Nama : Ludang Prasetyo Nugroho [Teknik Komputer ( S1)](https://www.utdi.ac.id/)

Nim : 225510017

Matkul : Prak System IOT

**# PRAKTIKUM**

# Persiapan

* 1. Download aplikasi MQTT.Fix (

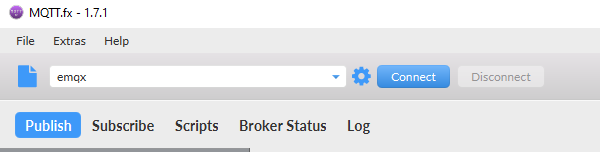
https://web.archive.org/web/20210514230412/https:/

[/www.jensd.de/apps/mqttfx/1.7.1/](http://www.jensd.de/apps/mqttfx/1.7.1/)

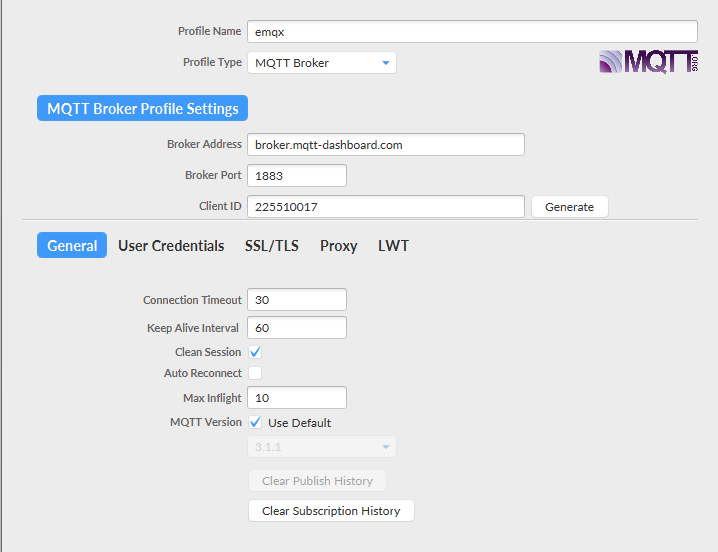
* 1. Install MQTT.Fix
  2. Tambahkan library Arduino PubSubClient melalui Manager Library

# Publish dan Subcribe Menggunakan MQTT.Fix

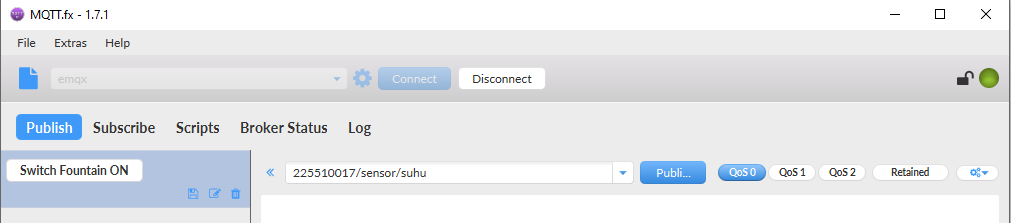
* 1. Atur Server MQTT (Broker) yang akan digunakan dengan menekan gambar gear.



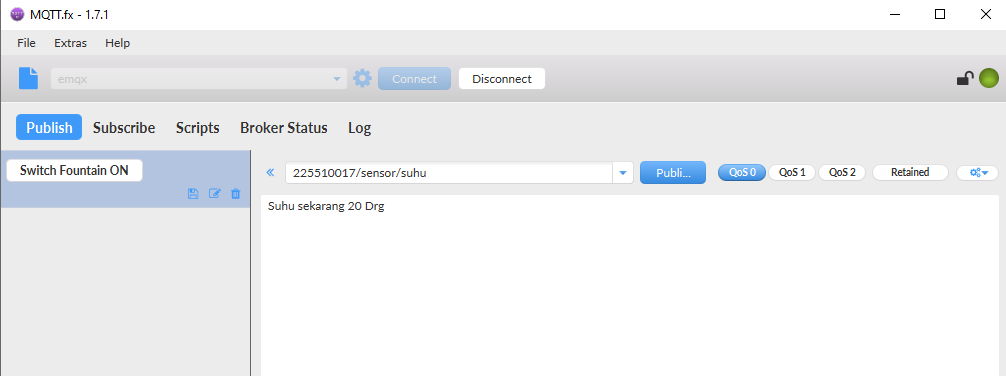
* 1. Isikan server broker dll seperti pada Gambar berikut ini.



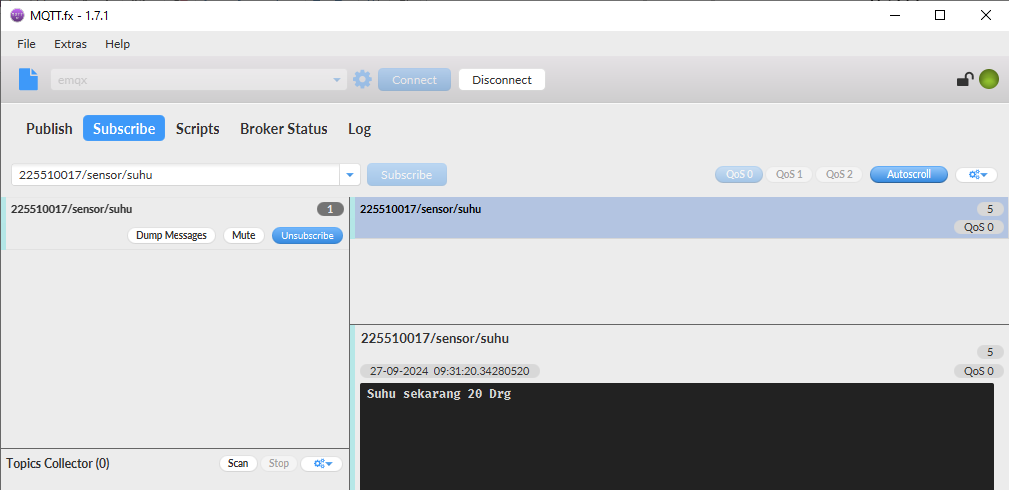
* 1. Tekan Tombol Connect agar terkoneksi dengan server MQTT



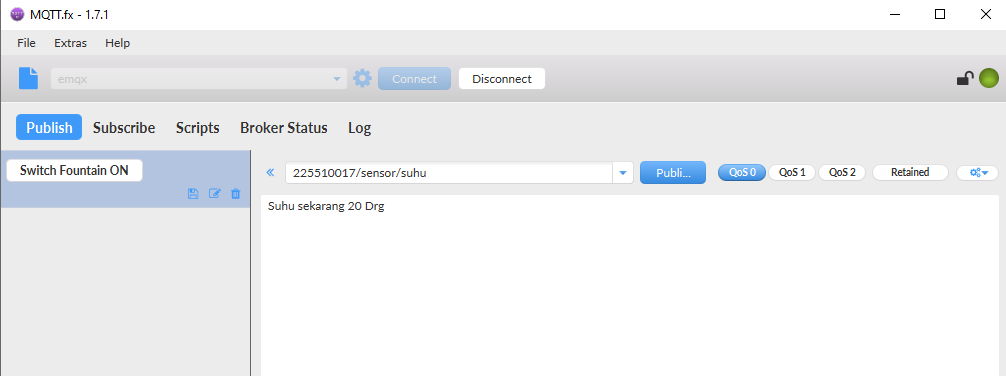
* 1. Bekerjalah berpasangan dengan teman.
  2. Tekan menu Publish dan isikan di kotak Text edit sebuah topic seperti gambar berikut ini.



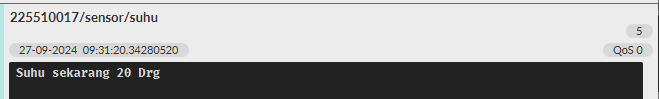
* 1. Untuk teman pasangannya tekan menu Subcribe.



* 1. Untuk yang Publish, ketik angka atau kalimat di text area kemudian tekan tombol Publish.

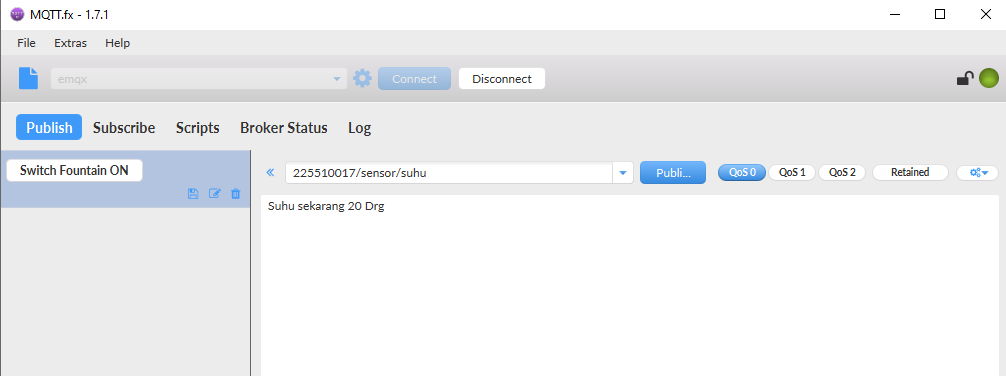


* 1. Untuk yang Subcribe perhatikan tampilan console di bawah.

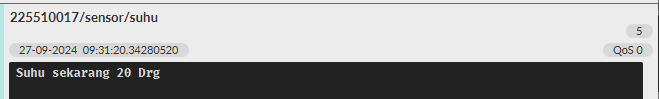


* 1. Anda uji coba dengan melakukan Publish topic temannya yang lain dan Subcribe temannya yang lain.

Topik yang di public

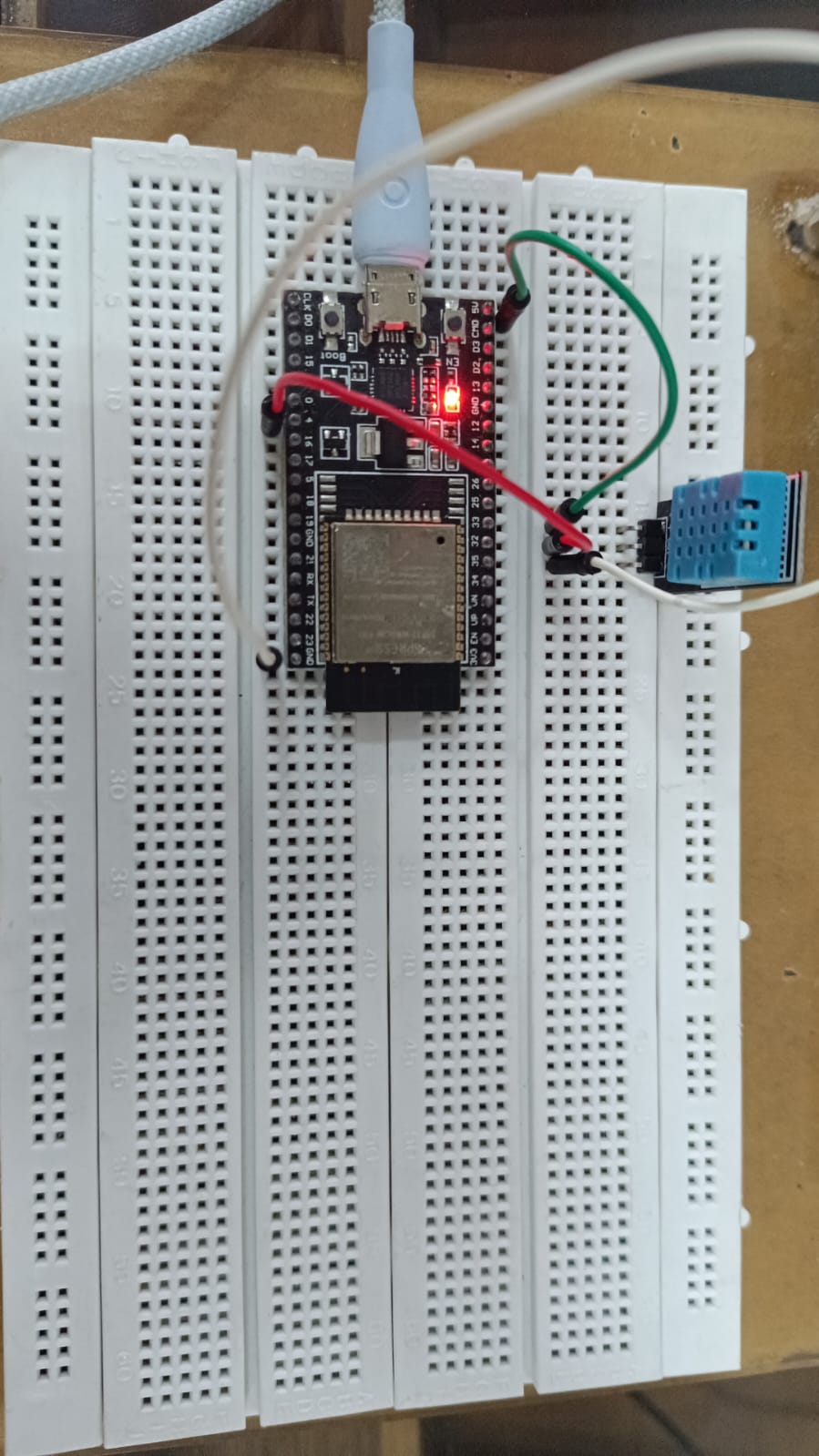


Yang di lihat teman



# Modul Dev ESP32 Publish ke Server MQTT

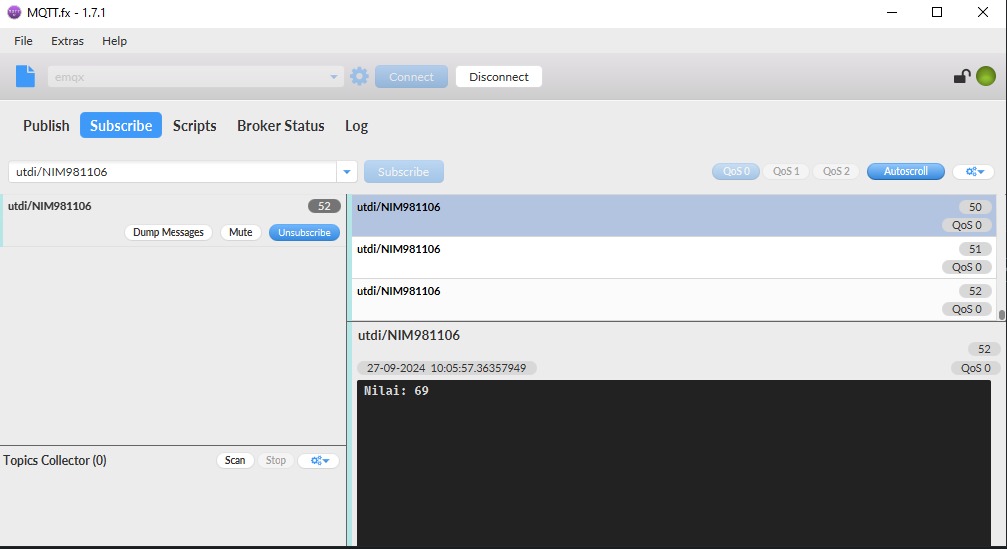
* 1. Siapkan modul Dev ESP32
  2. Buat rangkaian seperti program berikut ini



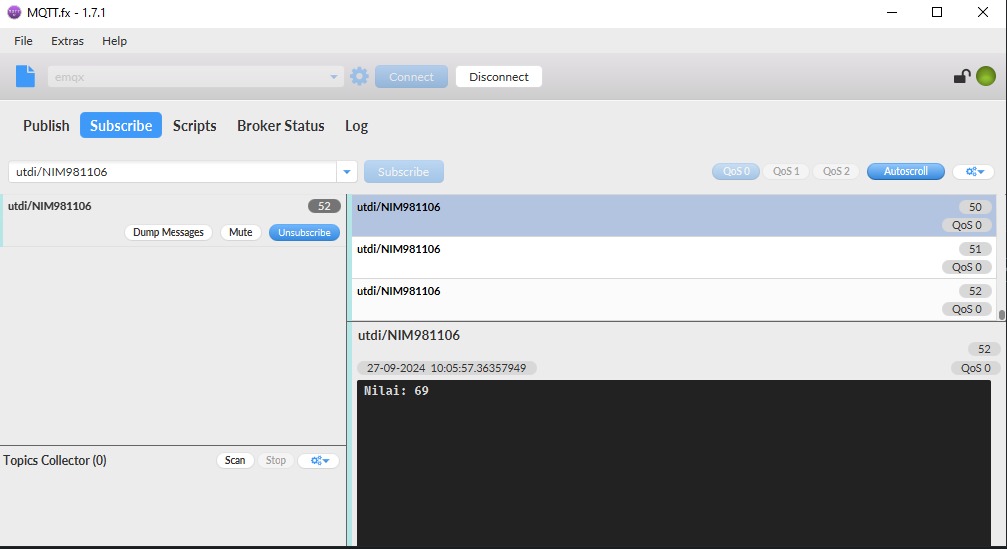
* 1. Copy dan upload program berikut ini.

|  |
| --- |
| #include <WiFi.h> |
| #include <PubSubClient.h> |
|  |
| #define pinAnalog 34 // Ganti dengan pin analog yang sesuai |
|  |
| const char\* ssid = "RPLA\_2.4"; |
| const char\* password = "utdijogja"; |
| const char\* mqtt\_server = "broker.mqtt-dashboard.com"; |
|  |
| WiFiClient espClient; |
| PubSubClient client(espClient); |
|  |
| char msg[50]; |
| long wktYll; |
|  |
| void setup\_wifi() { |
| Serial.println(); |
| Serial.print("Koneksi ke "); |
| Serial.println(ssid); |
| WiFi.begin(ssid, password); |
|  |
| while (WiFi.status() != WL\_CONNECTED) { |
| delay(500); |
| Serial.print("."); |
| } |
| Serial.println(""); |
| Serial.println("WiFi connected"); |
| Serial.print("IP address: "); |
| Serial.println(WiFi.localIP()); |
| } |
|  |
| void reconnectMQTT() { |
| 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"); |
| } 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(); |
|  |
| if (WiFi.status() != WL\_CONNECTED) { |
| Serial.println("Wifi terputus, mencoba reconnect..."); |
| setup\_wifi(); // Coba reconnect jika terputus |
| } |
|  |
| if (!client.connected()) { |
| reconnectMQTT(); |
| } |
|  |
| if (now - wktYll > 2000) { |
| wktYll = now; |
| int dataAnalogAsal = analogRead(pinAnalog); |
| snprintf(msg, 50, "Nilai: %d", dataAnalogAsal); // Menggunakan %d untuk integer |
|  |
| if (WiFi.status() == WL\_CONNECTED) { |
| Serial.print("Publish message: "); |
| Serial.println(msg); |
| client.publish("utdi/NIM981106", msg); |
| } else { |
| Serial.println("Wifi terputus"); |
| } |
| } |
|  |
| client.loop(); // Pastikan untuk memanggil client.loop() untuk menjaga koneksi MQTT |
| } |

* 1. Sesuaikan MQTT.Fix subcribenya dengan Publish dari nodeESP32

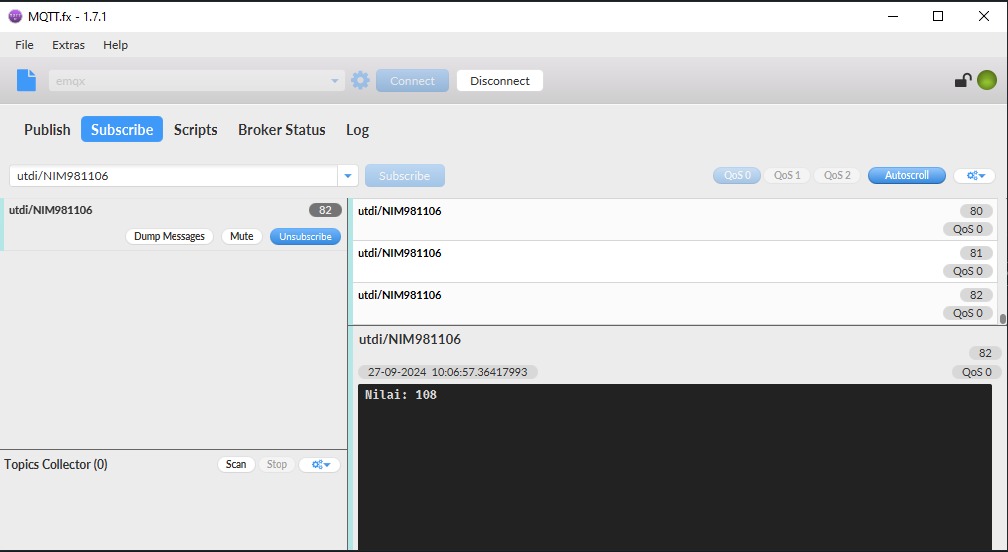


* 1. Dan amati data yang di terima.



Hasik keluaransuhunya munuldi MQTT

* 1. Coba dengan server/broker yang lainnya.

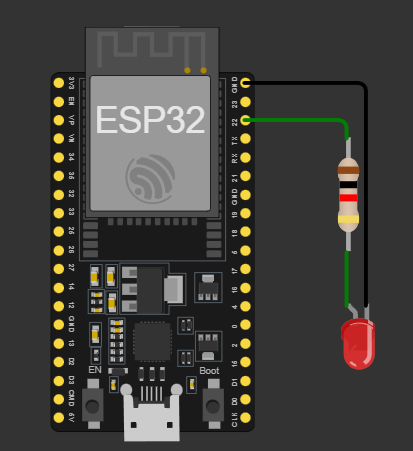
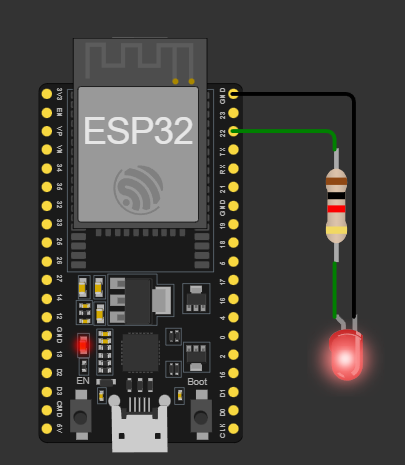


# Modul ESP32 Subcribe ke Server MQTT

* 1. Copy dan upload program berikut ini.

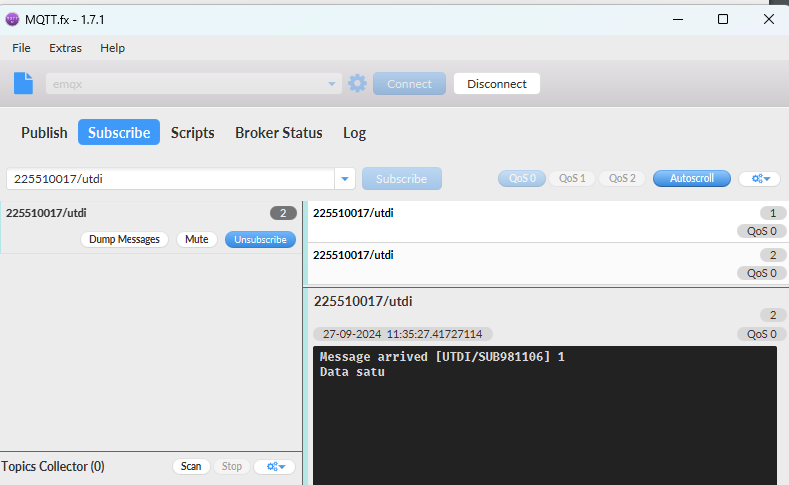
|  |
| --- |
|  |
| #include <WiFi.h> #include <PubSubClient.h> |
|  |
| #define LED1 22 |
|  |
| const char\* ssid = "harjolukito"; const char\* password = "ami1971da"; |
| const char\* mqtt\_server = "test.mosquitto.org"; |
|  |
| WiFiClient espClient; PubSubClient client(espClient); |
|  |
| char 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(); |
| } |
| } |

* 1. Rangkaianya Saat LEDnyamenyala dan mati

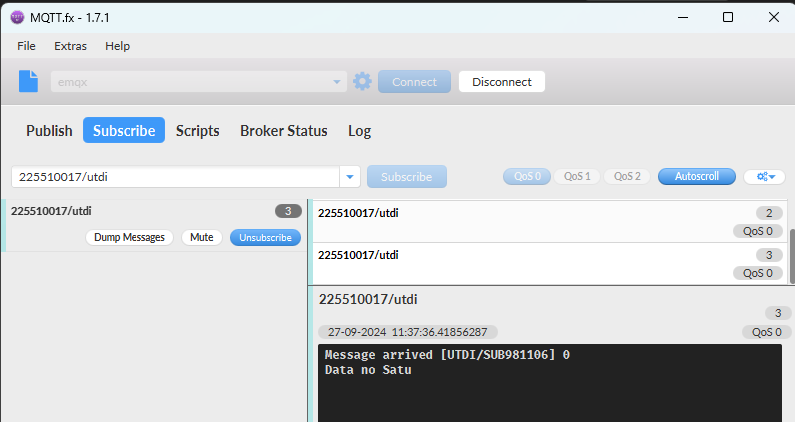
 

* 1. Lakukan publish via MQTTFx untuk mengendalikan LED1.

Bila LED nya Menyala



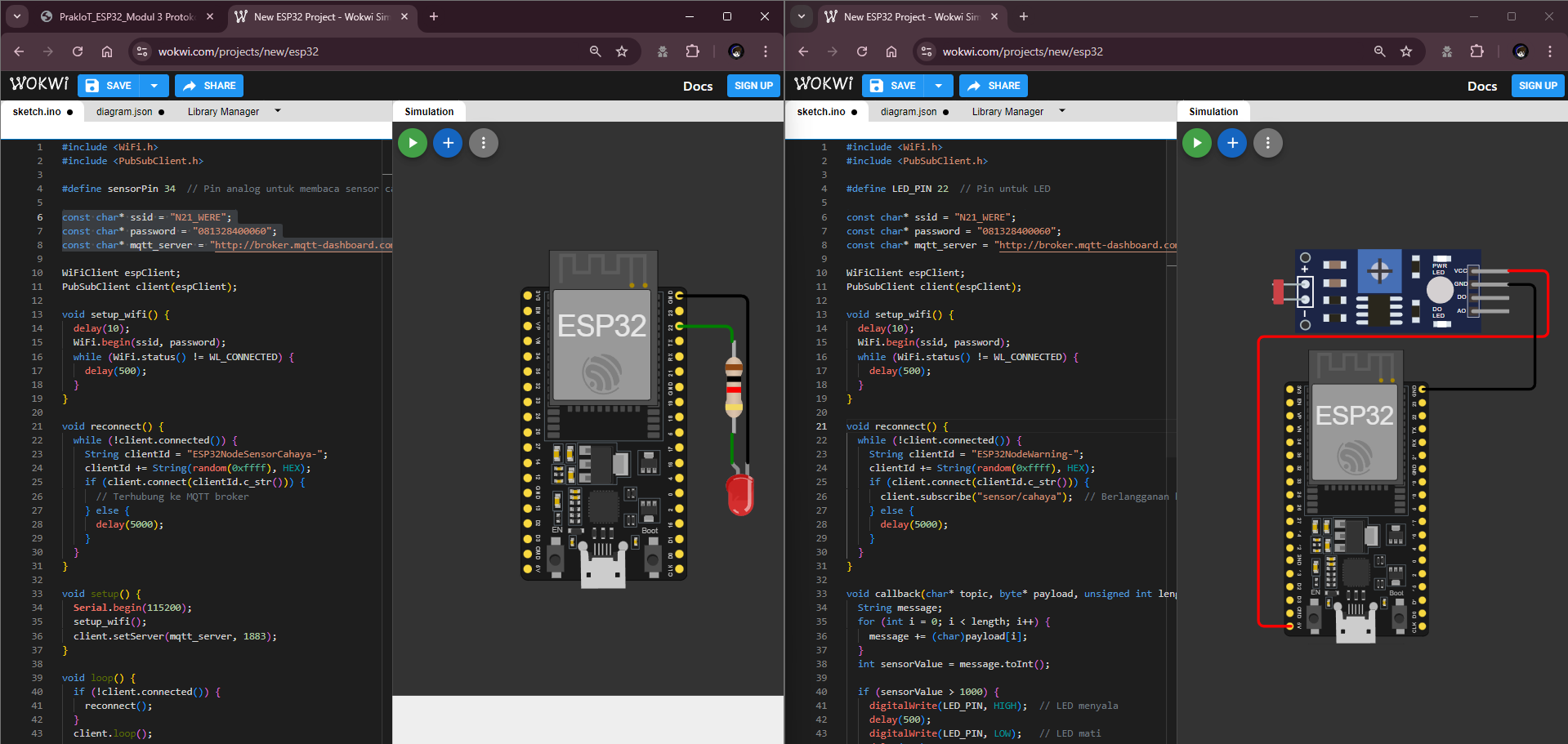
Bila LED nya Mati



**# 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 berkdip jika data yang diterima melebihi 1000.

Btw saya mengerjakan tugas ini sendiri jadi saya memakai 2 wokwi



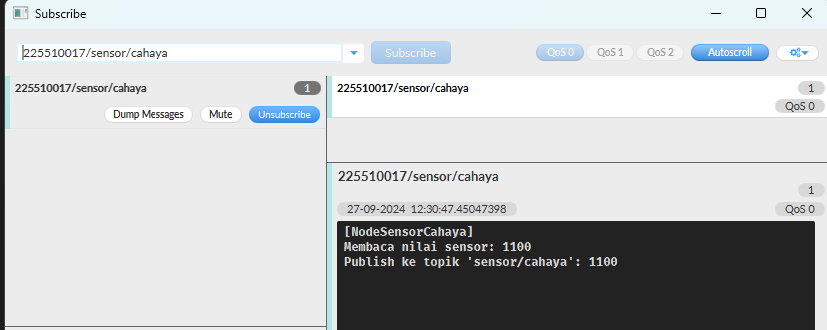
Code program **NodeSensorCahaya** (Pembaca sensor cahaya dan mengirim data via MQTT)

|  |
| --- |
| #include <WiFi.h> |
| #include <PubSubClient.h> |
|  |
| #define sensorPin 34 // Pin analog untuk membaca sensor cahaya |
|  |
| const char\* ssid = "N21\_WERE"; |
| const char\* password = "081328400060"; |
| const char\* mqtt\_server = "http://broker.mqtt-dashboard.com"; |
|  |
| WiFiClient espClient; |
| PubSubClient client(espClient); |
|  |
| void setup\_wifi() { |
| delay(10); |
| WiFi.begin(ssid, password); |
| while (WiFi.status() != WL\_CONNECTED) { |
| delay(500); |
| } |
| } |
|  |
| void reconnect() { |
| while (!client.connected()) { |
| String clientId = "ESP32NodeSensorCahaya-"; |
| clientId += String(random(0xffff), HEX); |
| if (client.connect(clientId.c\_str())) { |
| // Terhubung ke MQTT broker |
| } else { |
| delay(5000); |
| } |
| } |
| } |
|  |
| void setup() { |
| Serial.begin(115200); |
| setup\_wifi(); |
| client.setServer(mqtt\_server, 1883); |
| } |
|  |
| void loop() { |
| if (!client.connected()) { |
| reconnect(); |
| } |
| client.loop(); |
|  |
| int sensorValue = analogRead(sensorPin); // Baca nilai sensor cahaya |
| String sensorData = String(sensorValue); |
|  |
| client.publish("sensor/cahaya", sensorData.c\_str(225510017/sensor/cahaya)); // Publish data ke topik MQTT |
|  |
| delay(1000); // Tunggu 1 detik sebelum baca lagi |
| } |

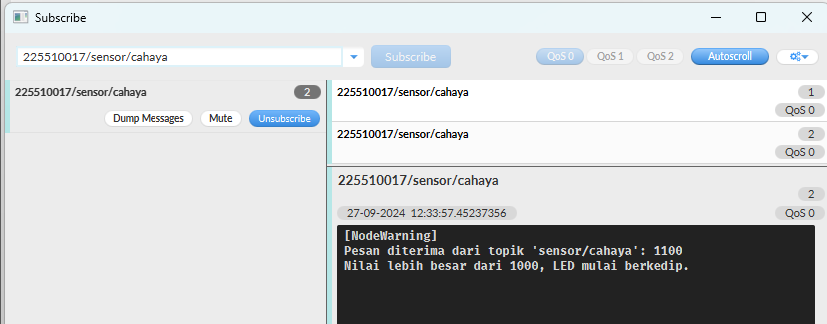
Code program **NodeWarning (Menerima data dan kontrol LED)**

|  |
| --- |
| #include <WiFi.h> |
| #include <PubSubClient.h> |
|  |
| #define LED\_PIN 22 // Pin untuk LED |
|  |
| const char\* ssid = "N21\_WERE"; |
| const char\* password = "081328400060"; |
| const char\* mqtt\_server = "http://broker.mqtt-dashboard.com"; |
|  |
| WiFiClient espClient; |
| PubSubClient client(espClient); |
|  |
| void setup\_wifi() { |
| delay(10); |
| WiFi.begin(ssid, password); |
| while (WiFi.status() != WL\_CONNECTED) { |
| delay(500); |
| } |
| } |
| void reconnect() { |
| while (!client.connected()) { |
| String clientId = "ESP32NodeWarning-"; |
| clientId += String(random(0xffff), HEX); |
| if (client.connect(clientId.c\_str())) { |
| client.subscribe("sensor/cahaya"); // Berlangganan ke topik sensor cahaya |
| } else { |
| delay(5000); |
| } |
| } |
| } |
|  |
| void callback(char\* topic, byte\* payload, unsigned int length) { |
| String message; |
| for (int i = 0; i < length; i++) { |
| message += (char)payload[i]; |
| } |
| int sensorValue = message.toInt(); |
|  |
| if (sensorValue > 1000) { |
| digitalWrite(LED\_PIN, HIGH); // LED menyala |
| delay(500); |
| digitalWrite(LED\_PIN, LOW); // LED mati |
| delay(500); |
| } |
| } |
|  |
| void setup() { |
| pinMode(LED\_PIN, OUTPUT); // Konfigurasi LED sebagai output |
| Serial.begin(115200); |
| setup\_wifi(); |
| client.setServer(mqtt\_server, 1883); |
| client.setCallback(callback); |
| } |
|  |
| void loop() { |
| if (!client.connected(225510017/sensor/cahaya)) { |
| reconnect(); |
| } |
| client.loop(); |
| } |

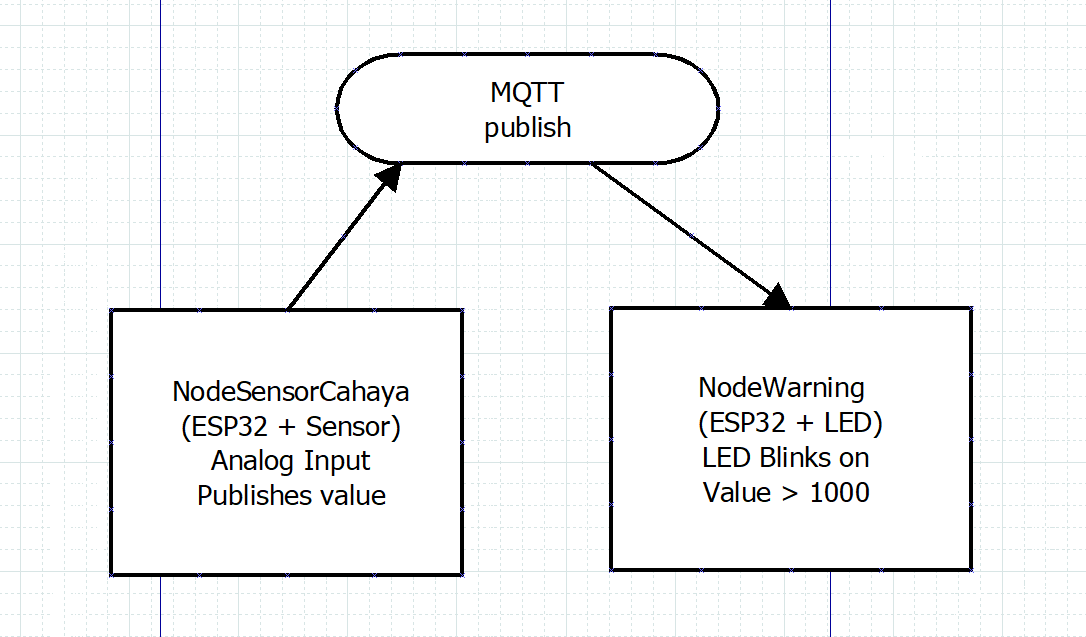
Keluaran NodeSensorCahaya



**Keluaran NodeWarning**

****

1. Gambarkan diagram sistem dari soal 1.



**# TUGAS**

1. Buat diagram alir nodeSensor dan nodeWarning

Diagram Alir NodeSensorCahaya

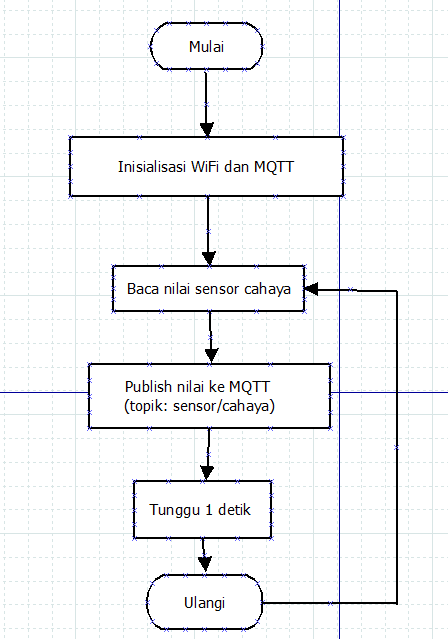


Diagram Alir NodeWarning

