

Nama : Ludang Prasetyo Nugroho
[Komputer \(S1\)](#)
Nim : 225510017
Matkul : Prak System IOT

[Teknik](#)

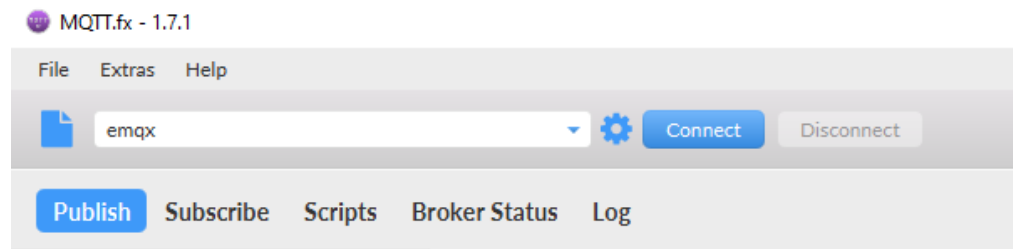
PRAKTIKUM

A. Persiapan

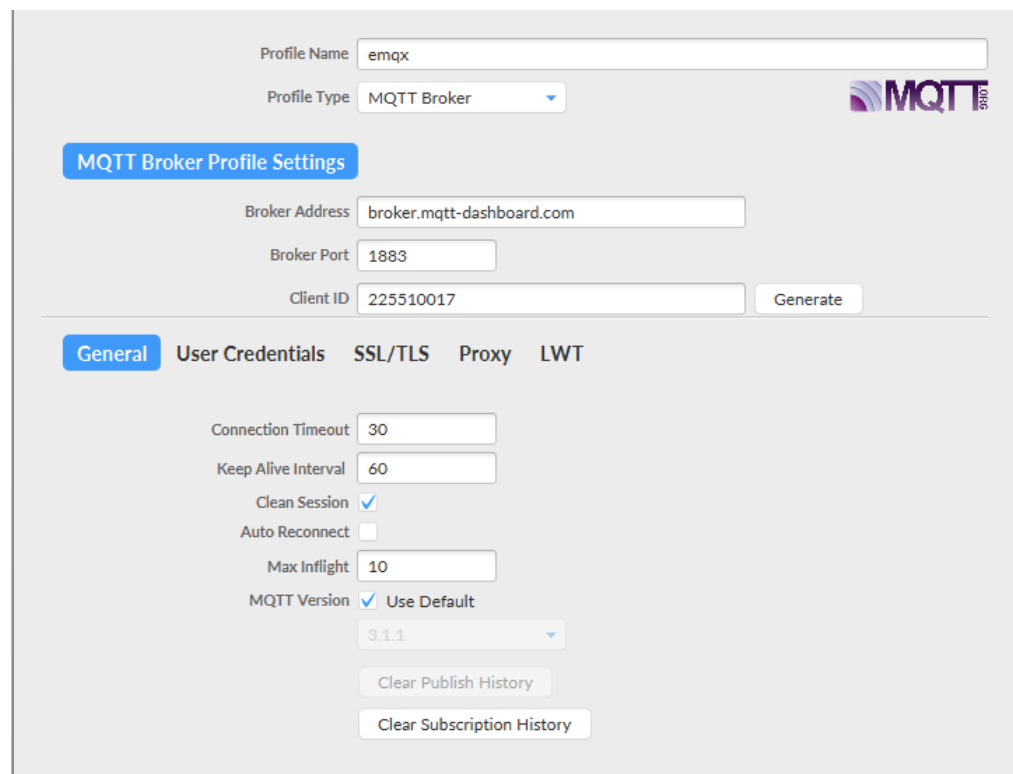
1. Download aplikasi MQTT.Fix (<https://web.archive.org/web/20210514230412/https://www.jensd.de/apps/mqttfx/1.7.1/>)
2. Install MQTT.Fix
3. Tambahkan library Arduino PubSubClient melalui Manager Library

B. Publish dan Subcribe Menggunakan MQTT.Fix

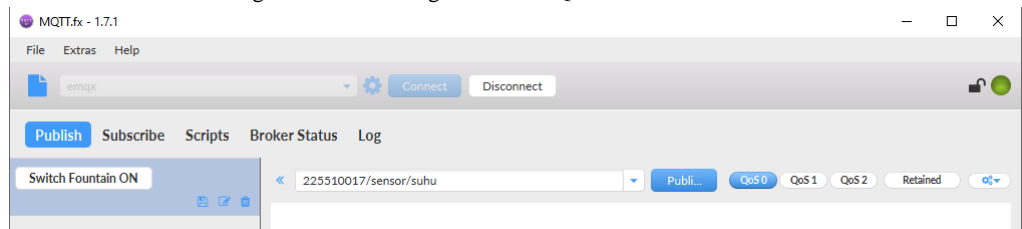
1. Atur Server MQTT (Broker) yang akan digunakan dengan menekangambar gear.



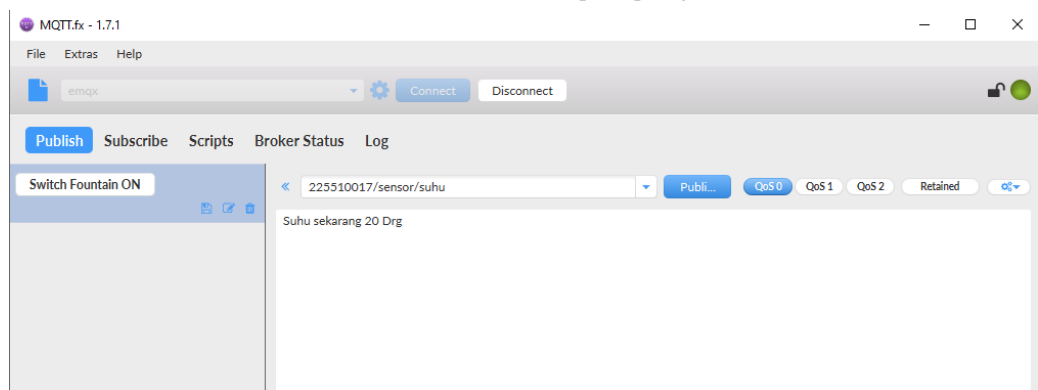
2. Isikan server broker dll seperti pada Gambar berikut ini.



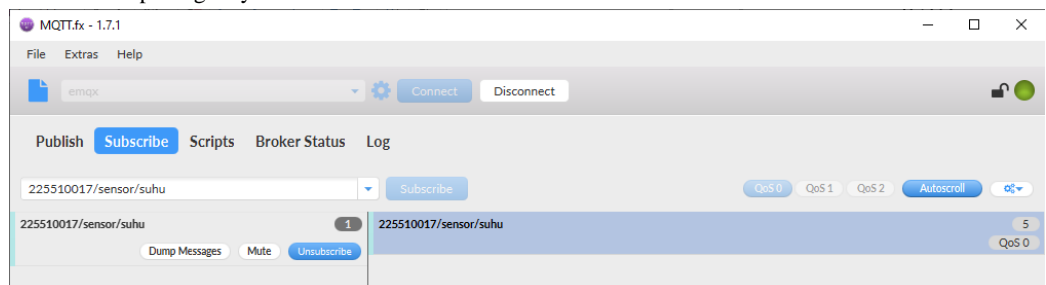
3. Tekan Tombol Connect agar terkoneksi dengan server MQTT



4. Bekerjalah berpasangan dengan teman.
5. Tekan menu Publish dan isikan di kotak Text edit sebuah topic sepertigambar berikut ini.



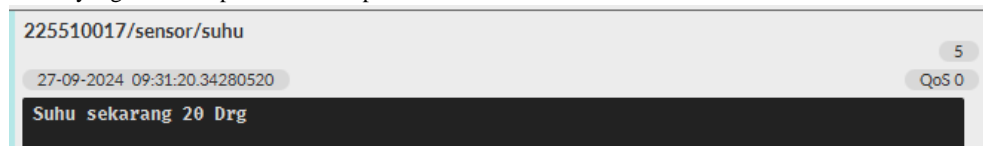
6. Untuk teman pasangannya tekan menu Subcribe.



7. Untuk yang Publish, ketik angka atau kalimat di text area kemudian tekantombol Publish.

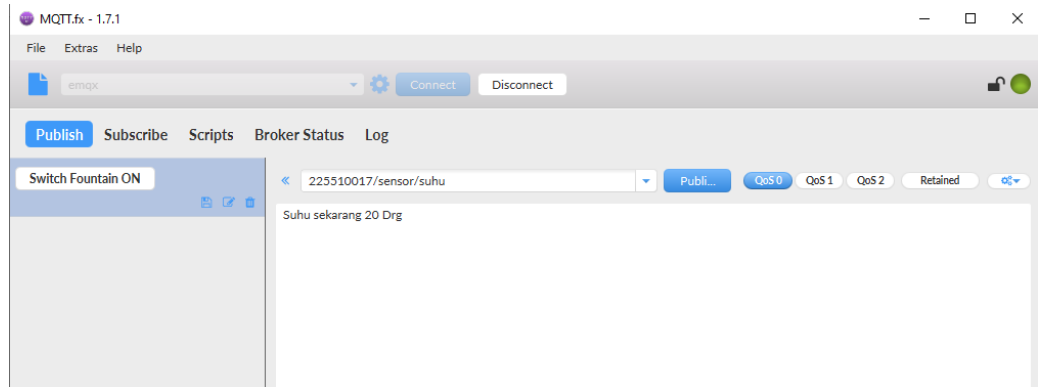


8. Untuk yang Subcribe perhatikan tampilan console di bawah.

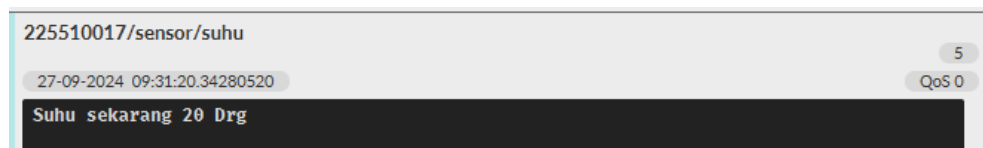


9. Anda uji coba dengan melakukan Publish topic temannya yang lain dan Subscribe temannya yang lain.

Topik yang di public

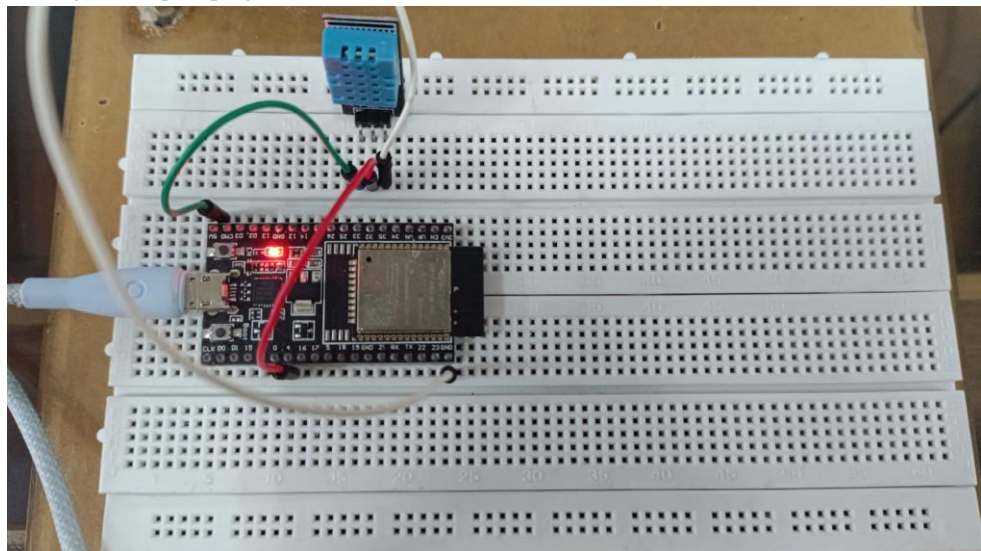


Yang di lihat teman



D. Modul Dev ESP32 Publish ke Server MQTT

- Siapkan modul Dev ESP32
- Buat rangkaian seperti program berikut ini



- 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);
        }
    }
}

```

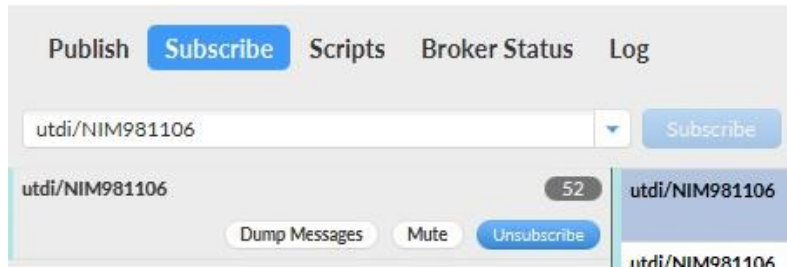
```

        client.publish("utdi/NIM981106", msg);
    } else {
        Serial.println("Wifi terputus");
    }
}

client.loop(); // Pastikan untuk memanggil client.loop() untuk menjaga koneksi MQTT
}

```

d. Sesuaikan MQTT.Fix subscribenya dengan Publish dari nodeESP32

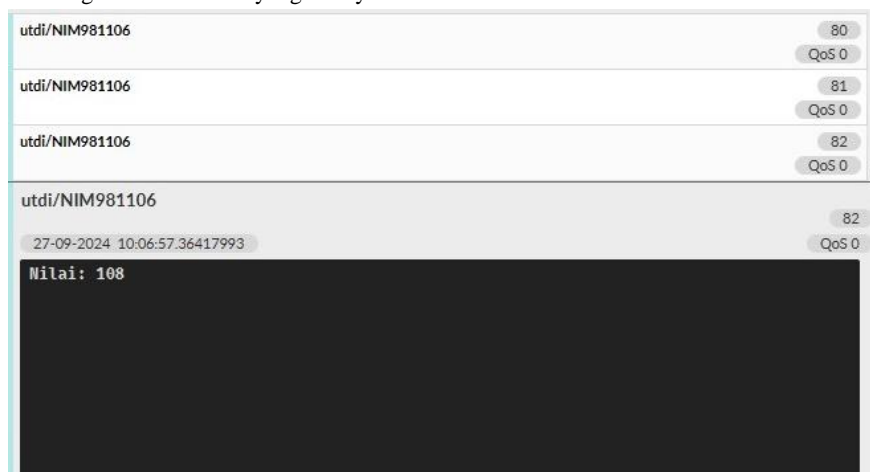


e. Dan amati data yang di terima.



Hasil keluarannya munuldi MQTT

f. Coba dengan server/broker yang lainnya.



LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

E. Modul ESP32 Subscribe ke Server MQTT

- a. 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("] ");
```

LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

```
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
}

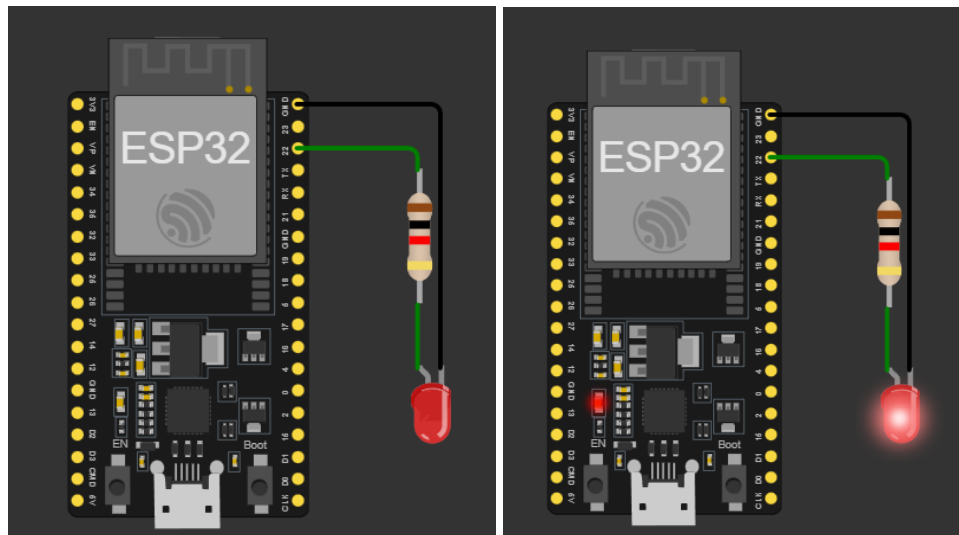
}

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();
}
}
```

b. Rangkaianya Saat LED Nyalamati dan mati

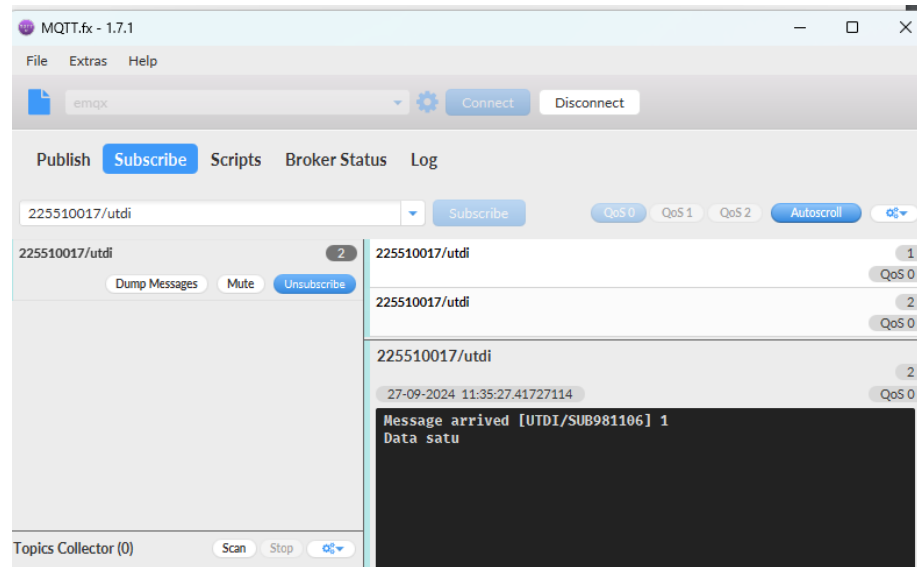


LAPORAN

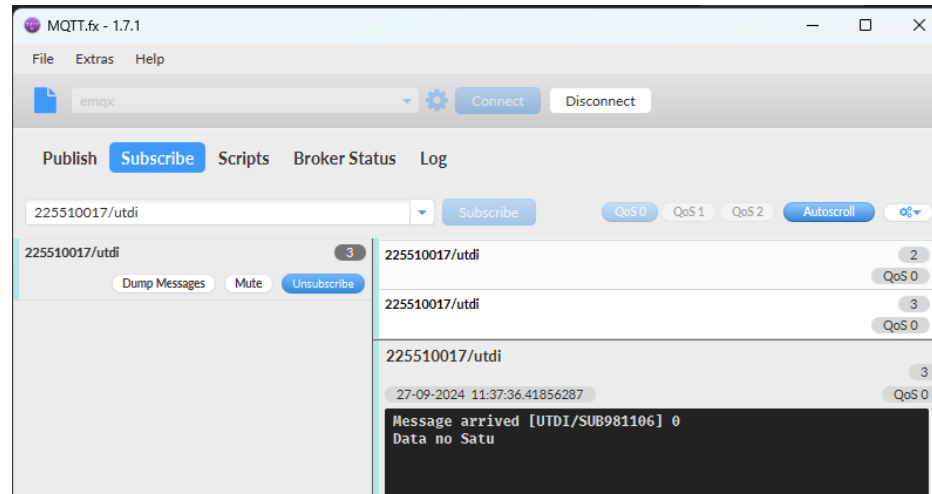
Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

- c. Lakukan publish via MQTTfx untuk mengendalikan LED1.

Bila LED nya Menyala



Bila LED nya Mati



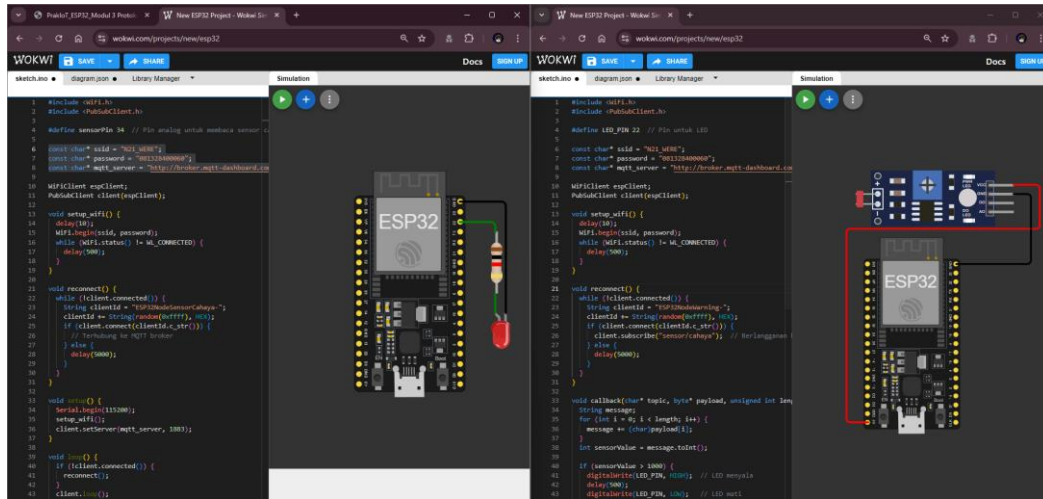
LAPORAN

Laporan praktikum Systemn Internet Of Things (Jumat 27 September)

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.

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() {
```

LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

```
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);
```

LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

```
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);
    }
}

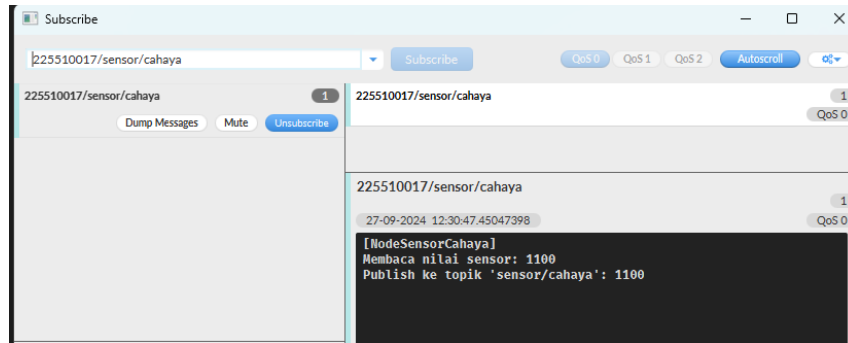
void setup() {
    pinMode(LED_PIN, OUTPUT); // Konfigurasi LED sebagai output
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

void loop() {
    if (!client.connected(225510017/sensor/cahaya)) {
        reconnect();
    }
    client.loop();
}
```

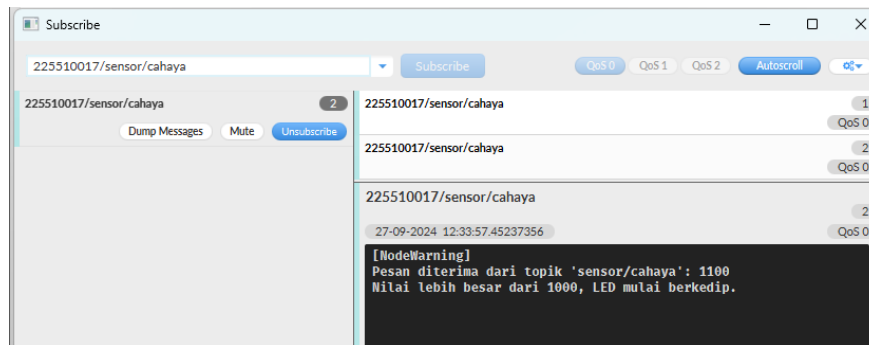
LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

Keluaran NodeSensorCahaya



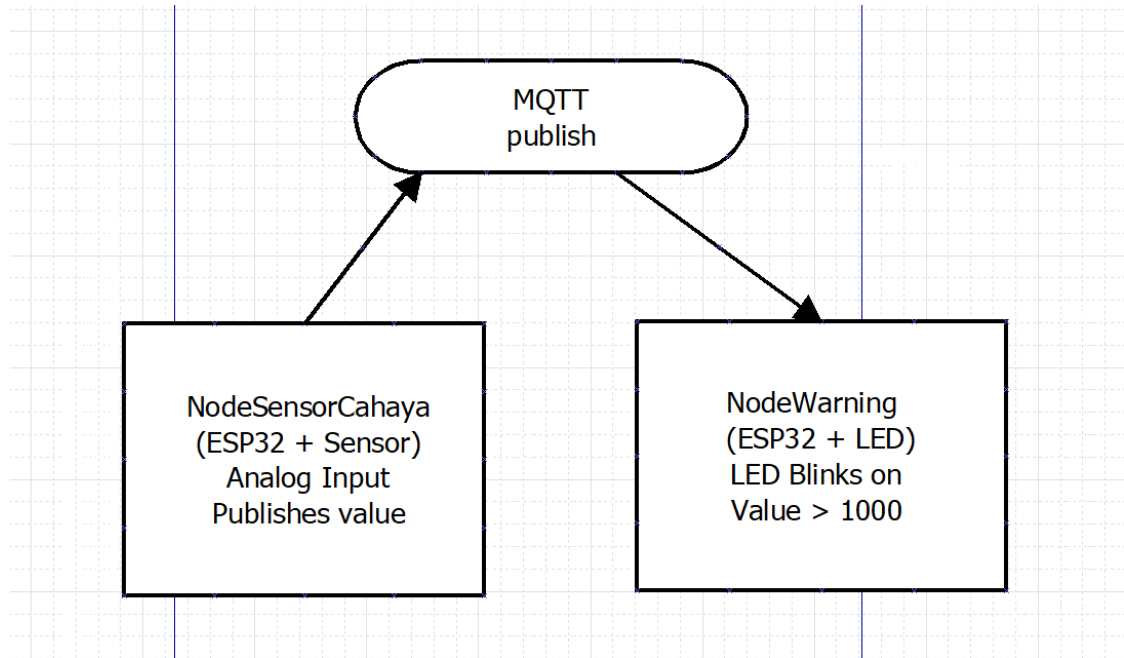
Keluaran NodeWarning



LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

2. Gambarkan diagram sistem dari soal 1.



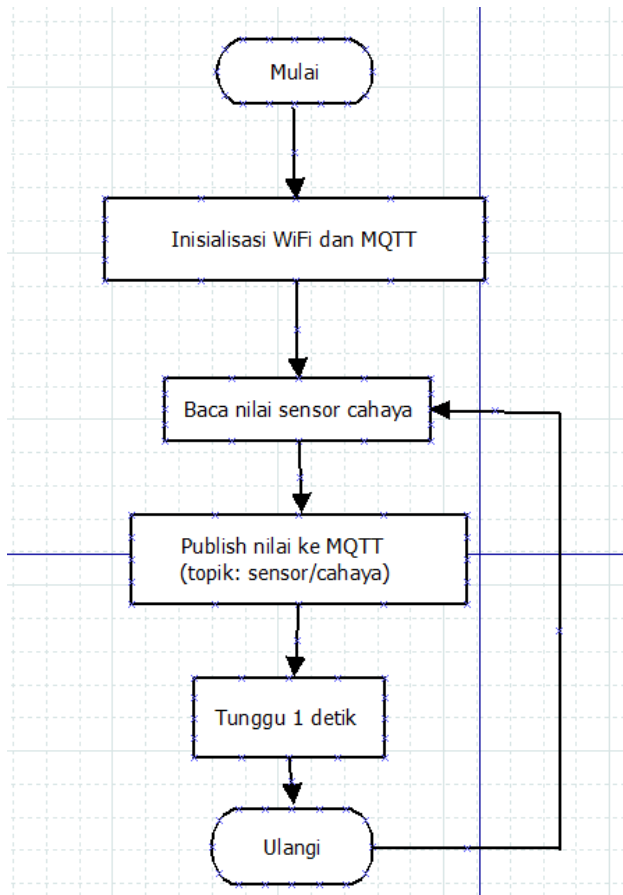
LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

TUGAS

1. Buat diagram alir nodeSensor dan nodeWarning

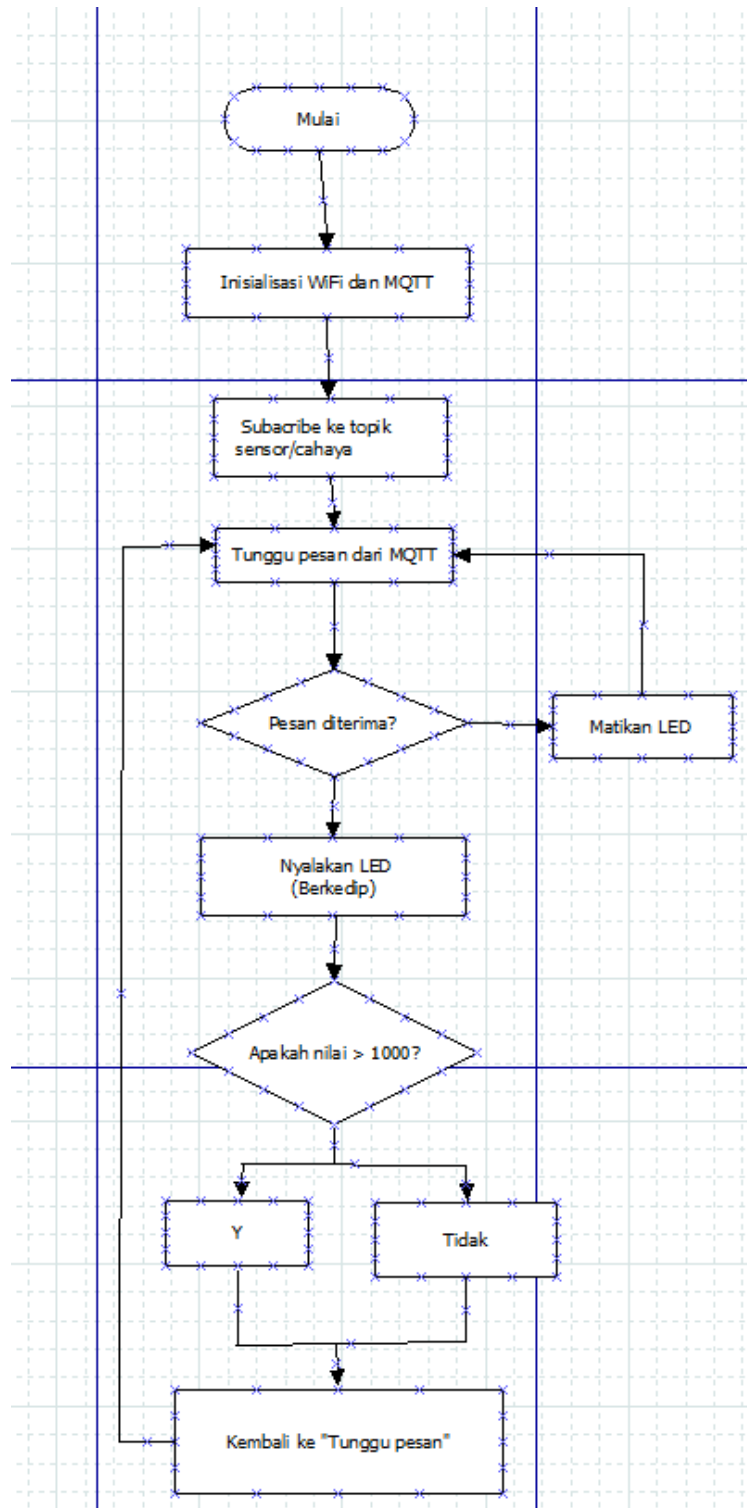
Diagram Alir NodeSensorCahaya



LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)

Diagram Alir NodeWarning



LAPORAN

Laporan praktikum Systemn Internet Of Thinks (Jumat 27 September)