

CONTROLLING DAN MONITORING MENGGUNAKAN ARDUINO SIM900 PROTOKOL MQTT DENGAN MOSQUITTO BROKER



Dosen Pengampu:

Sayyidul Aulia Alamsyah. S.T., M.T.

Nama Anggota Kelompok:

1. Filza Fivebrian Tariszaki (21050874003)
2. Mukhamad Rio Prasetyoaji (21050874030)

Mata Kuliah:

Sistem IoT

TEKNIK ELEKTRO 2021 B

Konsentrasi Elektronika

UNIVERSITAS NEGERI SURABAYA

FAKULTAS TEKNIK

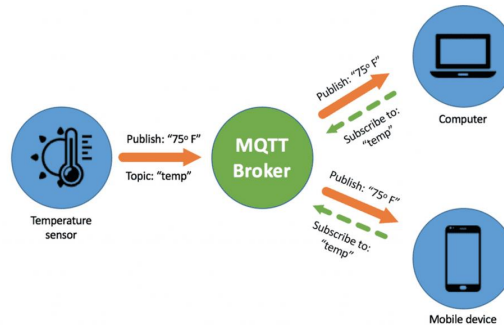
SURABAYA

2023

BAB II

TINJAUAN PUSTAKA

2.1 Protokol MQTT (Message Queuing Telemetry Transport)



Gambar 2.1 Protokol MQTT

Sumber : <http://tinyurl.com/protokol-mqtt>

Salah satu model komunikasi yang digunakan dalam sistem Internet of Things adalah publish/subscribe (PUB/SUB). MQTT menggunakan topik dalam mem-publish ataupun men-subscribe pada pengiriman pesan. MQTT (Message Queuing Telemetry Transport) adalah protokol yang berjalan pada diatas stack TCP/IP dan mempunyai ukuran paket data dengan low overhead yang kecil (minimum 2 bytes) sehingga berefek pada konsumsi catu daya yang juga cukup kecil. Protokol ini adalah jenis protokol data-agnostic yang artinya anda bisa mengirimkan data apapun seperti data binary, text bahkan XML ataupun JSON dan protokol ini memakai model publish/subscribe daripada model client-server .

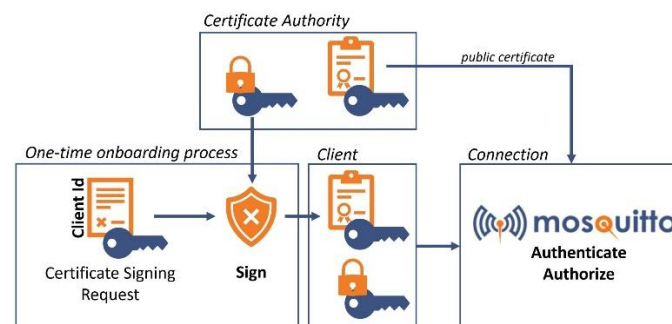
Sistem umum MQTT seperti pada gambar diatas membutuhkan dua komponen perangkat lunak utama yaitu :

- MQTT Client yang nantinya akan di install di device. Untuk Arduino anda bisa memakai pubsubclient, pustaka seperti mqtt.js bisa dipakai pada platform Node.js di Raspberry Pi ataupun laptop.
- MQTT Broker yang berfungsi untuk menangani publish dan subscribe data. Untuk platform Node.js anda bisa memakai broker mosca sedangkan untuk platform yg lain banyak broker tersedia seperti mosquitto, HiveMQ, dll.

MQTT menggunakan topik dalam mem-publish ataupun men-subscribe pada pengiriman pesan. Dikarenakan terhubung pada berbagai perangkat, maka komunikasi yang terjadi antar perangkat dilakukan dengan pengiriman pesan, dimana setiap pesan selalu memiliki topik yang

nantinya digunakan sebagai kata kunci yang berupa string layaknya password. Dalam komunikasi di dalam jaringan pada MQTT digunakan istilah MQTT Publisher dan MQTT Subscriber sebagai perangkat-perangkat yang ingin berkomunikasi satu sama lain, istilah dalam jaringan sebagai client nya. Sedangkan MQTT Broker bertindak sebagai pihak yang mengatur dan meneruskan pesan-pesan yang diterima bahkan bisa mempertahankan pesan pada setiap topik yang dikirimkan oleh MQTT Publisir.

2.2 Mosquitto Broker

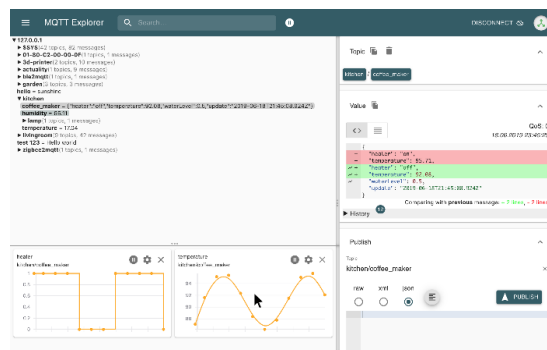


Gambar 2.2 Mosquitto Broker

Sumber : <http://tinyurl.com/mosquitto-broker>

Mosquitto broker adalah salah satu platform yang mendukung protokol MQTT untuk mengoordinasikan pesan antara klien yang berbeda. Mosquitto broker dapat digunakan pada berbagai platform seperti Node.js. Pada arsitektur MQTT-SN, node client terhubung dengan broker melalui gateway menggunakan MQTT. Klien dan broker berkomunikasi dengan menggunakan koneksi MQTT, dimana klien memulai koneksi dengan mengirimkan pesan CONNECT ke broker MQTT. Setelah terhubung, klien dapat memublikasikan pesan, berlangganan pesan tertentu, atau melakukan keduanya. Mosquitto broker juga dapat digunakan untuk membangun protokol MQTT broker pada virtual private server (VPS) sebagai integrasi sistem pemantauan dan peringatan dini bencana.

2.3 MQTT Explorer

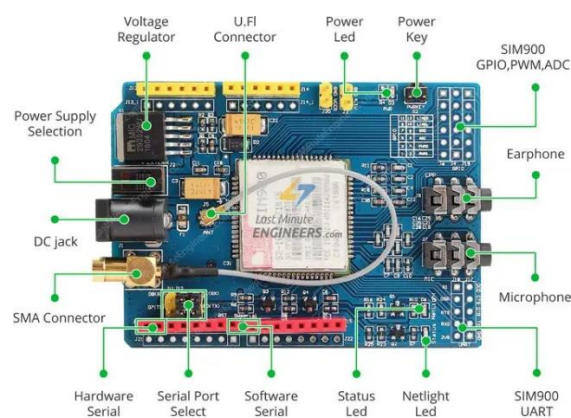


Gambar 2.3 MQTT Explorer

Sumber : <http://tinyurl.com/mqtt-explorer>

MQTT Explorer adalah salah satu klien MQTT yang menyediakan gambaran terstruktur dari topik MQTT dan memudahkan pengguna dalam bekerja dengan perangkat atau layanan pada broker MQTT. MQTT Explorer memiliki fitur visualisasi topik dan aktivitas topik, pencarian/filter topik, penghapusan topik secara rekursif, plot topik numerik, dan tema gelap/cahaya. MQTT Explorer juga dapat digunakan untuk menghubungkan perangkat ke broker MQTT pada Raspberry Pi menggunakan Node RED. MQTT Explorer dapat dijalankan pada platform Node.js dengan menggunakan broker mosca atau broker lain seperti HiveMQ dan Mosquitto. MQTT Explorer juga dapat digunakan untuk terhubung ke layanan MQTT Cloud seperti EMQX Cloud. MQTT Explorer memiliki kemampuan untuk menangani ribuan topik dan ratusan ribu pesan per menit. MQTT Explorer merupakan klien MQTT desktop yang aktif dan sering digunakan oleh pengembang.

2.4 Modul SIM 900



Gambar 2.4 Modul SIM 900

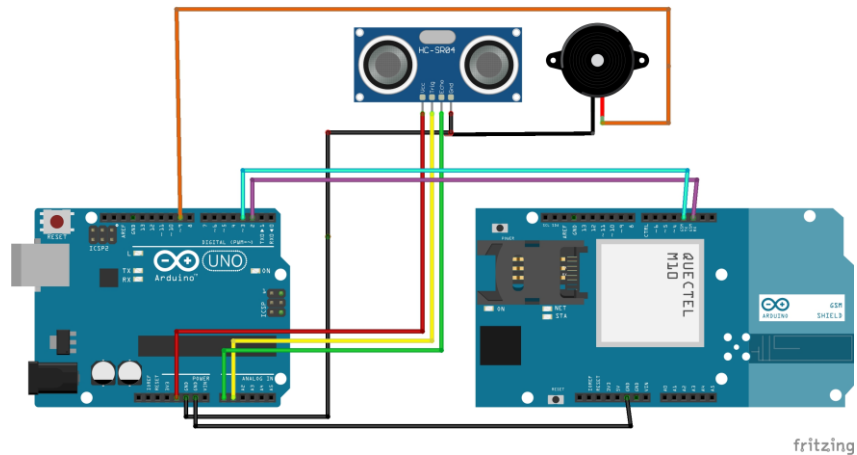
Sumber : <http://tinyurl.com/modul-sim-900>

Modul SIM900 GSM/GPRS adalah bagian yang berfungsi untuk komunikasi antara mikrokontroler Arduino dengan Web Service. Modul komunikasi GSM/GPRS menggunakan core IC SIM900A. Modul ini mendukung komunikasi dual band pada frekuensi 900 / 1800 MHz (GSM900 dan GSM1800) sehingga fleksibel untuk digunakan bersama kartu SIM dari berbagai operator telepon seluler di Indonesia. Operator GSM yang beroperasi di frekuensi dual band 900 MHz dan 1800 MHz sekaligus: Telkomsel, Indosat, dan XL. Operator yang hanya beroperasi pada band 1800 MHz: Axis dan Three.

BAB III

HASIL DAN PEMBAHASAN

3.1 Implementasi



Gambar 3.1 Skema Rangkaian Alat

(sumber: Dokumentasi Pribadi, 2023)

Berikut merupakan table konfigurasi pada sensor dan komponen pada skematik rangkaian alat controlling menggunakan SIM 900, Protokol MQTT, dengan server MQTT Explorer.

Tabel 3.1 Konfigurasi Pin SIM900

Konfigurasi Pin SIM900	
GSM TX	Pin 3
GSM RX	Pin 2
GND	GND

Tabel 3.2 Konfigurasi Pin Sensor Ultrasonik

Konfigurasi Pin Sensor Ultrasonik	
VCC	5V
TRIG	Pin A1
ECHO	Pin A0

GND	GND
-----	-----

Tabel 3.3 Konfigurasi Pin Buzzer

Konfigurasi Pin Buzzer	
VCC	Pin 9
GND	GND

3.2 Cara Kerja

Monitoring dan Controlling menggunakan Arduino + SIM900, Protokol MQTT, dengan MQTT Explorer sebagai client, Mosquitto sebagai broker MQTT, memakai buzzer sebagai aktuatornya, dan Ultrasonik sebagai sensor. MQTT Explorer yang dihubungkan dengan Arduino SIM900 dapat memungkinkan komunikasi pengiriman dan penerima data jarak jauh. Untuk mengontrol buzzer dan memonitor data jarak pembacaan sensor ultrasonik secara jarak jauh melalui MQTT Explorer, Arduino memerlukan koneksi internet agar dapat mem-publish dan men-subscribe data yang kemudian akan dikirimkan ke broker MQTT. Dalam hal ini, SIM900 berguna sebagai *gateway* agar Arduino mendapatkan akses internet dan dapat berkomunikasi dengan broker dan client. Protokol komunikasi yang digunakan adalah MQTT (Message Queueing Telemetry Transport).

3.3 Tutorial

3.3.1 Arduino

1. Berikut merupakan Program Arduino Uno

```
#define TINY_GSM_MODEM_SIM900

#include <SoftwareSerial.h> //Library komunikasi serial
#include <TinyGsmClient.h> //Library untuk modul GPRS
#include <PubSubClient.h> //Library untuk MQTT

#define Buzzer 9 //Input Buzzer Berada di Pin 9 Arduino
#define triggerPin A1 //Trigger pin sensor HC-SR04 Berada di Pin A1 Arduino
#define echoPin A0 //Echo pin sensor HC-SR04 Berada di Pin A0 Arduino
#define RX_PIN 2 //Mendefinisikan pin 2 pada Arduino sebagai pin RX
// untuk komunikasi serial
#define TX_PIN 3 //Mendefinisikan pin 3 pada Arduino sebagai pin TX
// untuk komunikasi serial

SoftwareSerial mySerial(RX_PIN, TX_PIN); //Konfigurasi komunikasi serial

const char apn[] = "internet";
const char gprsUser[] = "wap";
const char gprsPass[] = "wap123";

const char* mqtt_server = "test.mosquitto.org"; //Server dari broker MQTT
yang digunakan

unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;

TinyGsm modem(mySerial);
TinyGsmClient client(modem, 0);
PubSubClient mqttClient(client);

void reconnect() {
    while (!mqttClient.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Create a random client ID
        String clientId = "Client-";
        clientId += String(random(0xffff), HEX);
        // Attempt to connect
        if (mqttClient.connect(clientId.c_str())) {
            Serial.println("connected");
            // Once connected, publish an announcement...
```



```

        mqttClient.publish("Sensor Jarak", "5");
        // ... and resubscribe
        mqttClient.subscribe("Buzzer Command");
    } else {
        Serial.print("failed, rc=");
        Serial.print(mqttClient.state());
        Serial.println(" try again in 5 seconds");
        // Wait 5 seconds before retrying
        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();

    // Switch on the LED if an 0 was received as first character
    if ((char)payload[0] == '0') {
        digitalWrite(Buzzer, LOW);
    } else {
        digitalWrite(Buzzer, HIGH);
    }
}

void setup()
{
    pinMode (Buzzer, OUTPUT);
    pinMode (echoPin, INPUT);
    pinMode (triggerPin, OUTPUT);

    Serial.begin(115200);

    mySerial.begin(9600);
    Serial.println("\nInitializing modem...");
    modem.restart();
    String modemInfo = modem.getModemInfo();
    Serial.print("Modem: ");
    Serial.println(modemInfo);

    Serial.print(F("Connecting to "));
    Serial.print(apn);
    if (!modem.gprsConnect(apn, gprsUser, gprsPass))

```

```

{
    Serial.println(" fail");
    delay(1000);
    return;
}
Serial.println(" OK");

mqttClient.setServer(mqtt_server, 1883);
mqttClient.setCallback(callback);
}

void loop() {

    long duration, jarak;

    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);

    duration = pulseIn(echoPin, HIGH);

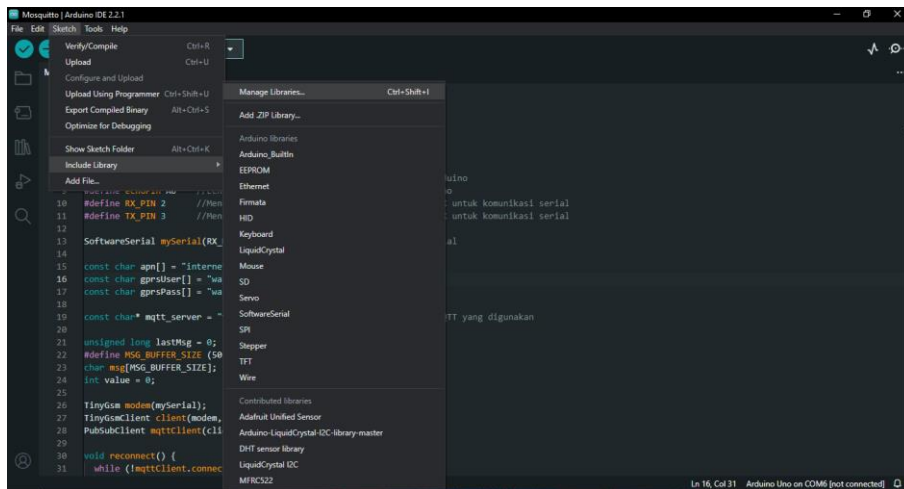
    jarak = (duration/2) / 29.1;

    if (!mqttClient.connected()) {
        reconnect();
    }
    mqttClient.loop();

    unsigned long now = millis();
    if (now - lastMsg > 5000) {
        lastMsg = now;
        snprintf (msg, MSG_BUFFER_SIZE, "Jarak: %ld cm", jarak);
        Serial.print("Publish message: ");
        Serial.println(msg);
        mqttClient.publish("Sensor Jarak", msg);
    }
}

```

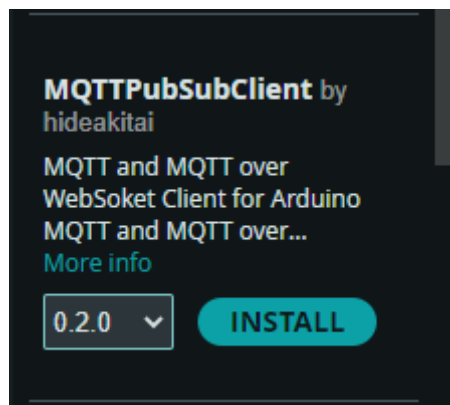
2. Berikut merupakan cara menginstal library SIM900 pada Arduino uno.



Gambar 3.2 Cara Install Library SIM900

(sumber: Dokumentasi Pribadi, 2023)

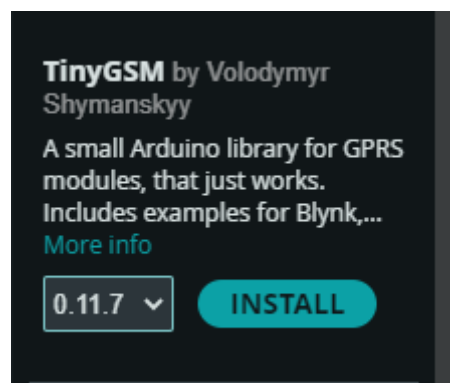
Menekan sketch, lalu tekan Include Library, setelah itu tekan Manager Libraries,



Gambar 3.3 Library MQTT

(sumber: Dokumentasi Pribadi, 2023)

Setelah keluar tampilan Library Manager menginstal MQTTPubSubClient, tekan instal



Gambar 3.4 Library TinyGSM

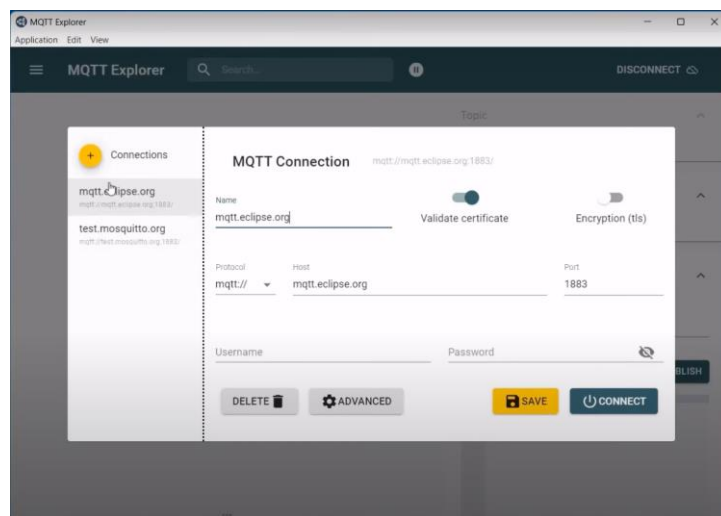
(sumber: Dokumentasi Pribadi, 2023)

Lalu menginstal TinyGSM, lalu tekan instal

3. kemudian menyalin program Arduino pada sketch Arduino.

3.3.2 Konfigurasi MQTT Explorer

1. Instal MQTT Explorer. Setelah itu, buka MQTT Explorer yang telah di instal.

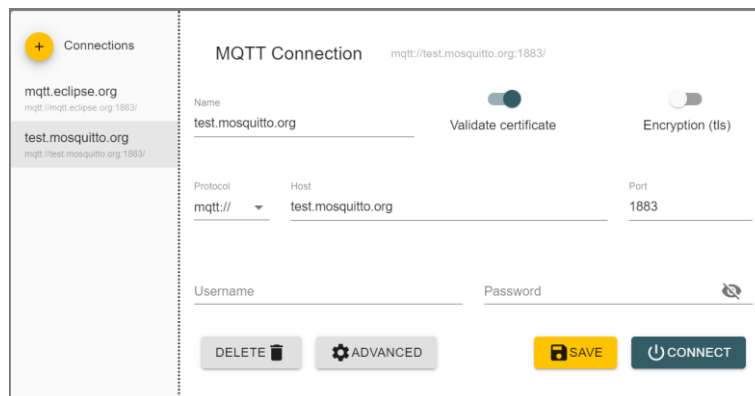


Gambar 3.5 Tampilan Awal MQTT

(sumber: Dokumentasi Pribadi, 2023)

(tampilan pertama kali MQTT Explorer di buka)

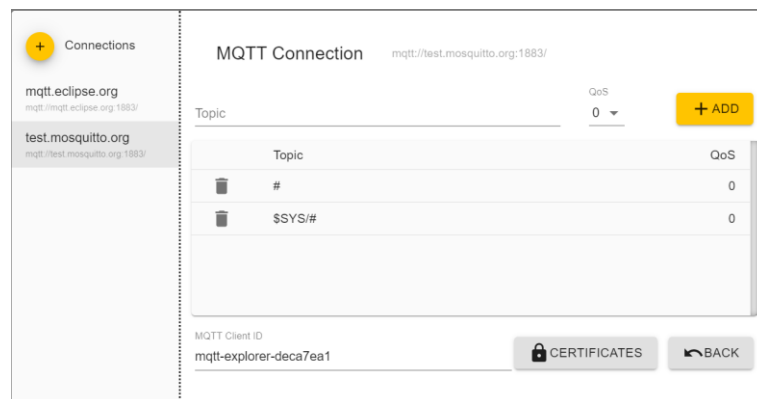
2. Masuk pada koneksi test.mosquitto.org. Lalu tekan kolom Advanced.



Gambar 3.6 Pemilihan Server

(sumber: Dokumentasi Pribadi, 2023)

3. Hapus 2 topik yang sudah ada dengan menekan ikon sampah.

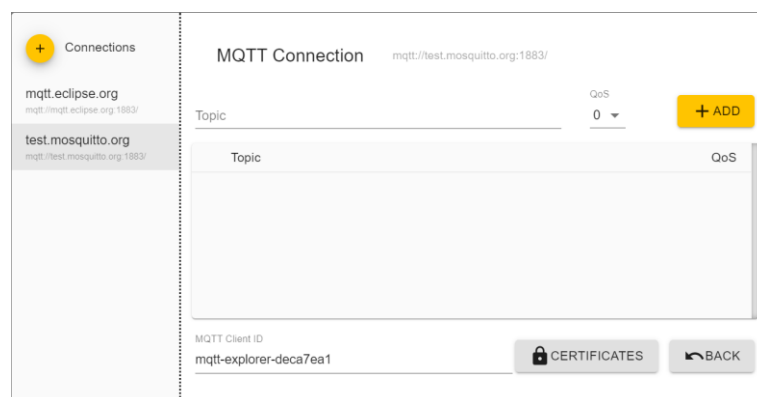


Gambar 3.7 Tampilan Kolom Advanced

(sumber: Dokumentasi Pribadi, 2023)

(Ini adalah tampilan setelah menekan kolom Advanced)

4. Setelah topik kosong, silahkan menambahkan topik dengan cara menulis nama topik di kolom Topic, lalu tekan ADD.



Gambar 3.8 Tampilan Setelah Menghapus Topik

(sumber: Dokumentasi Pribadi, 2023)

5. Pastikan topik yang telah ditambahkan sesuai dengan topik yang tertulis di program.

Contoh:

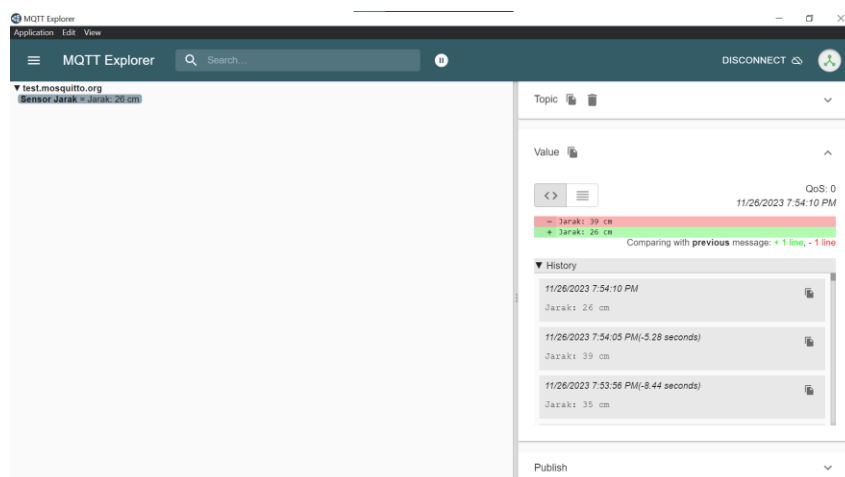


Gambar 3.8 Tampilan Setelah Menambahkan Topik

(sumber: Dokumentasi Pribadi, 2023)

(Dapat dilihat kedua topik yang sudah diinput sudah sesuai dengan program. Topik “Sensor Jarak” digunakan sebagai topik untuk mem-publish hasil pembacaan sensor HC-SR 04 sementara topik “Buzzer Command” digunakan sebagai topik untuk men-subscribe nilai yang nanti kita akan publish untuk mengaktifkan buzzer)

6. Setelah selesai mengkonfigurasi, silahkan tekan BACK untuk kembali ke menu awal. Selanjutnya silahkan koneksikan MQTT Explorer dengan broker dengan cara menekan CONNECT. Setelah berhasil terkoneksi, akan muncul tampilan berikut:

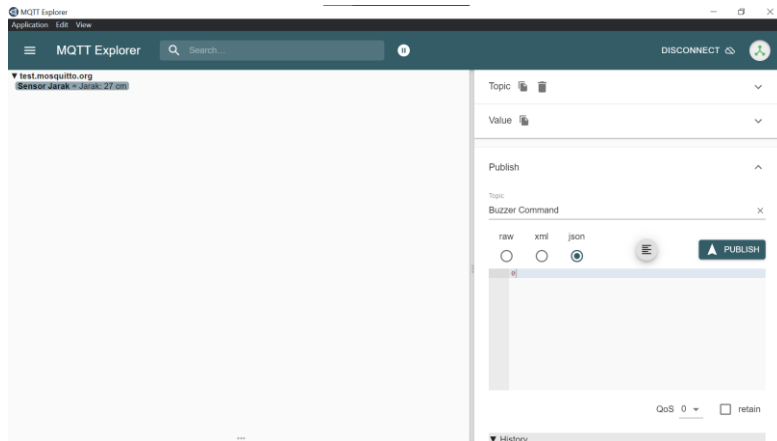


Gambar 3.9 Tampilan Setelah Berhasil Terkoneksi ke Broker

(sumber: Dokumentasi Pribadi, 2023)

(Berikut adalah nilai hasil pembacaan sensor yang telah berhasil diterima MQTT Explorer)

7. Untuk mengontrol Buzzer, kita harus melakukan publish nilai ke topik “Buzzer Command” dengan cara: Klik Publish, pastikan topik sudah sesuai dengan topik untuk mengontrol buzzer, silahkan masukkan nilai yang akan di publish ke kolom berikut. Lalu klik publish.



Gambar 3.10 Tampilan Untuk Mengontrol Buzzer

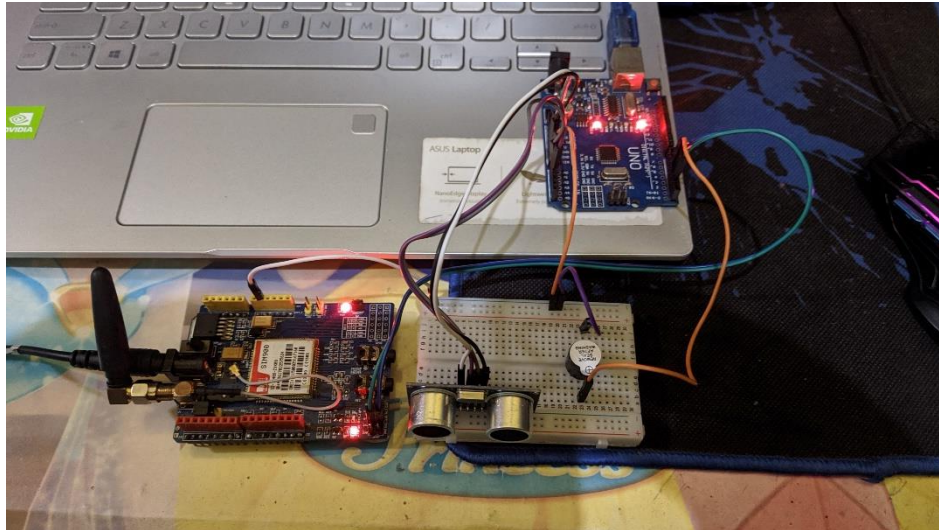
(sumber: Dokumentasi Pribadi, 2023)

(dalam program diatur untuk menghidupkan Buzzer Ketika nilai yang dipublish lebih dari 0 dan buzzer akan mati jika nilai yang dipublish adalah 0)

BAB IV

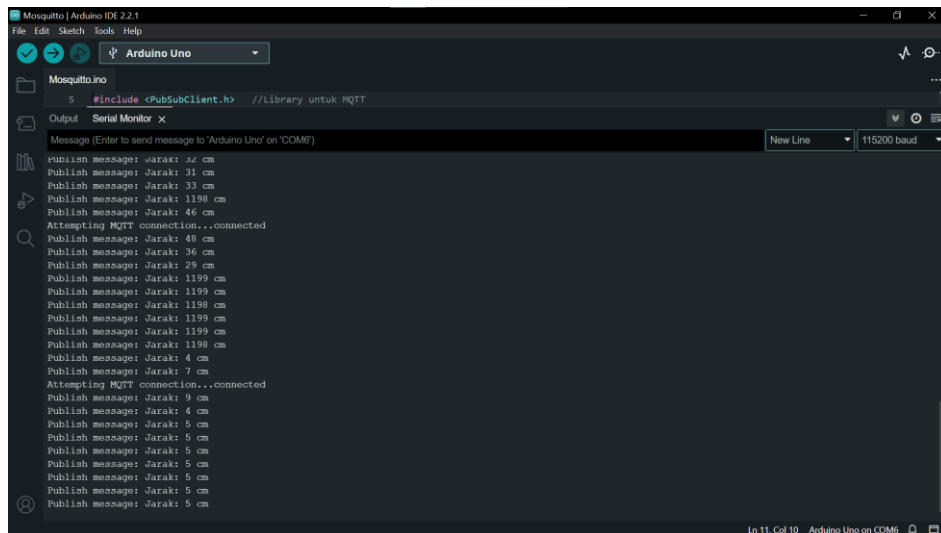
LAMPIRAN

4.1 Dokumentasi



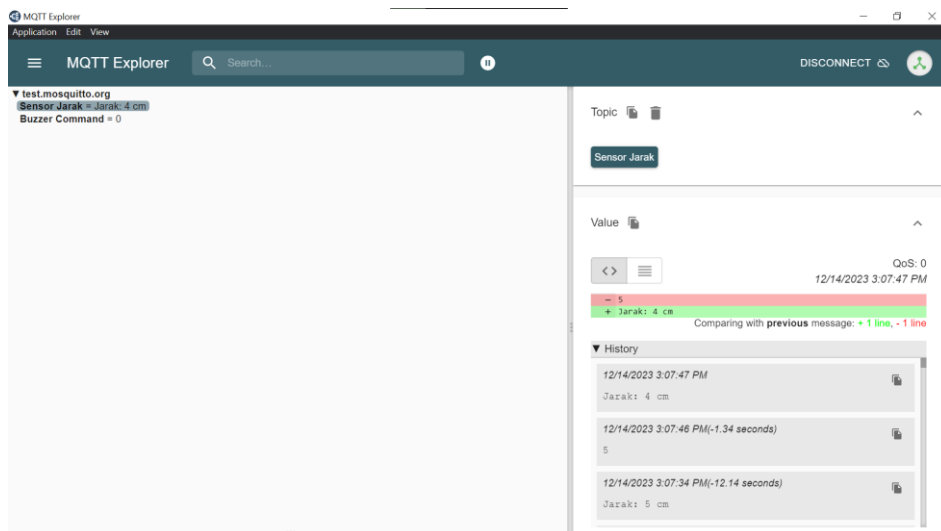
Gambar 4.1 Rangkaian Simulasi

(sumber: Dokumentasi Pribadi, 2023)



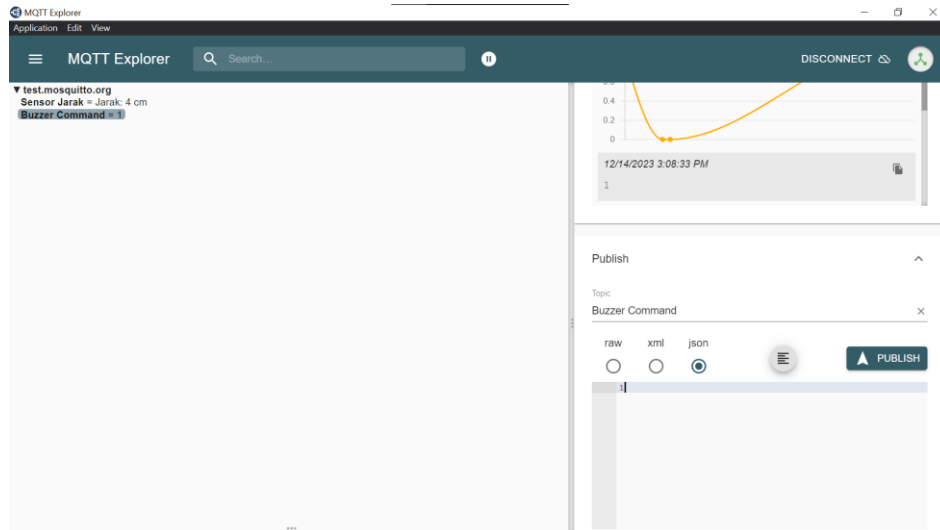
Gambar 4.2 Hasil Publish Pembacaan Nilai Jarak oleh Arduino

(sumber: Dokumentasi Pribadi, 2023)



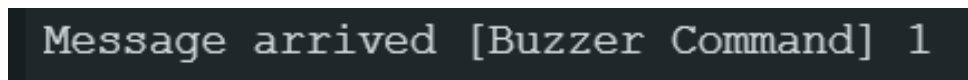
Gambar 4.3 Hasil Subscribe Pembacaan Nilai Jarak oleh MQTT Explorer

(sumber: Dokumentasi Pribadi, 2023)



Gambar 4.4 Hasil Publish Nilai 1 pada MQTT Explorer Untuk Menghidupkan Buzzer

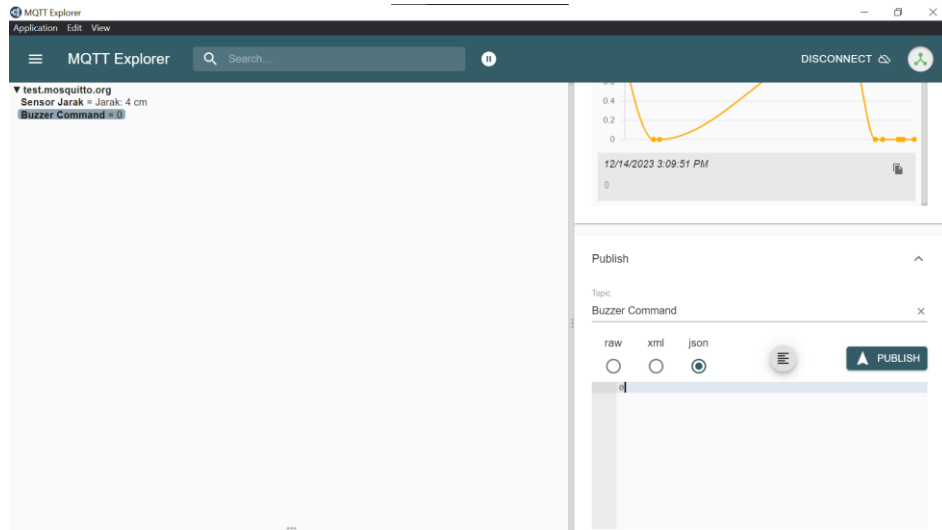
(sumber: Dokumentasi Pribadi, 2023)



Gambar 4.5 Hasil Subscribe Nilai 1 Pada Arduino Untuk Menghidupkan Buzzer

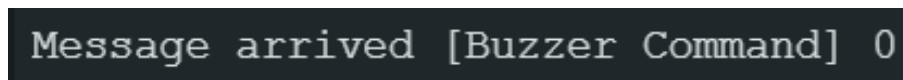
(sumber: Dokumentasi Pribadi, 2023)

Ketika Arduino telah berhasil men-subscribe nilai 1 dari MQTT Explorer maka buzzer akan menyala.



Gambar 4.6 Hasil Publish Nilai 0 pada MQTT Explorer Untuk Mematikan Buzzer

(sumber: Dokumentasi Pribadi, 2023)



Gambar 4.7 Hasil Subscribe Nilai 0 Pada Arduino Untuk Mematikan Buzzer

(sumber: Dokumentasi Pribadi, 2023)

Ketika Arduino telah berhasil men-subscribe nilai 0 dari MQTT Explorer maka buzzer akan mati.

BAB V

DAFTAR PUSTAKA

- Banks, A., Ed, B., Borgendale, K., & Gupta, R. (2019, Maret 7). *MQTT*. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/MQTT>
- Send Receive SMS & Call with SIM900 GSM Shield & Arduino*. (2023). Retrieved from Last Minute ENGINEER: <https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/>
- Yu, S. (2023, Desember 4). *MQTT Explorer Guide: Features, Demos, and Using Tips*. Retrieved from MQ: <http://tinyurl.com/mqtt-explorer-web>
- Innovate Yourself. (2023, April). HOW TO CONTROL LED WITH PUBLISH/SUBSCRIBE MQTT IOT [Video]. YouTube.
<https://youtu.be/deIRY5NxXo4?si=Jem43xBxnfQ3dJtX>
- Ravi Pujar. (2017). Arduino MQTT Tutorial - Coding & Live Demo using SIM900 [Video]. YouTube. <https://youtu.be/5FbafyVvN3E?si=Vvb8ucBsReaCISic>