

Modul Pelatihan  
Inkind *Matching Fund* Kedaireka  
Parametrik - ITS

## MQTT for IoT (*Internet of Things*)



By:

**parametrik**

PT. Parametrik Solusi Integrasi  
Bandung – Jawa Barat  
2021

## A. Pendahuluan

Merupakan pelatihan *MQTT for IoT* yang dilakukan untuk memenuhi kontribusi *Inkind* PT. Parametrik Solusi Integrasi sebagai pihak industri ke Departemen Teknik Elektro Otomasi (DTEO) sebagai insan dikti pada Matching Fund – Kedaireka – Kampus Merdeka 2021. Pelatihan ini akan dilakukan kurang lebih 1 – 2 jam dengan fokus untuk mempelajari dasar konsep protokol komunikasi MQTT ketika diimplementasikan pada *Internet of Things* (IoT). Harapan setelah pelatihan dilakukan, peserta dapat memahami secara penuh penggunaan protokol komunikasi MQTT untuk IoT dan menerapkannya secara praktik (hands – on).

## B. Hardware & Software

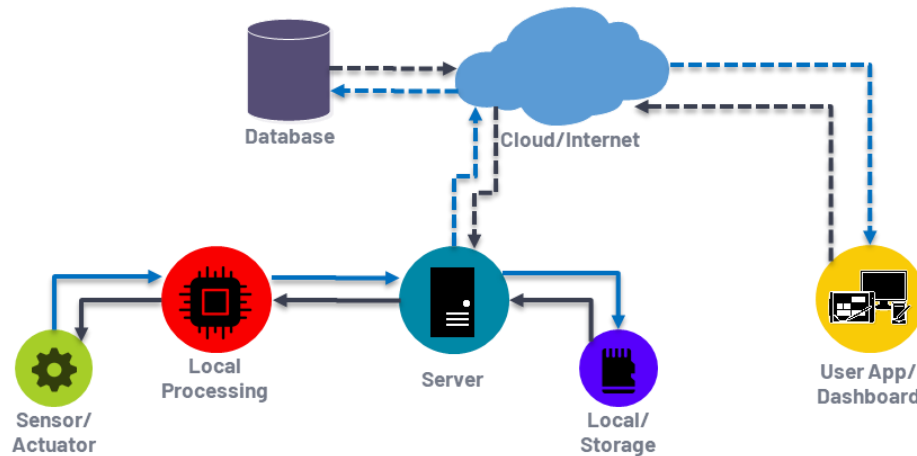
1. Computer/laptop
2. Micro USB to USB type A
3. Embedded System (ESP8266/ESP32)
  - ESP8266 : <https://tokopedia.link/dZkgPIZOzlb>
  - ESP32 : <https://tokopedia.link/X2f7PjaPzlb>
4. Arduino IDE
5. Download MQTT Broker Mosquitto
  - Mosquitto Broker : <https://mosquitto.org/download/>
6. Download MQTT Client
  - MQTTfx : <https://www.jensd.de/wordpress/?p=2746>
  - MQTTx : <https://mqttx.app/>
7. Akses Internet

## C. Dasar Teori

1. Overview *Internet of Things* (IoT)

Secara terminology *Internet of Things* (IoT) terdiri dari dua kata yakni “*Internet*” dan “*Things*”. “*Internet*” merupakan interkoneksi jaringan computer, berdasarkan standar protokol TCP/IP, sedangkan “*Things*” adalah semua objek unik (*unique object*) yang ada di dunia. Sehingga, dapat disimpulkan bahwa *IoT* adalah objek dengan alamat unik yang terkoneksi, berdasarkan standar protokol komunikasi. Selanjutnya, beberapa pengertian IoT menurut beberapa perusahaan IT besar di dunia dan *dictionary* yakni sebagai berikut:

- “*The IoT is a giant network of connected things and people – all of which collect and share data about the way they are used and about the environment around them*” - **International Business Machines (IBM)**
- “*The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet*” – **ORACLE**
- “*The networking capability that allows information to be sent to and received from objects and devices (such as fixtures and kitchen appliances) using the Internet*” – **Merriam-Webster**



Gambar 1 Schema IoT

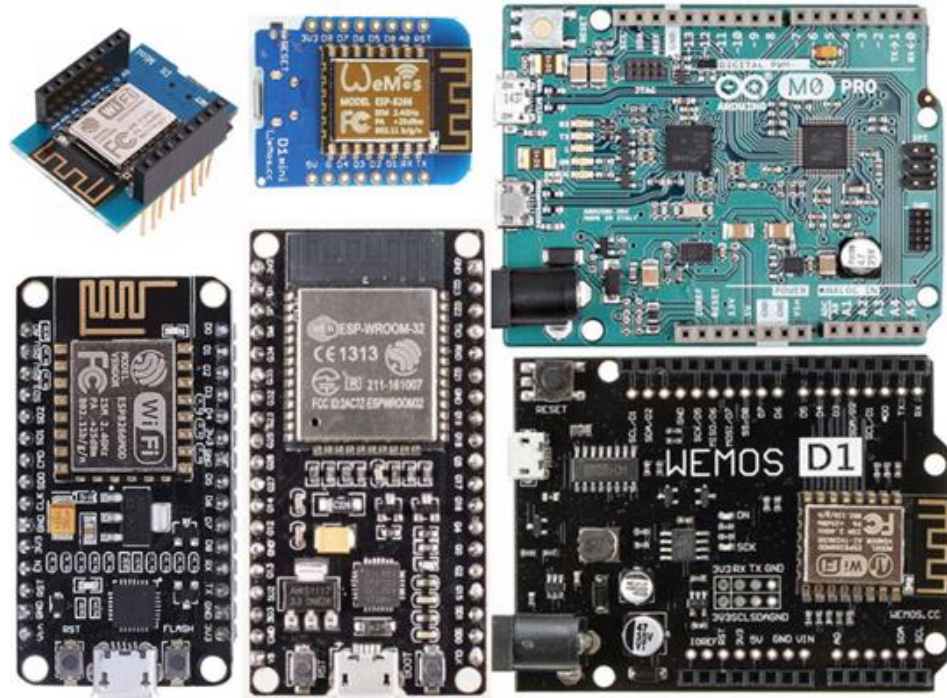
Pada Gambar 1 menunjukan skema atau arsitektur dasar IoT, dimana itu merupakan dasar referensi yang bisa diterapkan untuk mulai membangun IoT. Berdasarkan sudut pandang industri, IoT terdiri dari beberapa bidang keilmuan yakni:

- **Sensor/Actuator**  
 Sensor : Perangkat yang dapat mengubah besaran fisis menjadi besaran listrik  
 Aktuator : Perangkat mekanis yang menggerakkan sebuah objek sistem
- **Local Processing (MCU)**  
 Merupakan perangkat pada urutan ke – 2/3 pada IoT yang cukup penting untuk kalkulasi beberapa sensor untuk di teruskan ke *processing device* lebih *advanced* atau server. Pada level ini lebih banyak penggunaan hardware embedded system dan sejenisnya seperti: STM32, Arduino, Raspberry Pi, dll
- **Server + Storage**  
 Ini merupakan perangkat utama yang berbentuk hardware dengan tugas mengumpulkan, handle, kalkulasi, dan kontrol data dari semua perangkat peripheral (sensor dan actuator). Selain itu, biasanya pada server ini juga terdapat backup disk yang disediakan untuk historical log data.
- **Networking**  
 Merupakan jaringan komputer (telekomunikasi) yang memungkinkan antar komputer dengan komputer lain dapat saling bertukar informasi/data.
- **Information System**  
 Merupakan sebuah organisasi yang terintegrasi dengan perangkat lunak untuk mengumpulkan data/informasi, melakukan kalkulasi, dan mendistribusikan informasi untuk sampai kepada pengguna.

## 2. Embedded System (ESP)

Merupakan sistem tertanam perangkat keras dan juga perangkat lunak berbasis mikrokontroler atau mikroprosesor untuk menjalankan sebuah task spesifik tertentu. Instruksi pemrograman sistem tertanam, disebut sebagai firmware,

disimpan dalam memori *read-only* atau *flash-memory*, berjalan dengan sumber daya perangkat keras komputer yang terbatas.



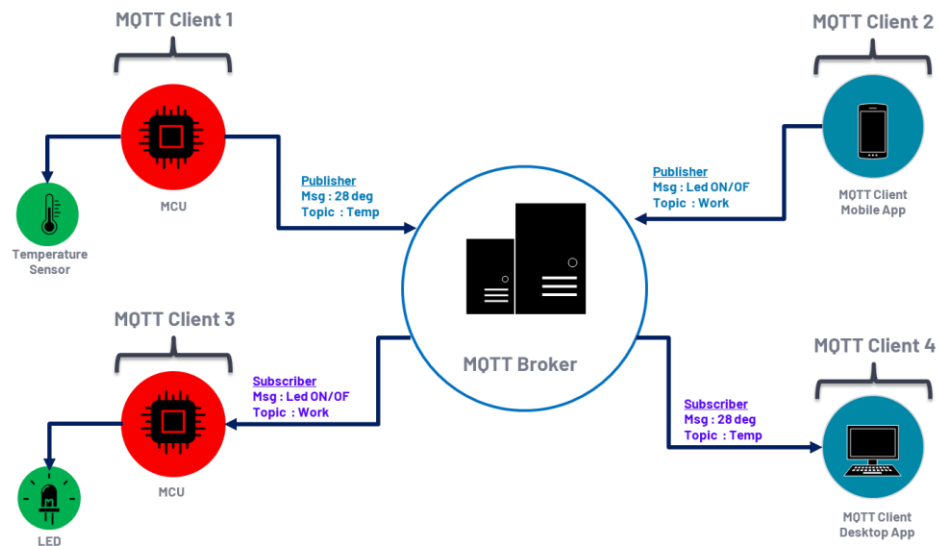
Gambar 2 Embedded System ESP

ESP sendiri adalah chip embbeded yang diproduksi oleh perusahaan Espressif dengan fungsi khusus untuk handling *AIoT system*. Contoh ESP Series umum yang sering kita jumpai yakni seperti ESP8266 & ESP32.

### 3. Protokol Komunikasi MQTT

Pada tahun 1999 ketika transformasi dari machine to machine(M2M). MQTT ditemukan oleh Dr Andy Stanford-Clark of IBM dan Arlen Nipper of Arcom (now Eurotech). MQTT sendiri adalah sebuah protocol konektifitas *machine to machine* (M2M) yang didesain mampu mengirimkan data dengan sangat ringan menggunakan arsitektur TCP/IP. MQTT bersifat *open-source*, sederhana, mudah untuk diimplementasikan, konsumsi daya yang rendah, dan mampu menangani ribuan client secara jarak jauh. Protokol MQTT juga mendukung *Quality of Service* (QoS) dengan tiga level yaitu QoS 0, QoS 1, dan QoS 2. Semakin besar levelnya, maka akan semakin ketat MQTT dalam aturan penerimaan paket dengan catatan akan mungkin terjadi peningkatan overhead.

Pada protokol MQTT terdapat dua komponen utama yaitu MQTT client dan MQTT Broker. MQTT client bertindak sebagai publisher/subscriber pesan dengan topic tertentu/spesifik. Sedangkan, MQTT Broker bertanggung jawab untuk handing proses publish/subscribe pesan yang dilakukan oleh MQTT Client, bisa diibaratkan sebagai server yang memiliki alamat khusus.



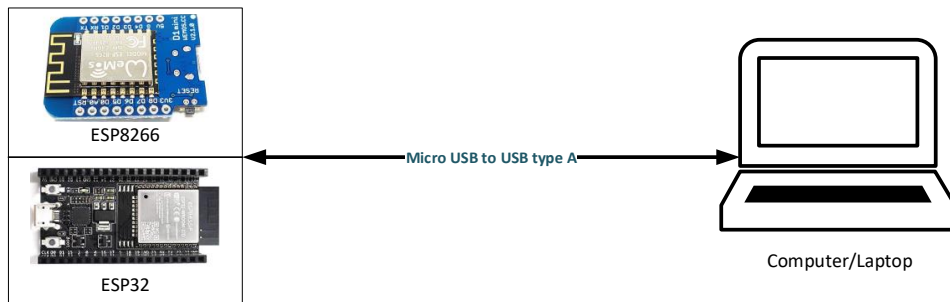
Gambar 3 Contoh Arsitektur MQTT

Gambar 3 menunjukkan arsitektur sederhana protokol komunikasi MQTT untuk proses publish dan subscribe (temperature, Led ON/OFF). Dengan beberapa keterangan detail yang dapat disimpulkan yakni:

- Client Publiher : MQTT Client 1, MQTT Client 2
- Client Publiher : MQTT Client 3, MQTT Client 4
- Paylod/Msg : Temperatur, Led ON/OFF
- Topic : Temp, Work

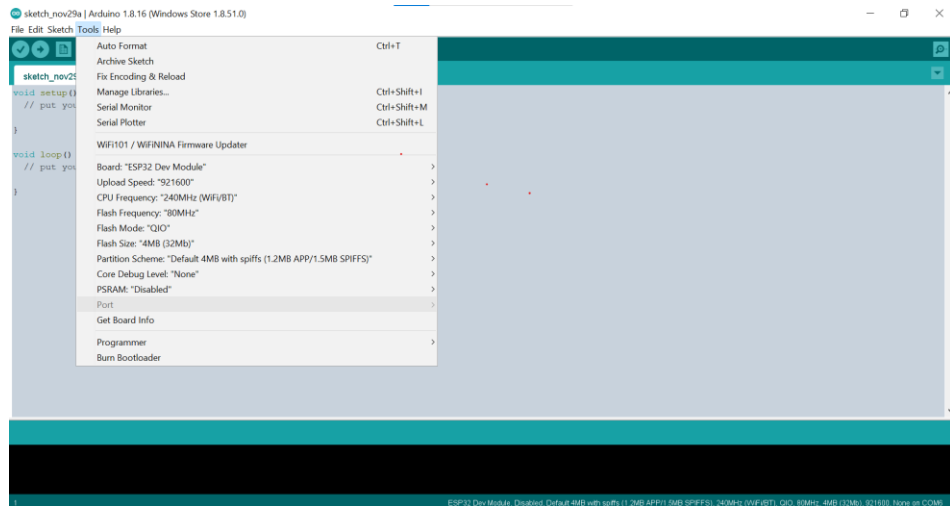
## D. Prosedur

1. Koneksikan hardware ESP8266/ESP32 ke Computer/Laptop anda



Gambar 4 Koneksi Hardware

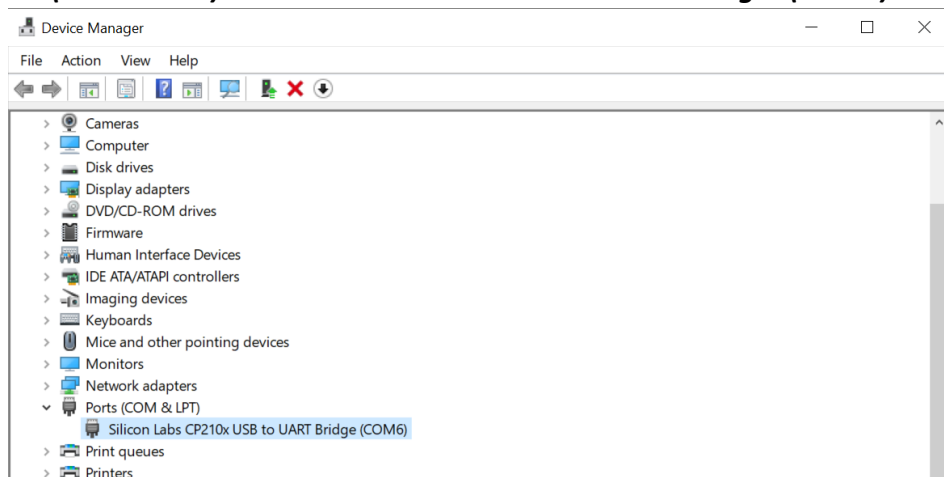
2. Buka Arduino IDE dan cek COM Port sudah terbaca atau belum - **Tools > Port**



Gambar 5 Tampilan COM Port Arduino IDE

- Ketika COM Port pada Computer/Laptop belum terbaca, download dan install driver CP210x\_USB.
- URL: <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

- Untuk memastikan bahwa driver telah berhasil terinstall - **Open Device Manager > Ports (COM & LPT) > Silicon Labs CP210x USB to UART Bridger (COMx) > terbaca**



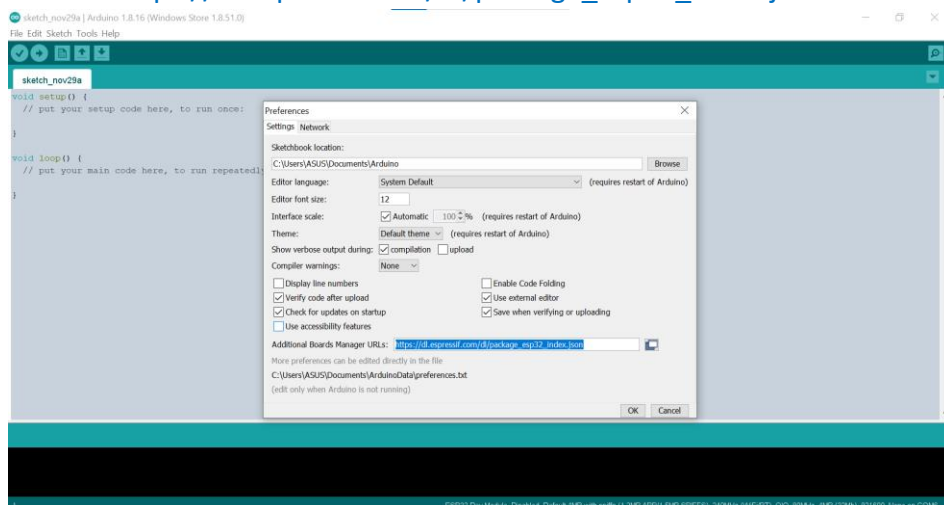
Gambar 6 Tampilan Ketika Driver CP210x\_USB Berhasil Terinstall

### 3. Tambah development board ESP8266/ESP32 pada Arduino IDE

- Tambahkan URL pada **Additional Board Manager – File > Preferences**

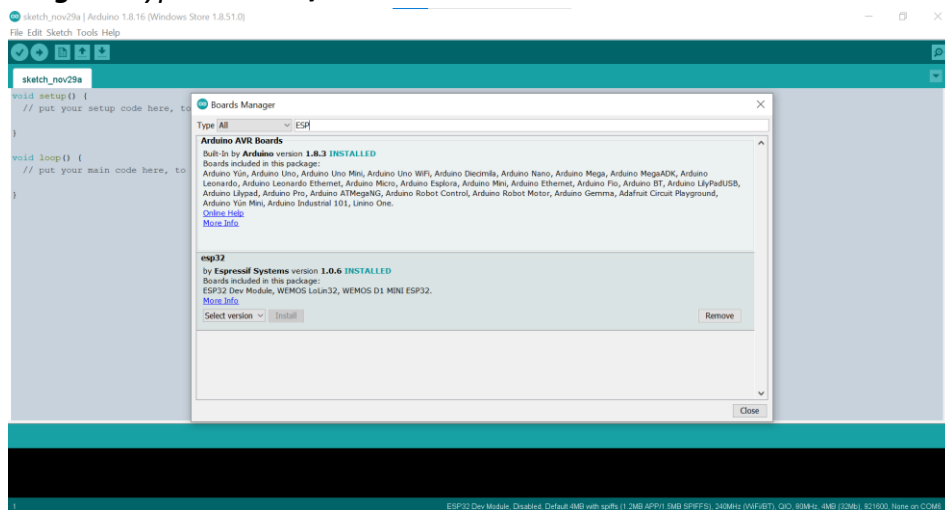
ESP8266 : [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

ESP32 : [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)



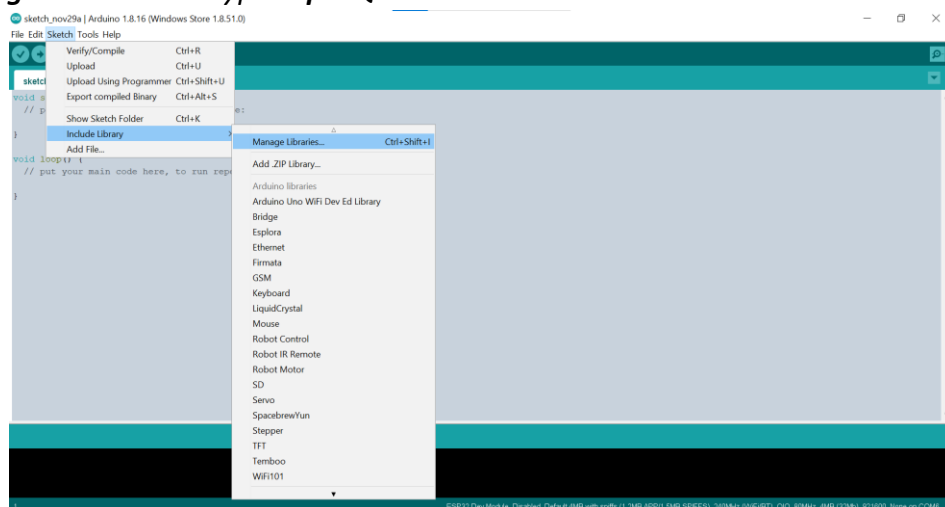
Gambar 7 Tampilan Arduino Preferences

- Tambahkan development board ESP8266/ESP32 – **Tools > Board > Board Manager > Type:ESP8266/ESP32 > Install**



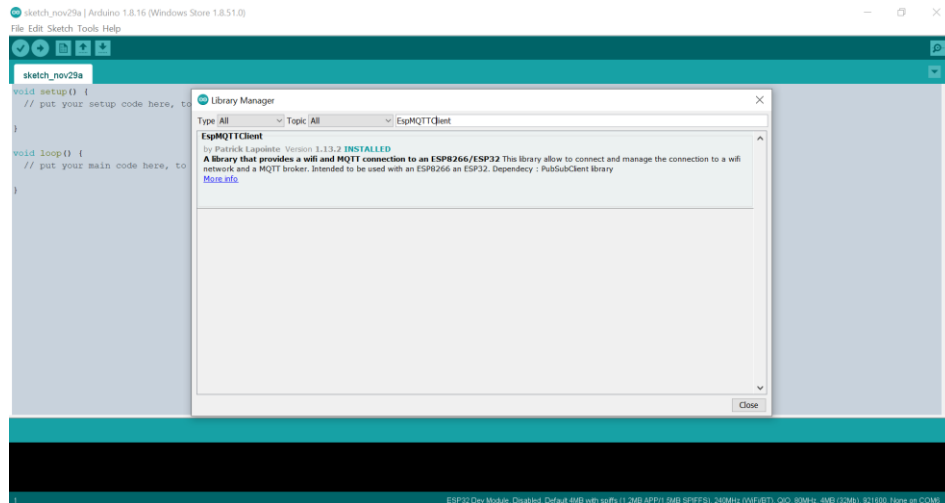
Gambar 8 Tampilan Board Manager ESP8266/ESP32

4. Tambahkan library “EspMQTTCClient” untuk Arduino – **Sketch > Include Library > Manage Libraries.. > Type:EspMQTTCClient**



Gambar 9 Tampilan Manage Libraries (1)

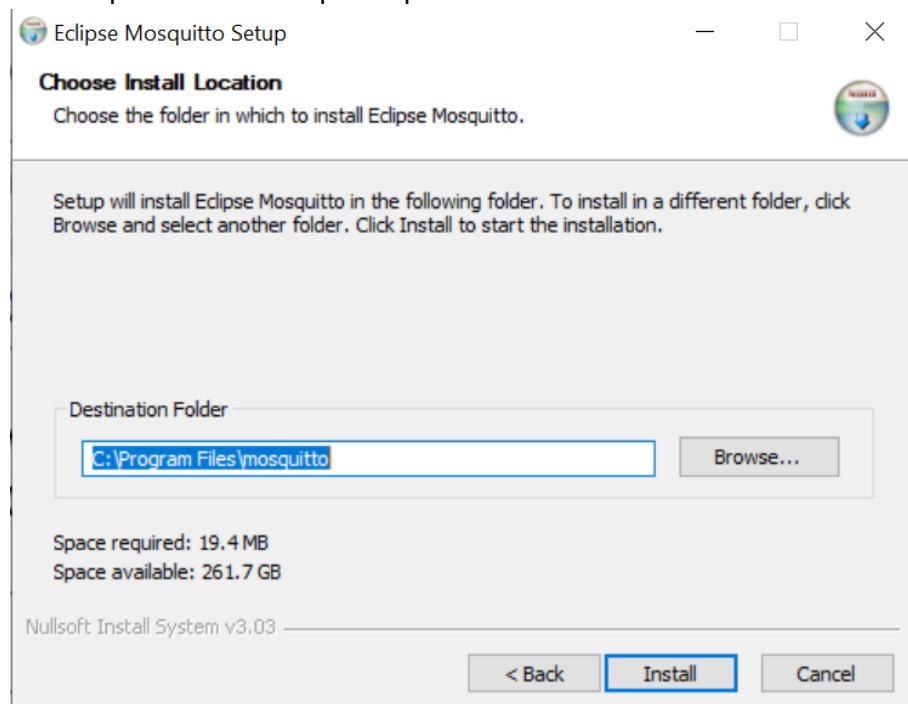




Gambar 10 Tampilan Manage Libraries (2)

##### 5. Download & install mosquitto broker

- URL: <https://mosquitto.org/download/>
- Pilih file sesuai spesifikasi computer/laptop yang digunakan  
ex: [mosquitto-2.0.14-install-windows-x64.exe](#) (64-bit build, Windows Vista and up, built with Visual Studio Community 2019)
- Install mosquitto as service pada spesifik folder



Gambar 11 Install Mosquitto Broker pada spesifik folder

##### 6. Jalankan mosquitto:

- *Open Command Prompt > Right Click > Run as administrator*
- *C:/> cd spesifik folder: (C:\Program Files\mosquitto) > type: mosquitto > enter*
- *C:\Program Files\Mosquitto> type: mosquitto -v > enter*

- *Another command:*
  - mosquitto* = running mosquitto broker
  - mosquitto -v (-verbose)* = see console messages
  - mosquitto -h* = mosquitto command help

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files\mosquitto
C:\Program Files\Mosquitto>mosquitto
C:\Program Files\Mosquitto>mosquitto -v
1638188388: mosquitto version 1.6.8 starting
1638188388: Using default config.
1638188388: Opening ipv6 listen socket on port 1883.
1638188388: Error: Only one usage of each socket address (protocol/network address/port) is normally permitted.

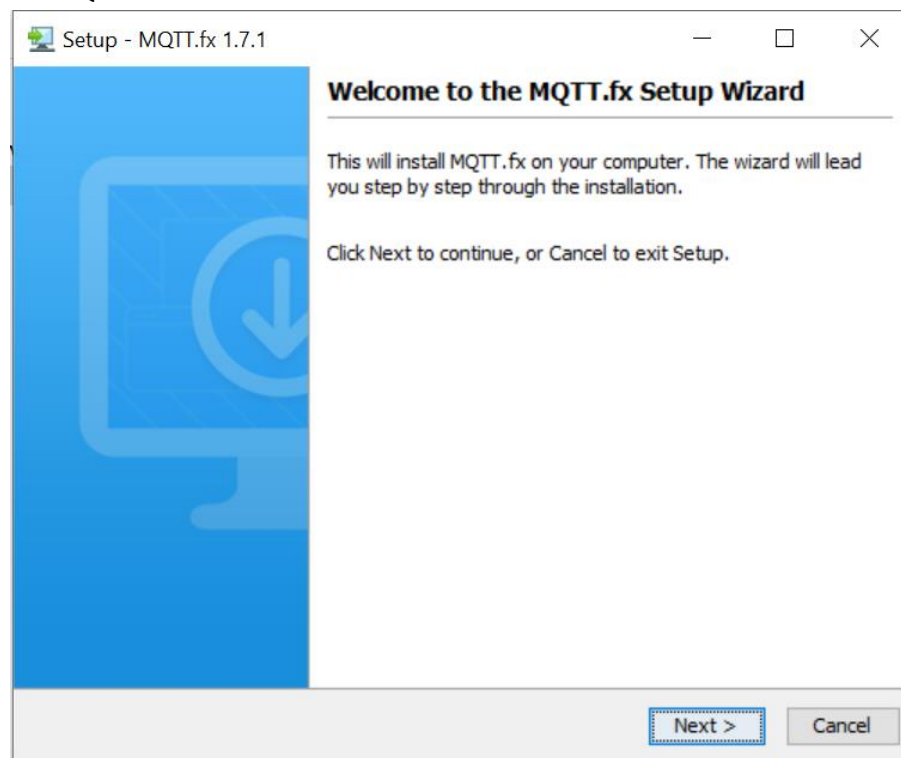
C:\Program Files\Mosquitto>

```

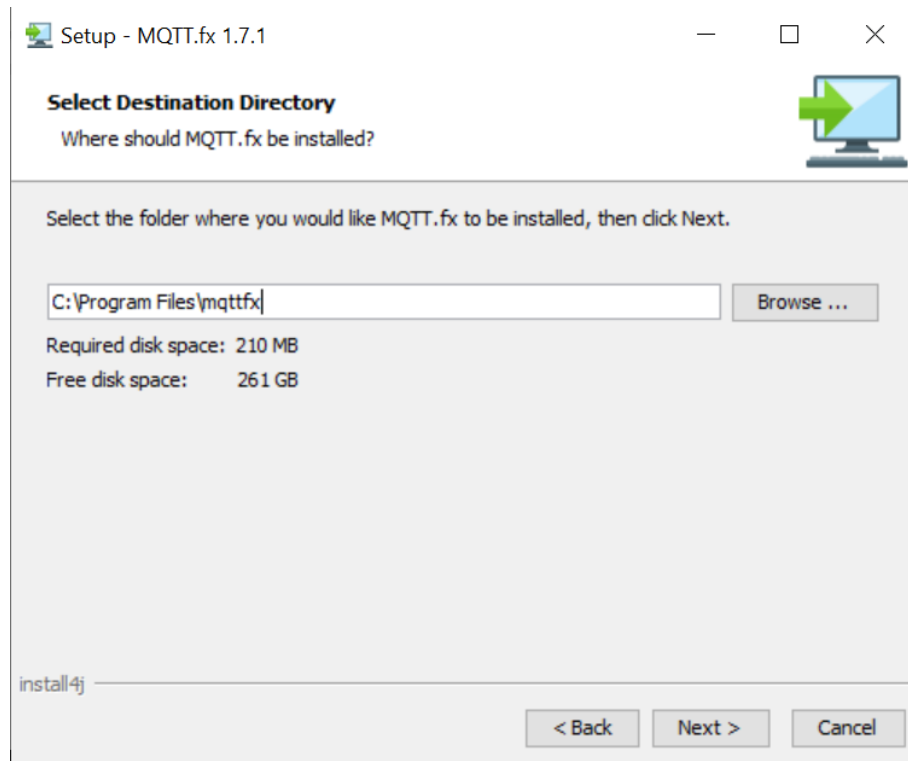
Gambar 12 Tampilan *Command Prompt* untuk menjalankan mosquitto broker

## 7. Install MQTT Client (MQTTfx, MQTTx)

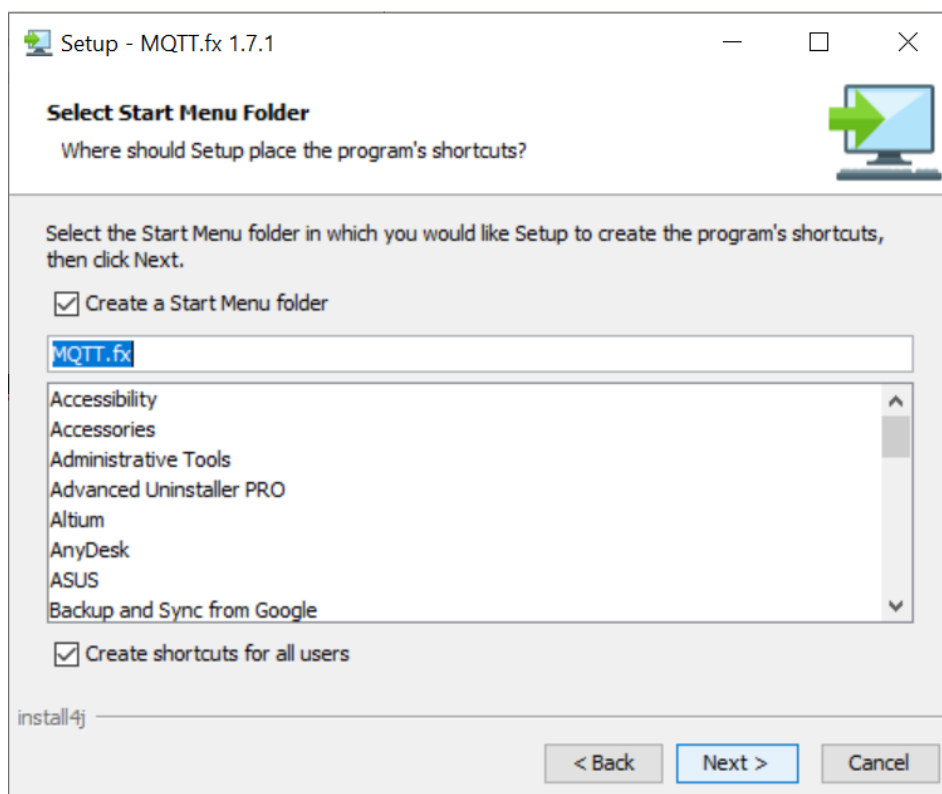
- Install MQTTfx:



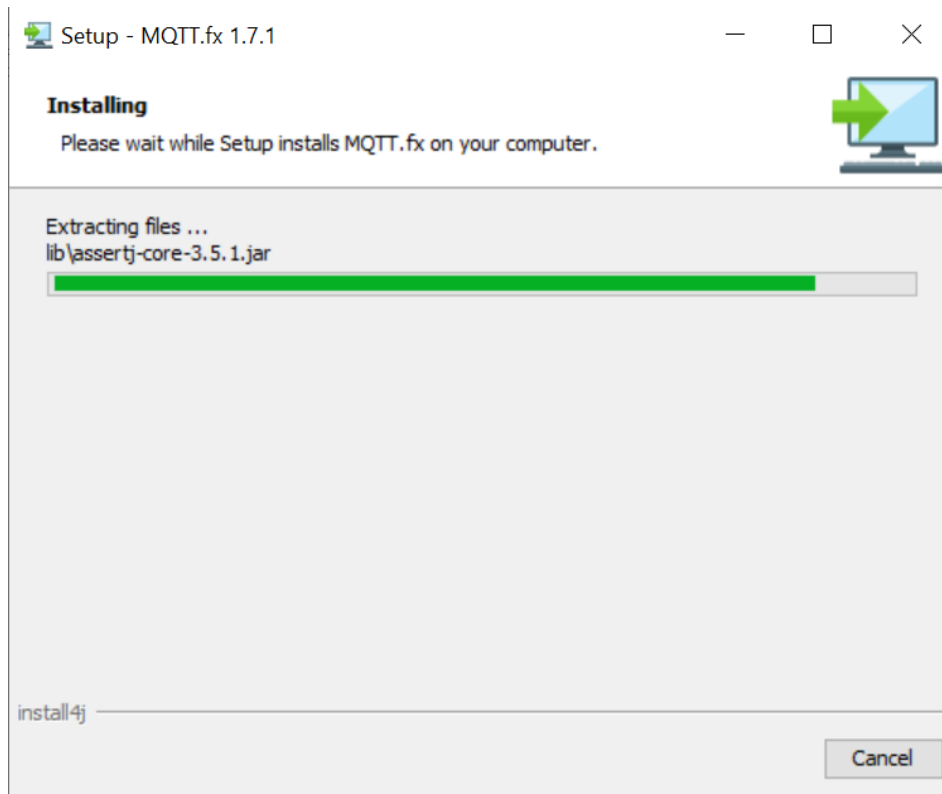
Gambar 13 Tampilan Install MQTTfx (Step 1)



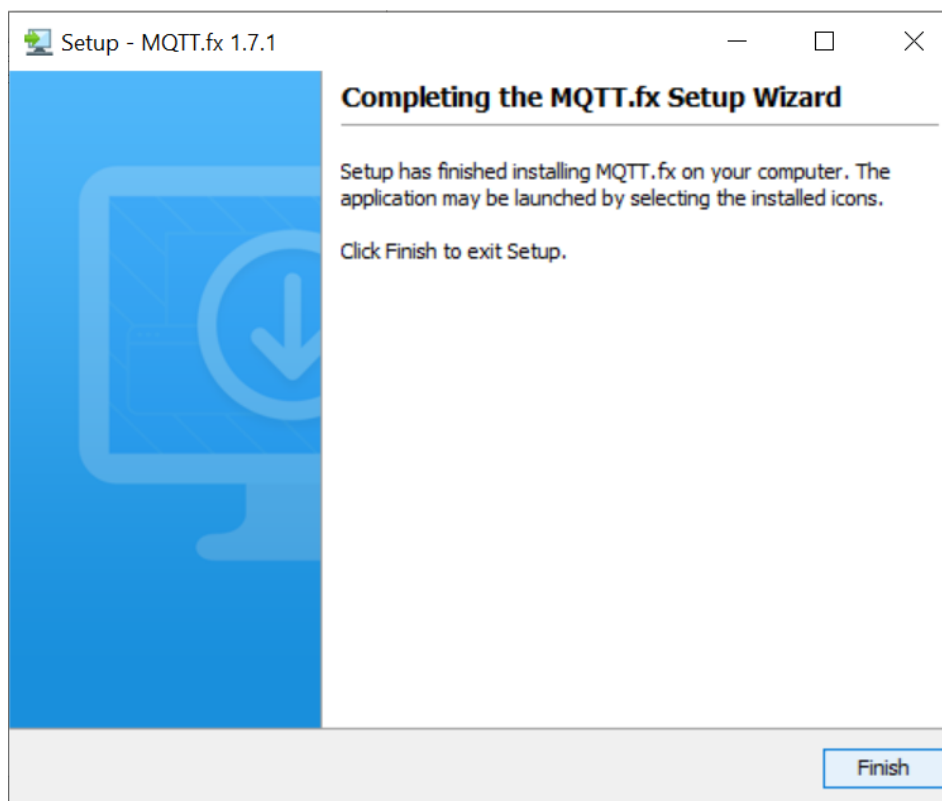
Gambar 14 Tampilan Install MQTTfx (Step 2)



Gambar 15 Tampilan Instal MQTTfx (Step 3)

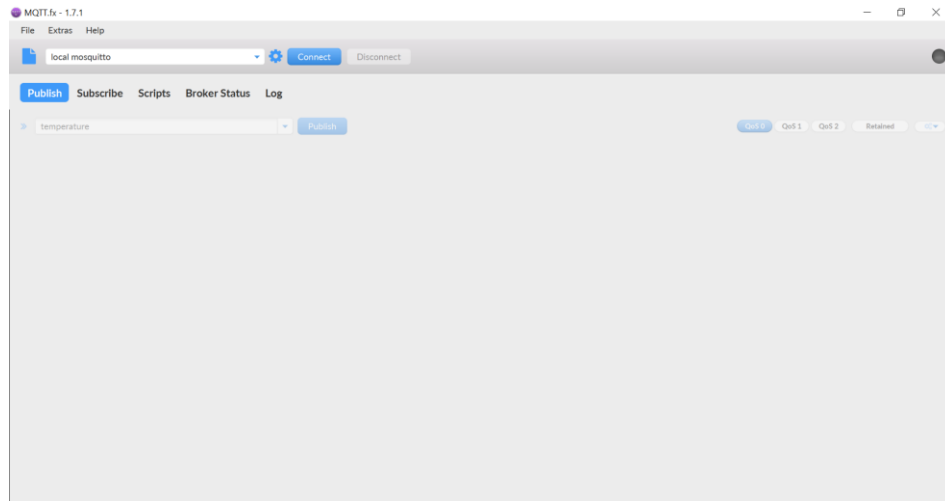


Gambar 16 Tampilan Install MQTTfx (Step 4)



Gambar 17 Tampilan Install MQTTfx (Step 5)

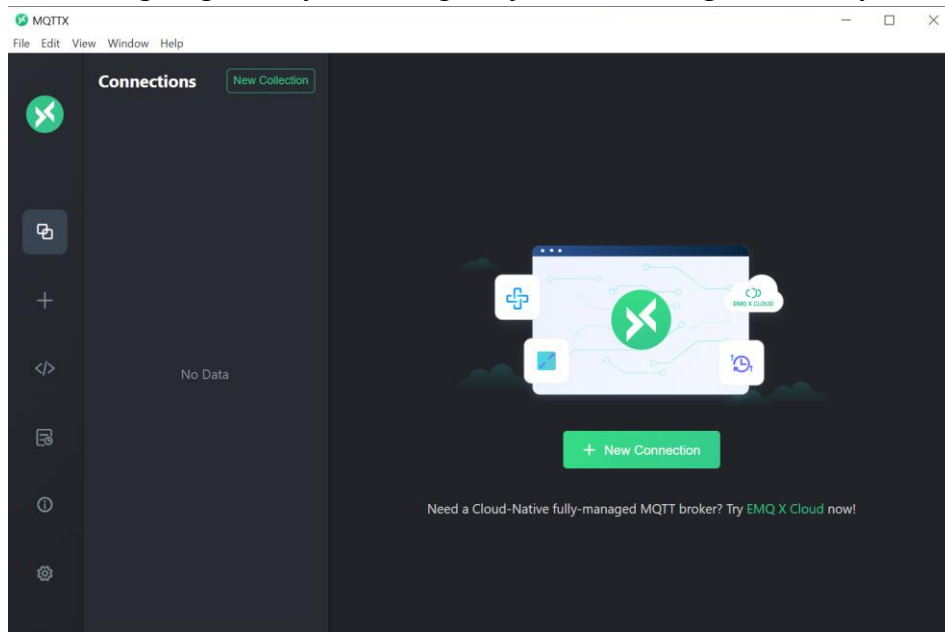
type: **MQTTfx** pada windows search > open



Gambar 18 Tampilan Akhir MQTTfx

- Install MQTTx

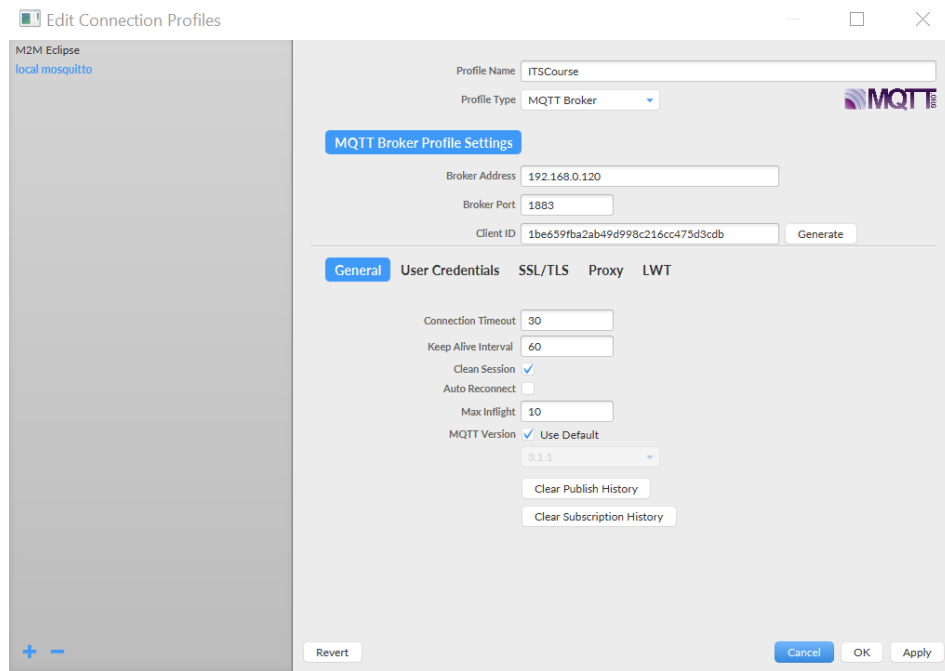
Untuk MQTTx karena ekstensi file **.exe**, maka tidak ada prosedur untuk instalasi, software langsung bisa dijalankan - *go to file MQTTx > right click > open*



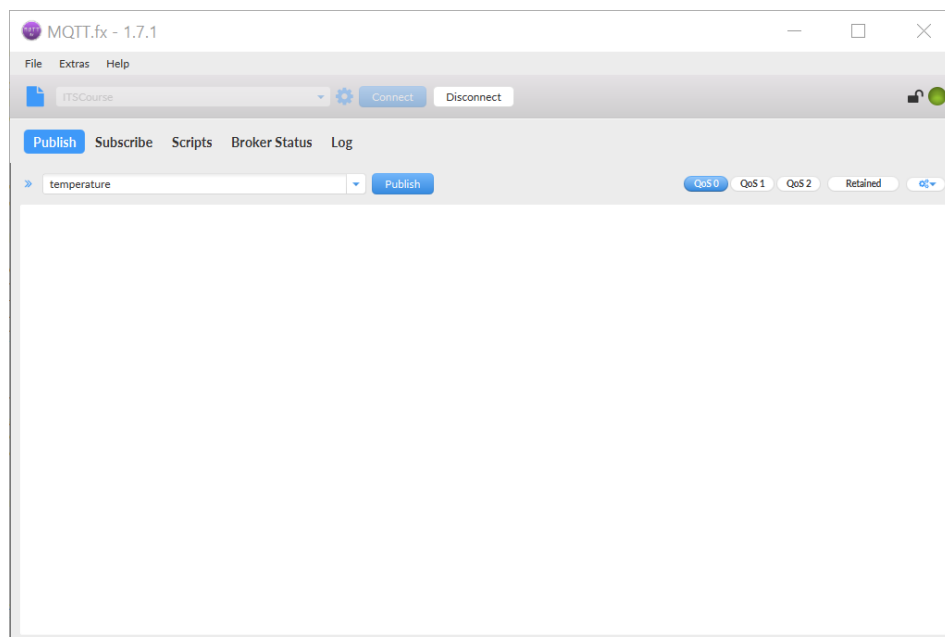
Gambar 19 Tampilan Akhir MQTTx

8. Lakukan konfigurasi sederhana MQTT Client terkoneksi ke Broker

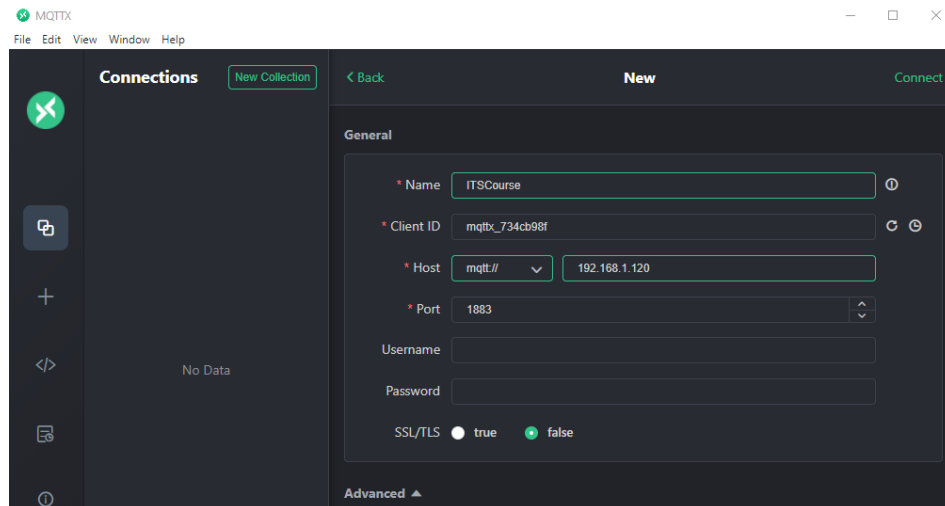
- Profile Name : blblblbl
- Profile Type : MQTT Broker
- Broker Address : 192.168.xxx.xxx, test.mosquitto.org, broker.emqx.io
- Broker Port : 1883



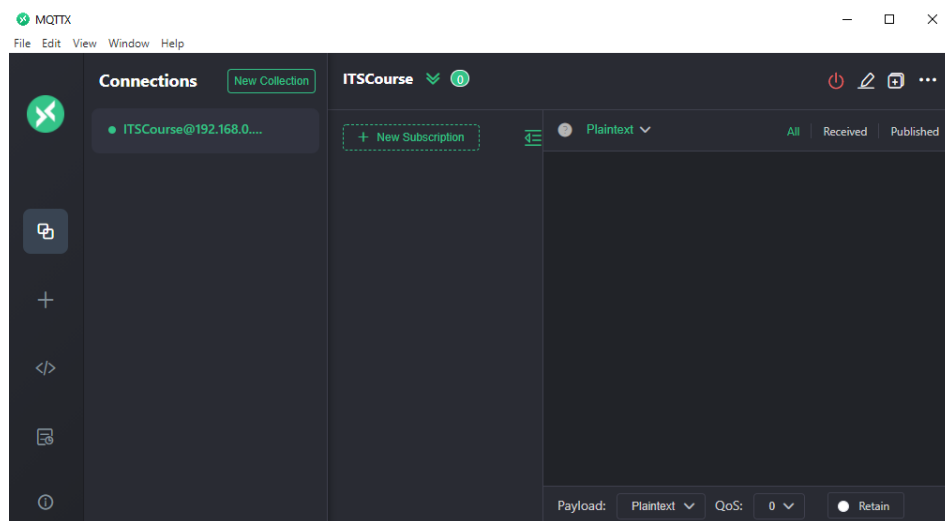
Gambar 20 Tampilan Konfigurasi MQTTfx



Gambar 21 Tampilan MQTTfx *Connected*



Gambar 22 Tampilan Konfigurasi MQTTx



Gambar 23 Tampilan MQTTx *Connected*

**! Jika muncul permasalahan MQTT Client tidak bisa terkoneksi**

- Cek *aditional file*: **libcrypto-1\_1\_x64.dll**, & **libssl-1\_1\_x64.dll**

Jika tidak ada file tersebut pada folder Mosquitto, copy file pada link URL download dan arahkan pada folder Mosquitto

Note 1: File bisa di dapatkan dengan install openssl pada link download berikut:

<https://slproweb.com/products/Win32OpenSSL.html>

Note 2: They are now included in the 1.5.8 install package

- konfigurasi file mosquitto.config:

per\_listener\_settings [false]

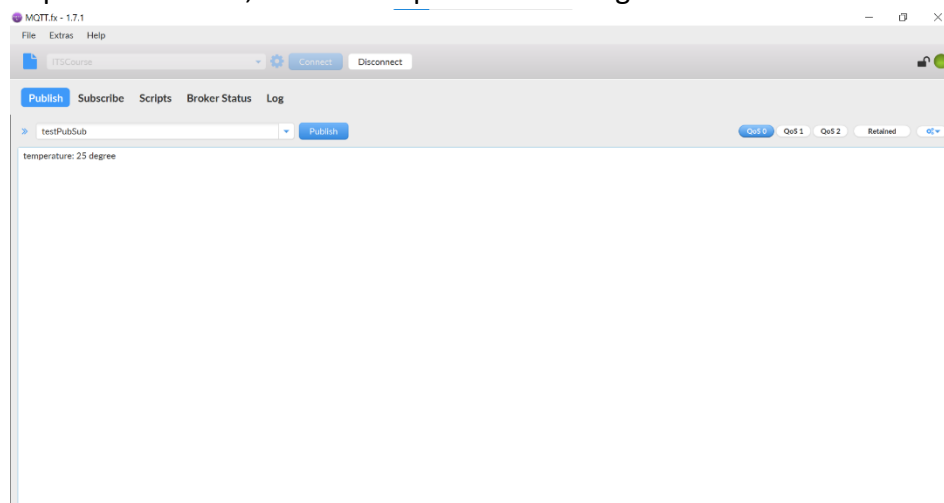
allow\_anonymous [true]

- Check windows service: *open service*> find: **mosquitto broker**> **test start/stop**

- Port cascade: <https://stackoverflow.com/questions/39632667/how-do-i-kill-the-process-currently-using-a-port-on-localhost-in-windows>

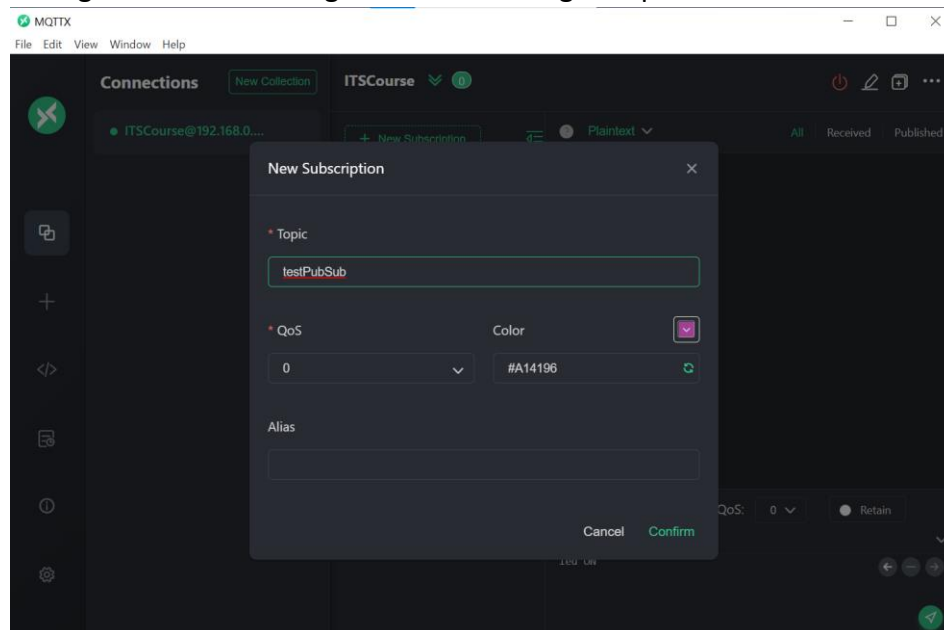
10. Test skema *publish* dan *subscribe* antar MQTT Client (MQTTfx & MQTTx)

- Konfigurasi MQTTfx sebagai *publisher* dengan,  
Topik: testPubSub, Pesan : temperature: 25 degree



Gambar 24 Tampilan Konfigurasi MQTTfx Sebagai *Publisher*

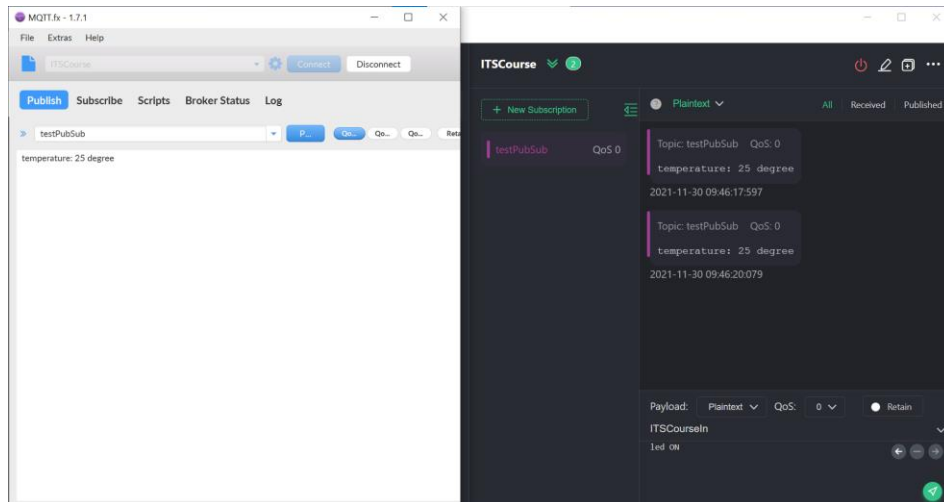
- Konfigurasi MQTTx sebagai *subscriber* dengan topik: testPubSub



Gambar 25 Tampilan Konfigurasi MQTTx Sebagai *Subscriber*



- Cek skema *publish* dan *subscribe* apakah telah berhasil dengan cara mengecek pesan yang diterima MQTTx sebagai *subscriber*



Gambar 26 Tampilan Skema *Publish* & *Subscribe* antar MQTT Client Berhasil

#### 11. Programming pada Arduino IDE – **Open Arduino IDE > File > New**

- Create program test Wifi connection
- Create program MQTT connection

#### E. Studi Kasus

1. Create program untuk skema publish dan subscribe antara ESP8266/ESP32 dengan MQTTfx/MQTTx pada computer/laptop anda sendiri
2. Create program untuk skema publish dan subscribe ESP8266/ESP32 dengan MQTTfx/MQTTx antar computer/laptop rekan pelatihan anda

Ex: Computer 1: ESP8266 & Broker Configuration

Computer 2: MQTTx

Note:

Contoh program Arduino untuk MQTT for IoT dapat diakses pada link berikut.

URL: <https://github.com/dhikihandika/MQTTforIoT-Internet-of-Things->