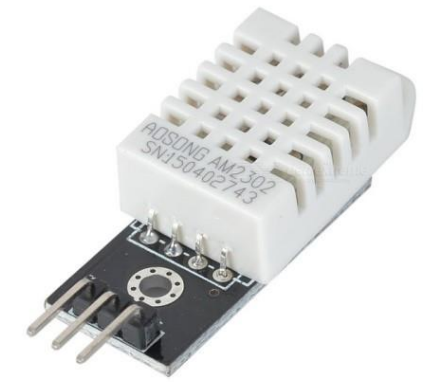
- **ESP32**  
Dalam praktikum kali ini ESP32 akan digunakan sebagai publisher dan subscriber. Artinya dalam praktikum kali ini membutuhkan 2 ESP32
  - **Sensor**  
Sensor yang digunakan adalah jenis sensor suhu DHT22.
  - **Kabel Jumper**  
Untuk menghubungkan ESP 32 dengan Sensor
- Bagi yang belum memiliki hardwarenya bisa menggunakan simulator yaitu wokwi. Pada simulator wokwi sudah tersedia ESP32 beserta dengan sensornya.

Untuk softwarenya yang akan digunakan adalah MQTT Explorer.

MQTT brokernya kita akan menggunakan mosquitto, yang nanti akan dimasukan *test.mosquitto.org* ke dalam MQTT Explorer serta di dalam pemrograman ESP32



ESP32



DHT22



Kabel Jumper

# C. Mosquitto

Eclipse Mosquitto™ adalah Message broker yang mengimplementasikan protokol MQTT versi 3.1 dan 3.1.1. Broker Mosquitto cukup handal terutama dalam sistem internet of things karena sudah digunakan oleh banyak user.

Mosquitto telah mendukung berbagai sistem operasi mulai dari Mac OS, Microsoft Windows, dan berbagai varian distro Linux.

Website: <https://mosquitto.org>



# C. Mosquitto

Mosquitto menggunakan protokol MQTT, untuk port berikut ini :

1883 : MQTT, unencrypted, unauthenticated

1884 : MQTT, unencrypted, authenticated

8883 : MQTT, encrypted, unauthenticated

8884 : MQTT, encrypted, client certificate required

8885 : MQTT, encrypted, authenticated

8886 : MQTT, encrypted, unauthenticated

8887 : MQTT, encrypted, server certificate deliberately expired

8080 : MQTT over WebSockets, unencrypted, unauthenticated

8081 : MQTT over WebSockets, encrypted, unauthenticated

8090 : MQTT over WebSockets, unencrypted, authenticated

8091 : MQTT over WebSockets, encrypted, authenticated

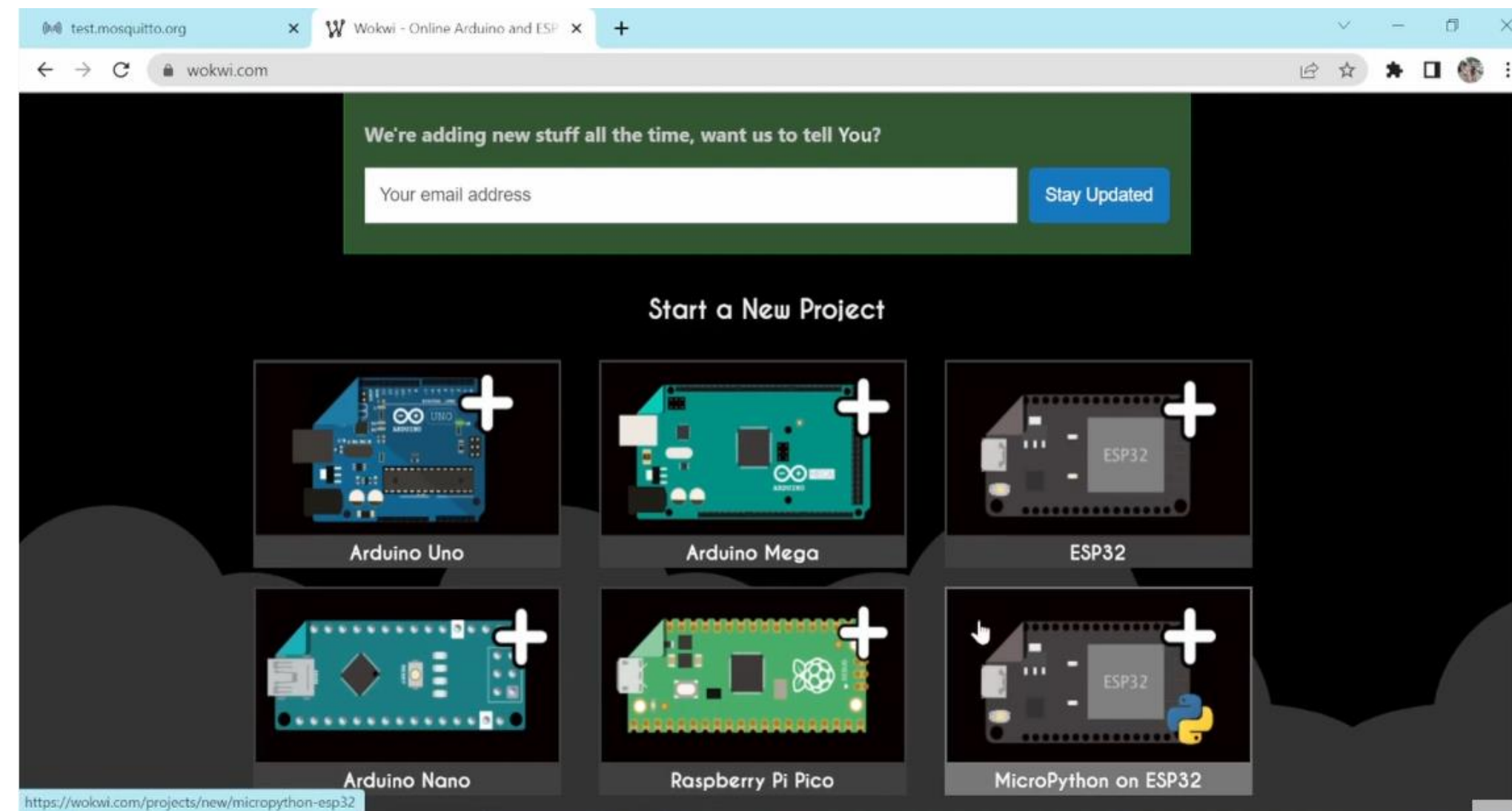
Tetapi port yang biasa digunakan MQTT adalah port 1883





# D. Wokwi

Wokwi adalah platform yang dapat digunakan sebagai simulator electronic development board. Di dalam wokwi ini sudah tersedia beberapa mikrokontroller diantaranya Arduino Uno, Arduin Mega, ESP32, Arduino Nano, Raspberry Pi Pico, dan MicroPhyton ESP32.

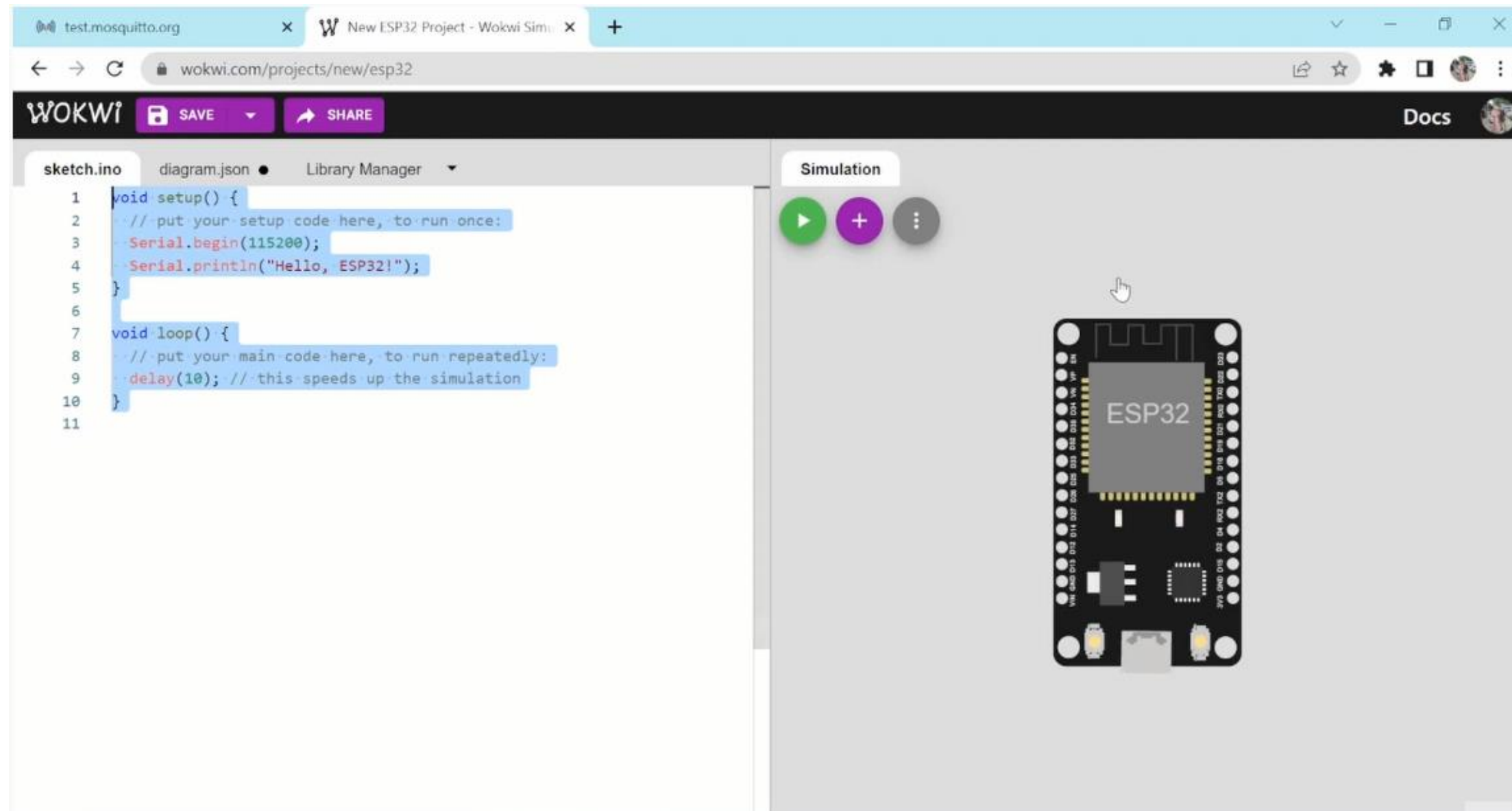


Tampilan Awal Wokwi



# D. Wokwi

Wokwi terdiri dari 2 bagian.



Bagian kiri ini merupakan tempat untuk melakukan pemrograman dan atau sketch

Di samping kanan ada hardware, disini kita bisa menambahkan beberapa sensor.

# E. Menginstall MQTT Explorer

- Download dan Install terlebih dahulu software MQTT Explorer
- Download melalui <http://mqtt-explorer.com/>
- Kemudian pilih sesuai sistem operasi Laptop/Komputer








- Delete topics recursively
- Diff view of current and previous received messages
- Publish topics
- Plot numeric topics
- Retain a history of each topic
- Dark/Light Themes
- ... [See Changelog to see all features](#)


The hierarchical view makes this tool so easy to use and differentiates the **MQTT Explorer** from other great MQTT clients like [MQTTLens](#), [MQTTBox](#) and [MQTT.fx](#).  
This MQTT Client strives to be a MQTT swiss-army-knife, the perfect tool to integrate new services and IoT devices on your network.

### // Download

Developing this tool takes a lot of effort, sweat and time, please consider rating the App on the Windows or Mac app store ★★★★★.

If you feel like a feature is missing or you found a bug, please leave me a [comment](#) / [issue](#) and I'll see what I can do.

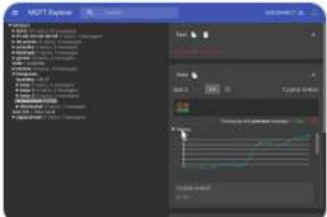
Platform		Downloads
	Windows	 <a href="#">portable, installer</a>
	Mac	 <a href="#">dmg</a>
	Ubuntu <i>debian, mint, neon, fedora, etc...</i>	 <a href="#">Get it from the Snap Store</a> <code>snap install mqtt-explorer</code> <a href="#">Ubuntu Store</a>
	Linux <i>almost every linux</i>	<a href="#">Applimage</a> <i>Run Applimage: Make it executable and double-click it.</i>

**Ubuntu**  
@ubuntu · Follow  

If you're seeking a comprehensive and easy to use **#MQTT** client for **#Ubuntu** then look no further than MQTT Explorer.

snap install mqtt-explorer

[bit.ly/2Y95g6P](https://bit.ly/2Y95g6P)



8:02 PM · May 7, 2019

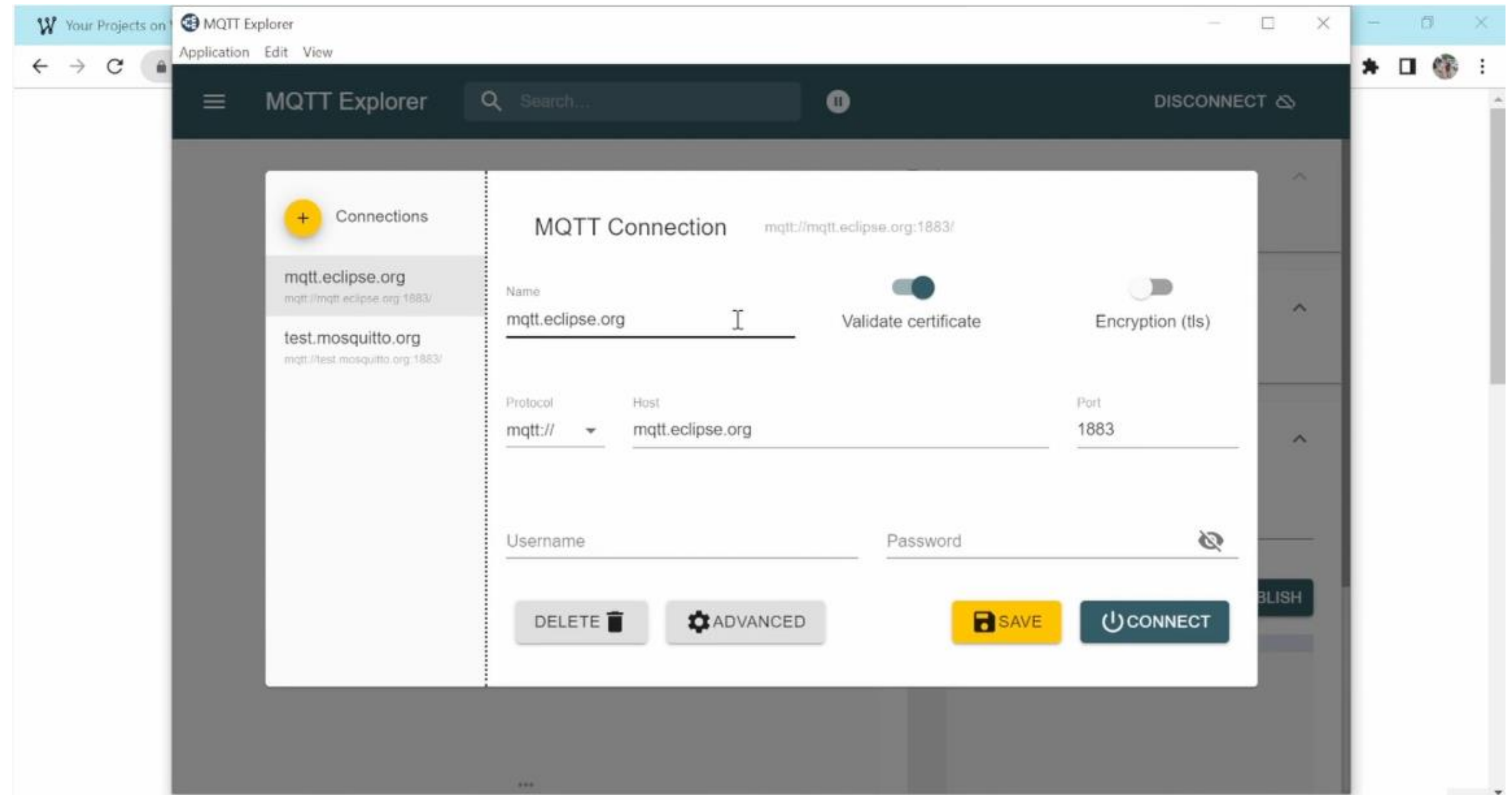
117 ❤️ Reply Co...

[Read 1 reply](#)

# E. Menginstall MQTT Explorer

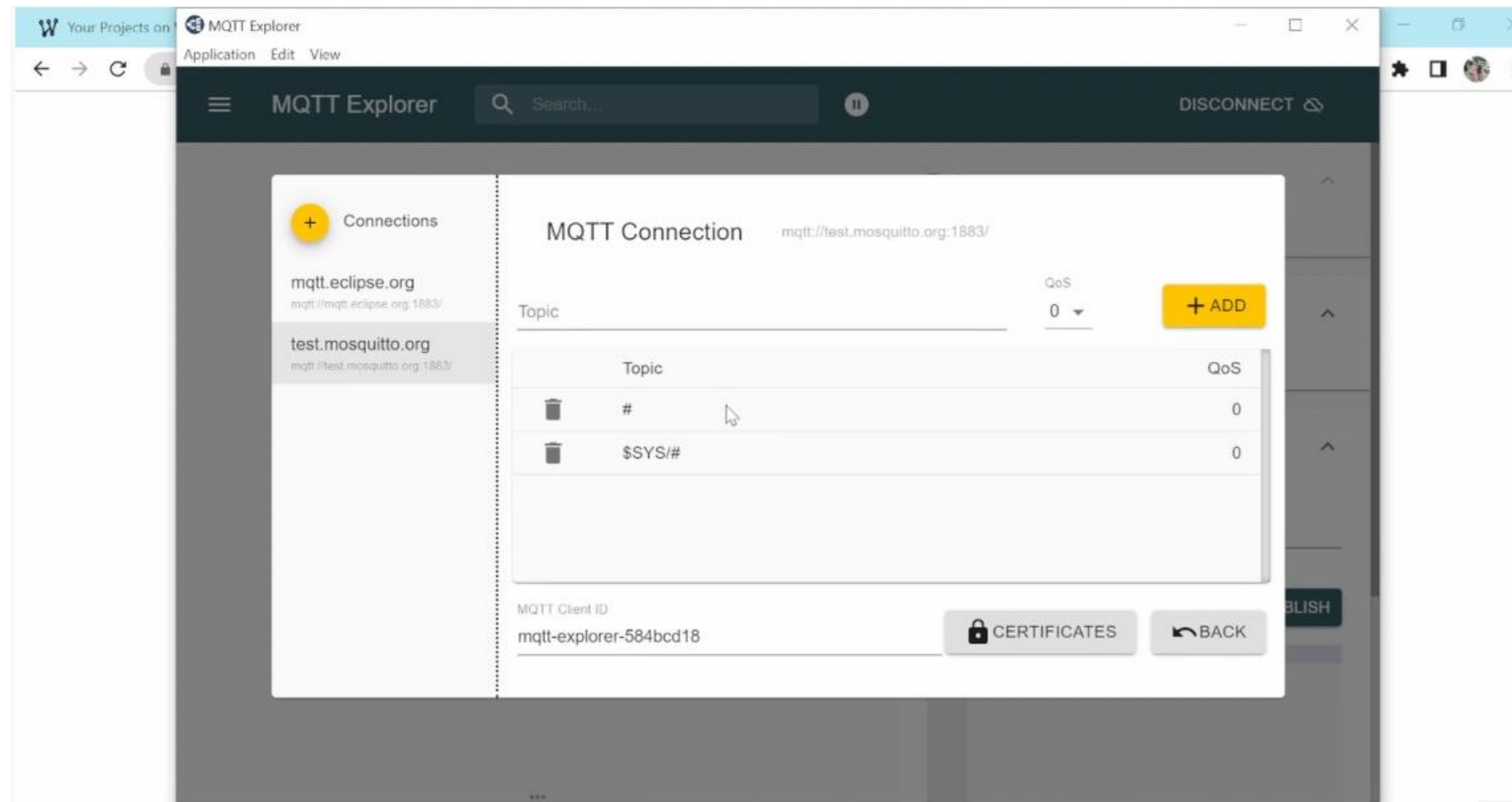
Tersedia 2 broker explorer setelah melakukan penginstalan

- mqtt.eclipse.org
- Test.mosquitto.org



# E. Menginstall MQTT Explorer

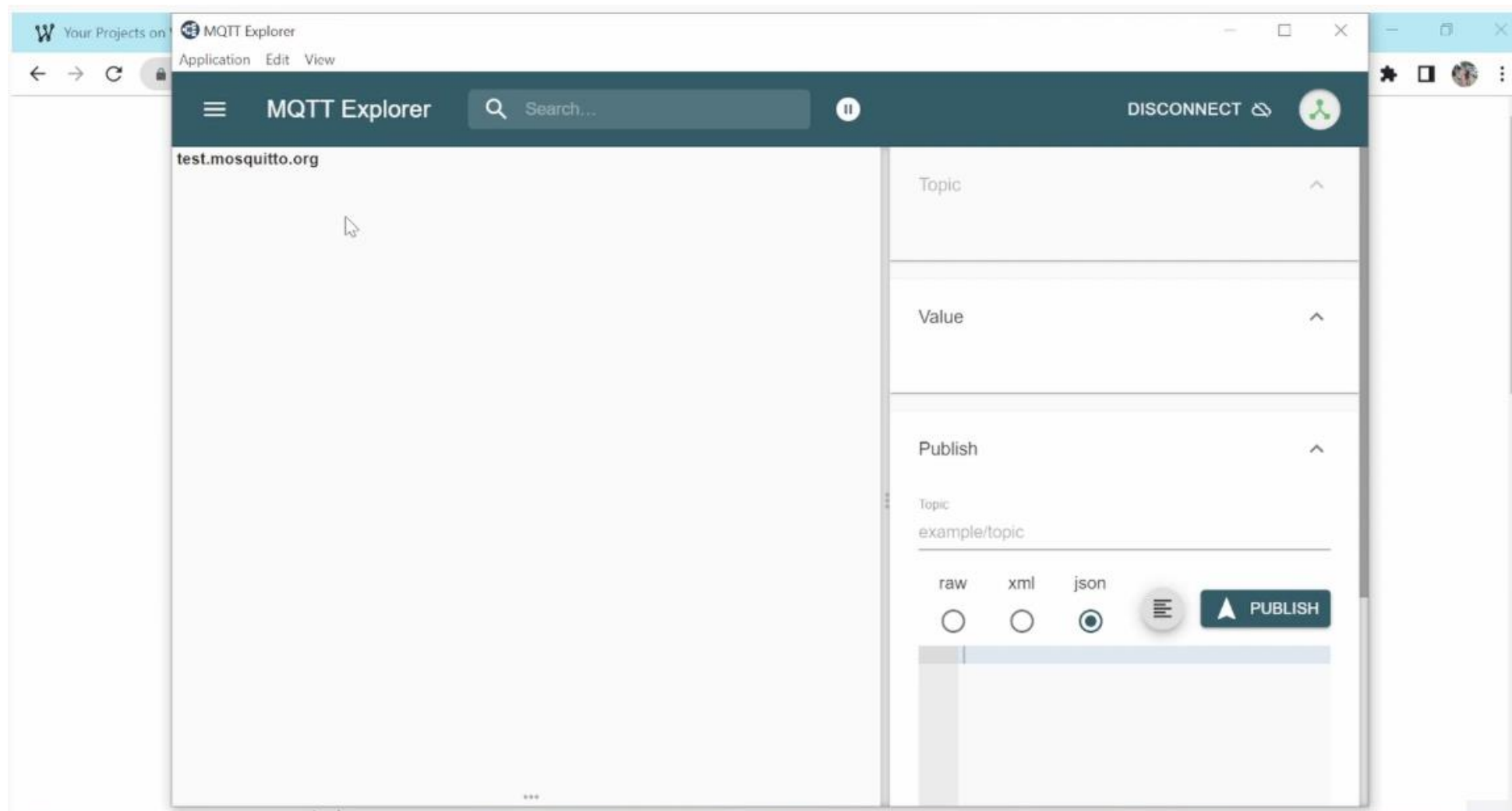
Membuat alamat atau topic yang dibutuhkan oleh publisher atau subsribernya



- Masuk ke bagian advance
- Hapus 2 topik yang telah tersedia. Hal ini bertujuan agar MQTT menampilkan data yang kirimkan
- Setelah itu buat topik baru
- Karena sensor DHT22 terdiri dari 2 data yaitu temperatur dan humidity, buat topik dengan nama "User1/Temp" dan yang satunya "User1/Hum"
- Topik sudah bisa digunakan oleh ESP32 sebagai publisher untuk mengirimkan datanya ke topik ini, dan juga subscriber untuk mendapatkan data yang dikirim oleh publisher dengan mengakses topik yang sudah disiapkan.



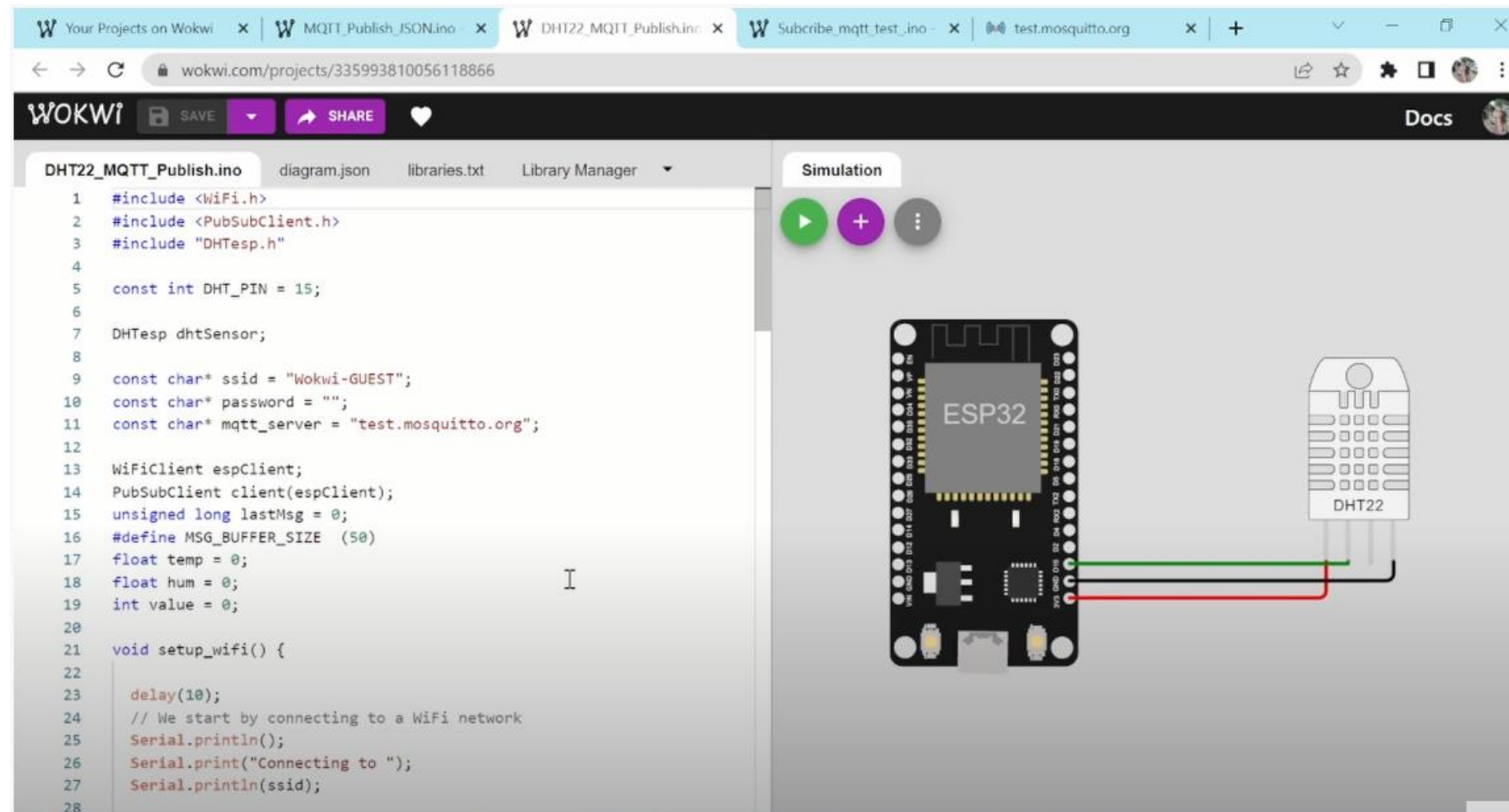
# E. Menginstall MQTT Explorer



- Setelah itu klik back lalu connect
- Selanjutnya akan ada pada tahap mengirimkan data dari ESP32 sebagai publisher dan subscribe dari ESP32 sebagai subscribenya.

# F. Mengirimkan Data Dari Publisher ke MQTT Broker

Membuat alamat atau topic yang dibutuhkan oleh publisher atau subsribernya



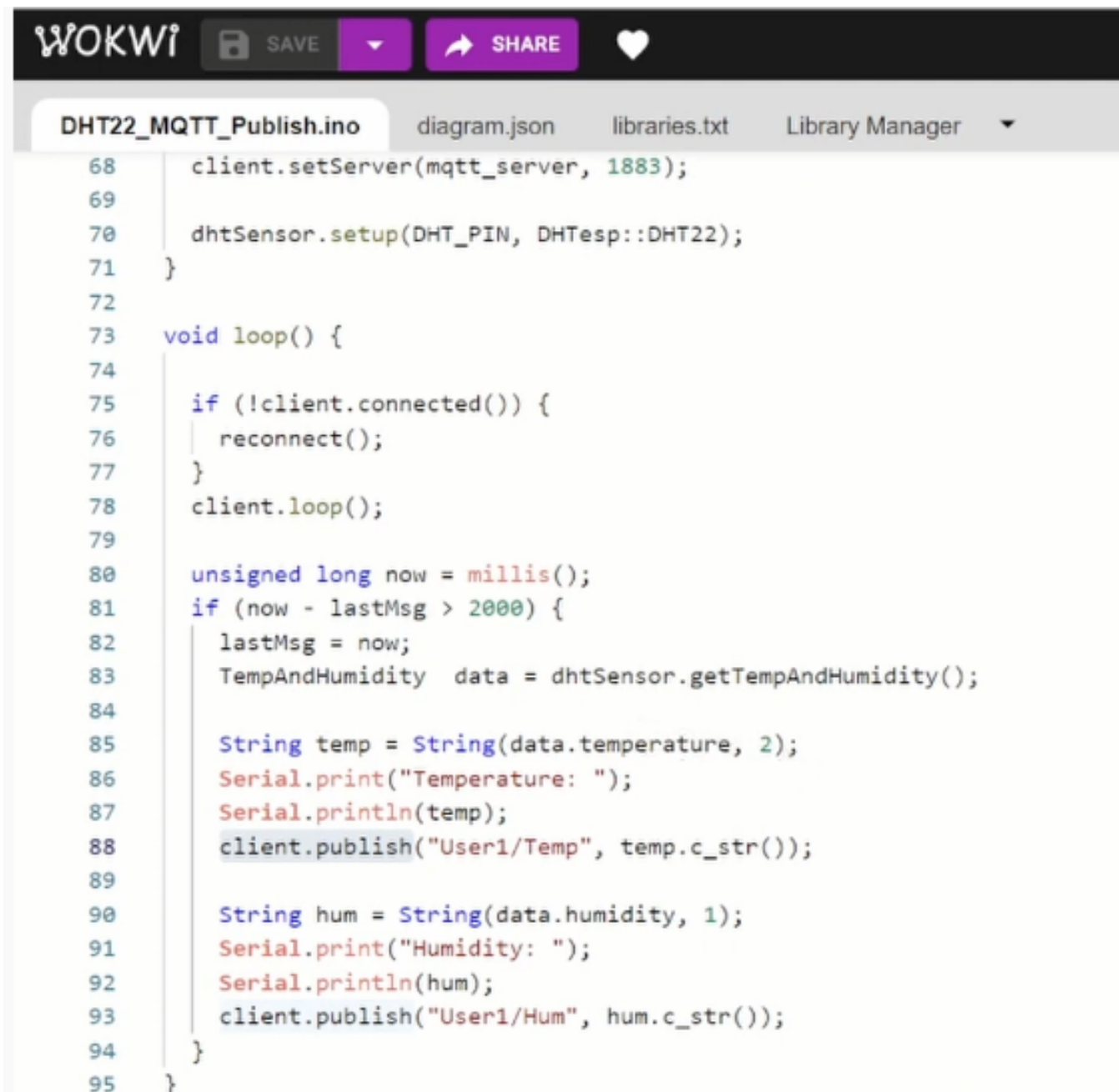
- Disini kita siapkan 3 pemrograman
- 2 untuk pemrograman sebagai publisher. Yang pertama publisher dengan mengirimkan data dalam bentuk string dan kedua yaitu publisher yang mengirimkan datanya dalam bentuk string dan juga json.
- Yang ke 3 yaitu pemrograman sebagai subscriber yang akan mengambil data yang telah dikirimkan oleh publisher ke MQTT broker
- Kemudian Buatlah rangkaian seperti gambar disamping

Coding MQTT Publish Strings : [Download](#)

Coding MQTT Publish Json : [Download](#)

Coding MQTT Subscribe : [Download](#)

# F. Mengirimkan Data Dari Publisher ke MQTT Broker



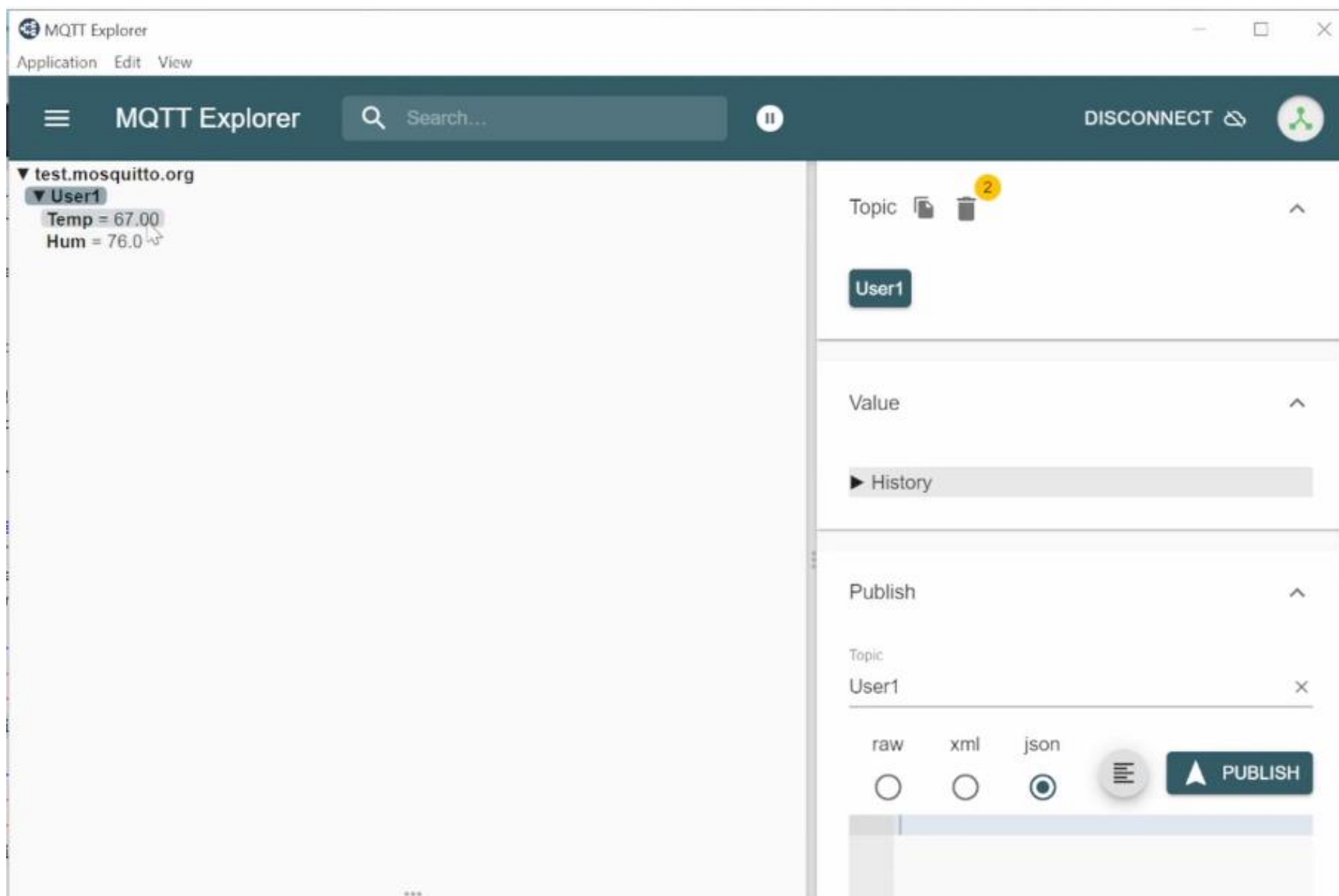
```
WOKWI SAVE SHARE
DHT22_MQTT_Publish.ino diagram.json libraries.txt Library Manager
68 client.setServer(mqtt_server, 1883);
69
70 dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
71 }
72
73 void loop() {
74
75     if (!client.connected()) {
76         reconnect();
77     }
78     client.loop();
79
80     unsigned long now = millis();
81     if (now - lastMsg > 2000) {
82         lastMsg = now;
83         TempAndHumidity data = dhtSensor.getTempAndHumidity();
84
85         String temp = String(data.temperature, 2);
86         Serial.print("Temperature: ");
87         Serial.println(temp);
88         client.publish("User1/Temp", temp.c_str());
89
90         String hum = String(data.humidity, 1);
91         Serial.print("Humidity: ");
92         Serial.println(hum);
93         client.publish("User1/Hum", hum.c_str());
94     }
95 }
```

- Buat rangkaian seperti pada gambar disamping
- Sesuaikan bagaian "client.publish" dengan nama topik yang sudah dibuat tadi
- Kemudian jalankan projectnya, lalu cek apakah datanya sudah terkirim ke MQTT broker

Coding MQTT Publish Strings : [Download](#)

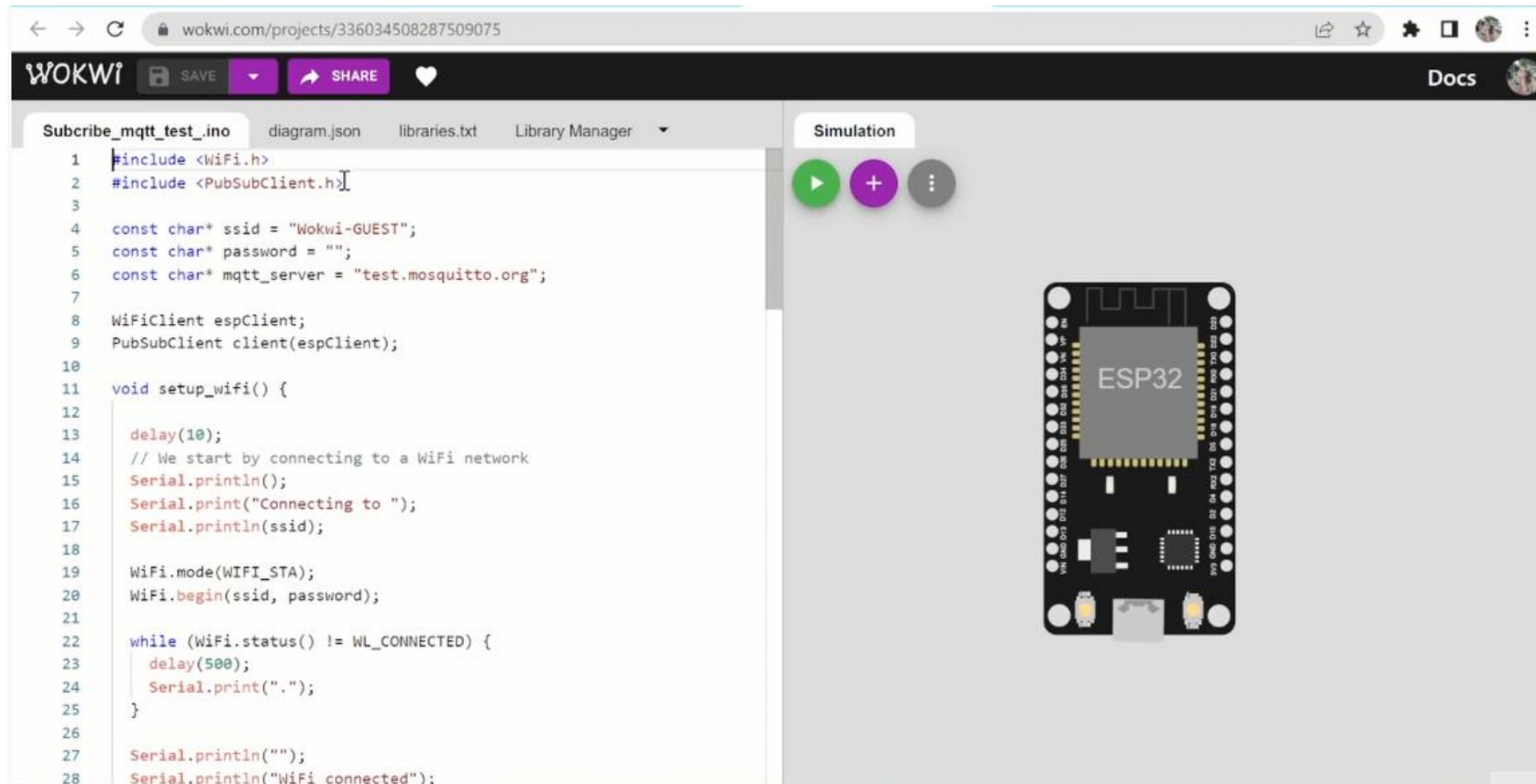
# F. Mengirimkan Data Dari Publisher ke MQTT Broker

- Data temperature dan Kelembapan sudah terkirim ke MQTT broker.





# G. Subscribe data yang Dikirimkan Oleh Publisher

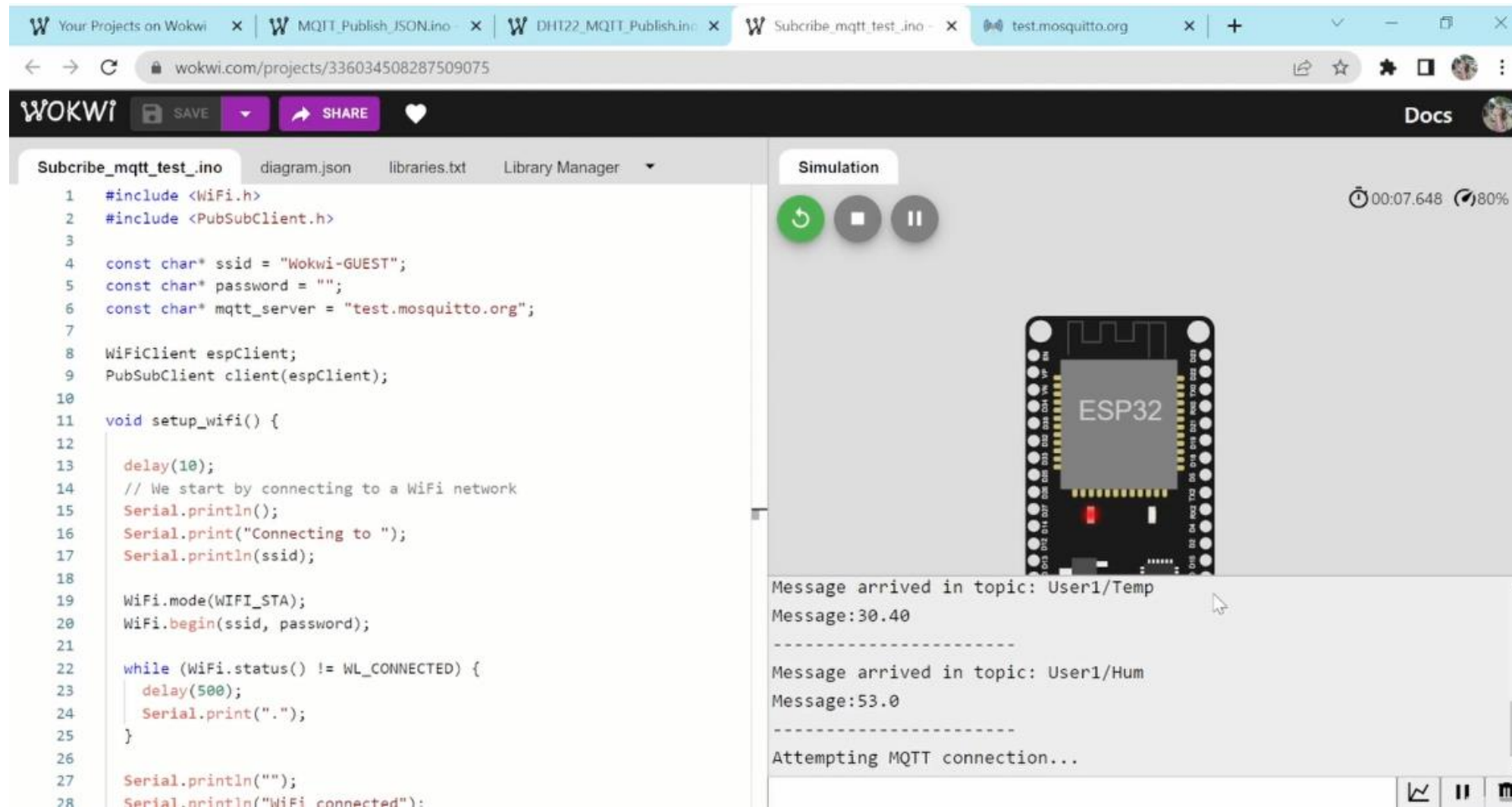


The screenshot shows the Wokwi IDE interface. On the left, a code editor displays a C++ sketch named 'Subscribe\_mqtt\_test\_ino'. The code includes libraries for WiFi and PubSubClient, defines constants for SSID, password, and MQTT server, and implements a setup function for WiFi and a loop for connecting to the network. On the right, a simulation window shows a 3D model of an ESP32 microcontroller board. The browser address bar indicates the project URL: wokwi.com/projects/336034508287509075.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 const char* ssid = "Wokwi-GUEST";
5 const char* password = "";
6 const char* mqtt_server = "test.mosquitto.org";
7
8 WiFiClient espClient;
9 PubSubClient client(espClient);
10
11 void setup_wifi() {
12     delay(10);
13     // We start by connecting to a WiFi network
14     Serial.println();
15     Serial.print("Connecting to ");
16     Serial.println(ssid);
17
18     WiFi.mode(WIFI_STA);
19     WiFi.begin(ssid, password);
20
21     while (WiFi.status() != WL_CONNECTED) {
22         delay(500);
23         Serial.print(".");
24     }
25
26     Serial.println("");
27     Serial.println("WiFi connected");
28 }
```

- Perbedaan sketch program dari sebelumnya adalah bagian "void callback". Void callback ini akan mensubscribe data dari MQTT broker
- Masukkan kode programnya
- Kemudian jalankan bagian dari subscriernya untuk mensubscribe
- Jalankan juga pemrograman dari publishernya

# G. Subscribe data yang Dikirimkan Oleh Publisher

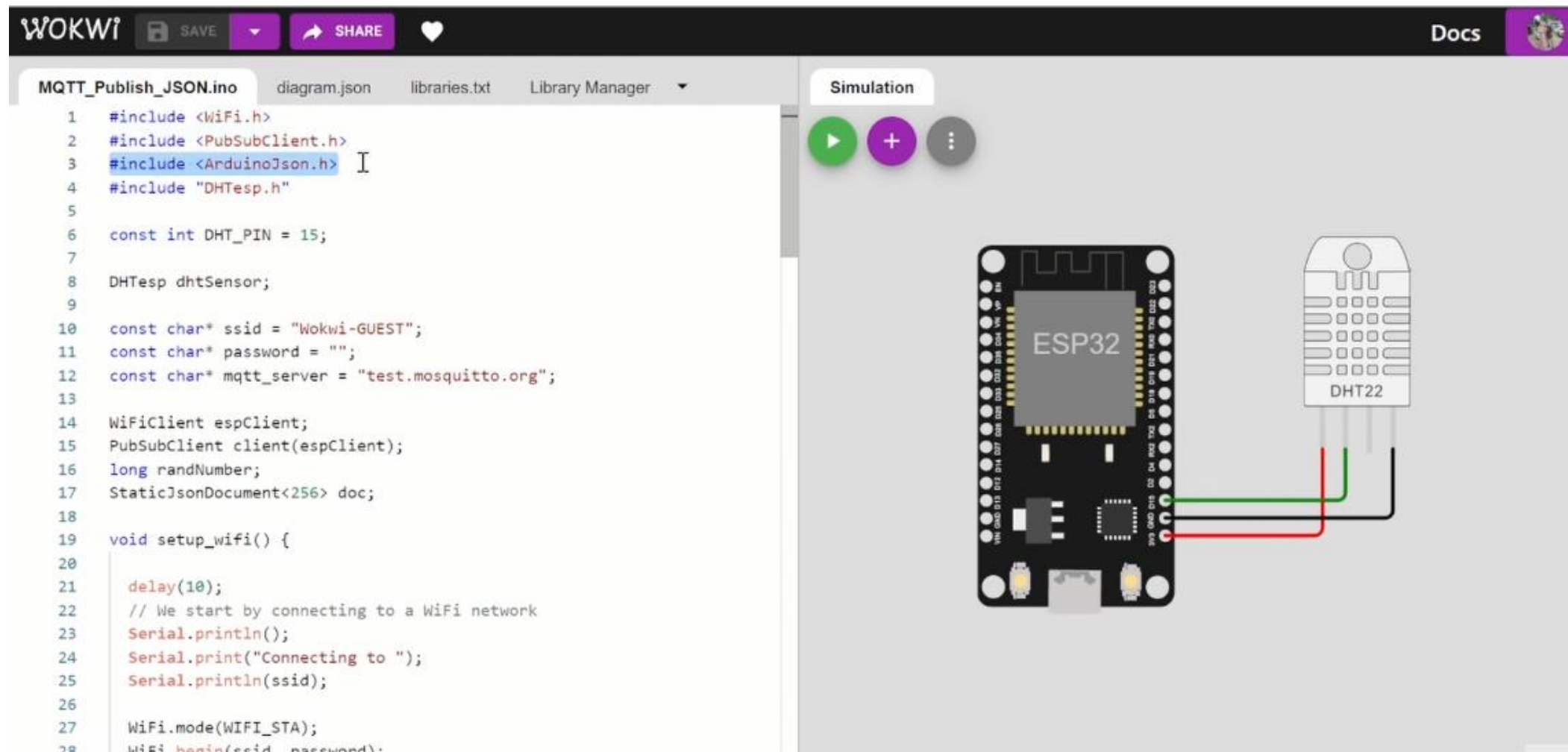


- Data yang dikirimkan oleh ESP32 publisher ke MQTT broker akan diakses oleh ESP32 Subscribe.
- Jadi data yang diterima oleh ESP32 Subscriber adalah data yang dikirim ESP32 Publisher melalui MQTT Broker

Coding MQTT Subscribe : [Download](#)

# H. Mengirimkan data dalam bentuk json dan string

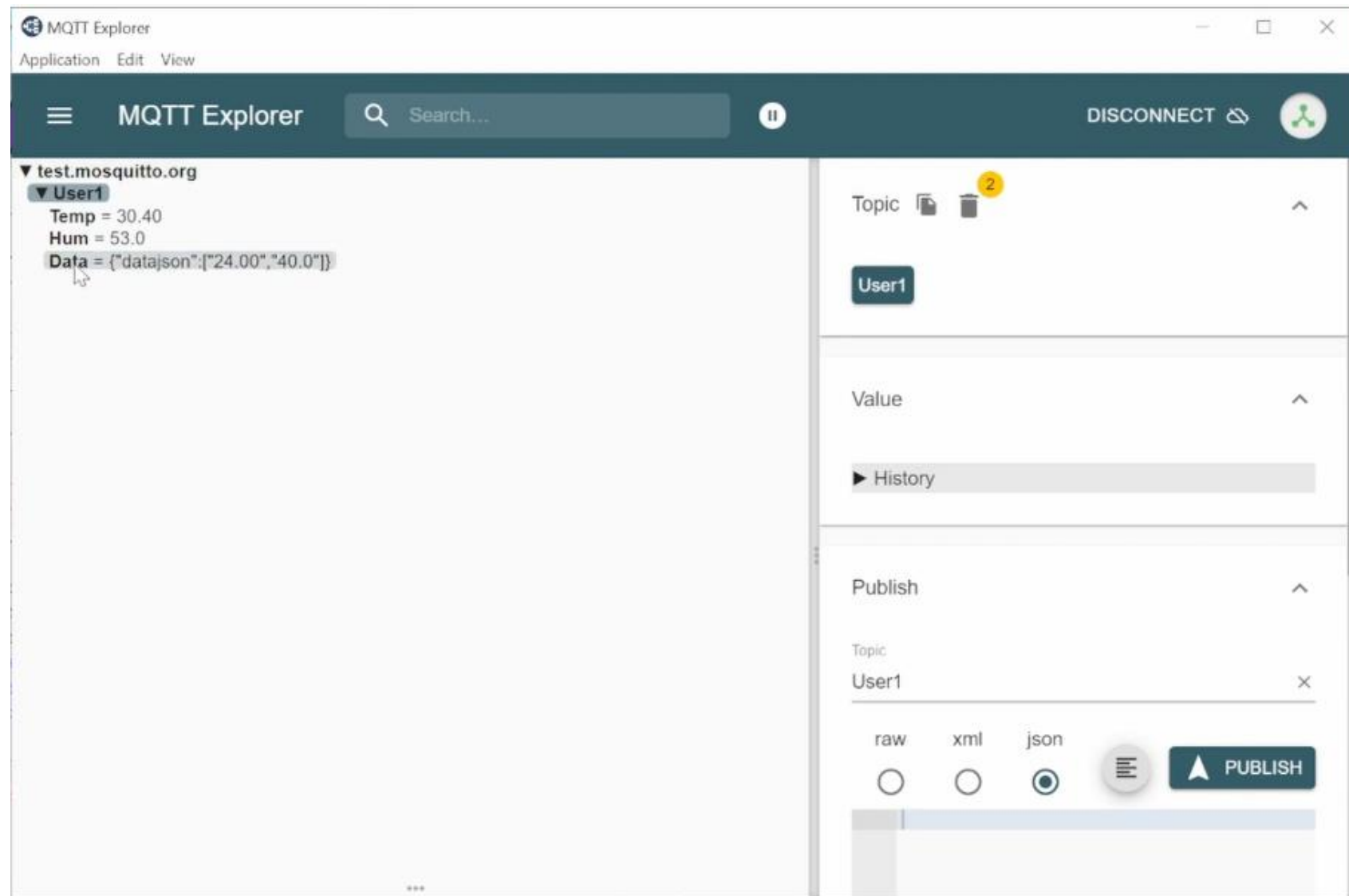
Praktikum mengirimkan data dalam bentuk json dan string secara bersamaan. Perbedaannya jika dalam bentuk json kita akan mengirimkan data temperature dan humiditynya dalam 1 paket yang sama



- Buatlah rangkaian seperti gambar disamping
- Masukan kode programnya
- Jalankan terlebih dahulu ESP32 Publisher, kemudian jalankan juga ESP32 Subscriber.
- Jika semua sudah terkoneksi, jalankan ESP32 json ini.

Coding MQTT Publish Json : [Download](#)

# H. Mengirimkan data dalam bentuk json dan string

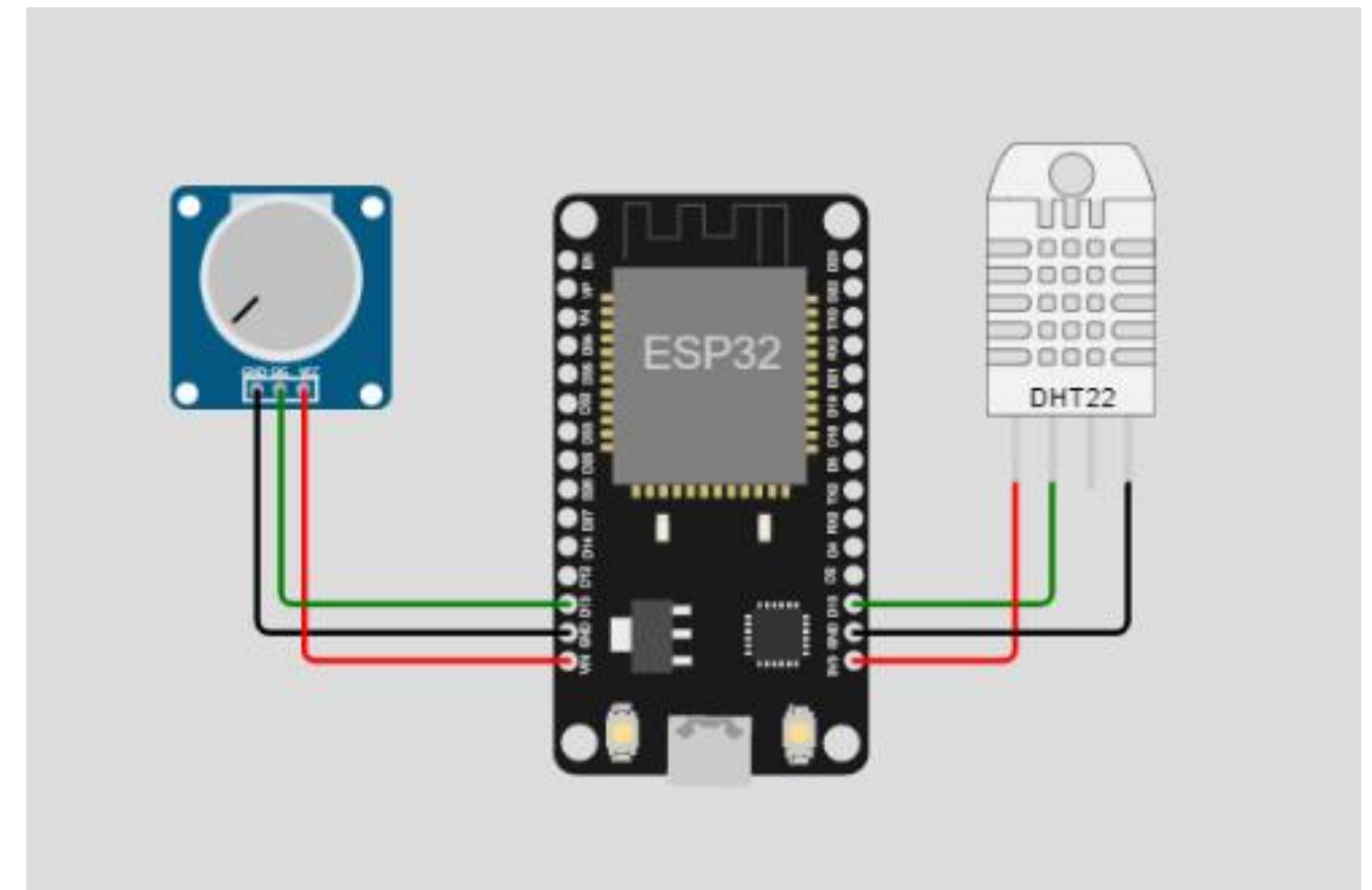


- "Temp" dan "Hum" merupakan data yang dikirimkan ESP32 Publisher
- Sedangkan "Data" adalah data yang dikirimkan oleh ESP32 Json
- Pada ESP32 Subscriber juga akan menampilkan data seperti pada MQTT explorer yaitu data dari ESP32 Publisher dan ESP32 Json



# Challenge Mengirimkan Data dalam bentuk JSON & String dari Publisher ke MQTT Broker

Kirimkan data dalam bentuk JSON &STRING dari publisher ke MQTT broker dengan memanfaatkan ESP32, potensio serta DHT22



**Sekian Materi**

# **Pengenalan dan Praktikum Open Source MQTT Broker Untuk Pendukung Open-Source IoT Platform**

**Digitalent Scholarship Professional Academy**