

SISTEM MONITORING JARAK DAN KONTROL BUZZER BERBASIS IOT MENGGUNAKAN ARDUINO ETHERNET SHIELD DENGAN PROTOKOL MQTT DAN VIRTUINO



Dosen Pengampu:

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

Disusun Oleh:

Kelompok 1 TE 2021 B (Elektronika)

1. Meilinda Mutiara Susilo (21050874008)
2. Ika Sudy Utami (21050874067)
3. Fikri Firmansyah (21050874075)

PROGRAM STUDI S1 TEKNIK ELEKTRO

FAKULTAS TEKNIK

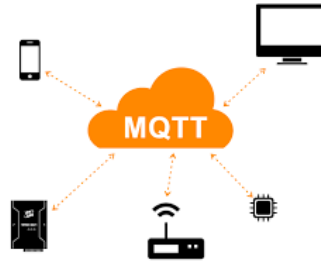
UNIVERSITAS NEGERI SURABAYA

2023

BAB II

TINJAUAN PUSTAKA

A. Message Queuing Telemetry Transport (MQTT)

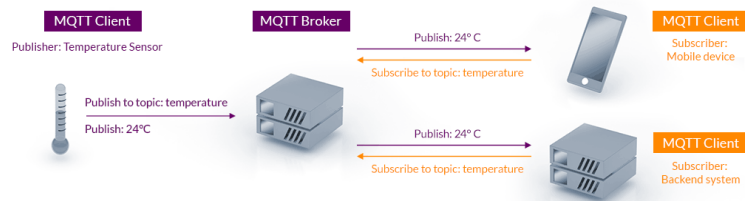


Gambar 1. MQTT

(Sumber: <https://iotstudio.labs.telkomuniversity.ac.id/berkenalan-dengan-mqtt/>)

MQTT adalah protokol perpesanan standar OASIS (*Organization for the Advancement of Structured Information Standards*) untuk *Internet of Things* (IoT). MQTT dirancang sebagai transportasi pesan *Publish/Subscribe* yang sangat ringan dan ideal untuk menghubungkan perangkat jarak jauh dengan *footprint* kode yang kecil dan *bandwidth* jaringan minimal. MQTT saat ini digunakan di berbagai industri, seperti otomotif, manufaktur, telekomunikasi, minyak dan gas, dll. MQTT ringan dan efisien karena klien MQTT berukuran sangat kecil dan hanya memerlukan sumber daya minimal sehingga dapat digunakan pada mikrontroler kecil. Header pesan MQTT juga berukuran kecil untuk mengoptimalkan *bandwidth* jaringan. MQTT mendukung komunikasi dua arah, sehingga memungkinkan pengiriman pesan antara perangkat ke *cloud* dan *cloud* ke perangkat. Hal ini memudahkan penyebaran pesan ke berbagai grup atau kelompok. MQTT dapat berkembang untuk terhubung dengan banyak perangkat IoT [1].

Keandalan pengiriman pesan sangat penting bagi banyak kasus penggunaan IoT. Oleh karena itu MQTT memiliki 3 QoS (*Quality of Service*) yaitu 0 – paling banyak satu kali, 1 – setidaknya satu kali, 2 – tepat satu kali. Banyak perangkat IoT terhubung melalui jaringan seluler yang tidak dapat diandalkan. Dukungan MQTT untuk sesi persisten mengurangi waktu untuk menghubungkan kembali klien dengan broker. MQTT memudahkan enkripsi pesan menggunakan TLS dan mengautentikasi klien menggunakan protokol autentikasi modern, seperti OAuth [1]. Arsitektur dari *Publish/Subscribe* MQTT dapat dilihat pada gambar 2. Publisher mem-publish nilai pembacaan sensor ke topik tertentu. Nilai tersebut disalurkan ke perangkat mobile dan sistem backend yang merupakan subscriber melalui broker MQTT.



Gambar 2. *Arsitektur Publish/Subscribe MQTT*
(Sumber: <https://mqtt.org/>)

B. Virtuino

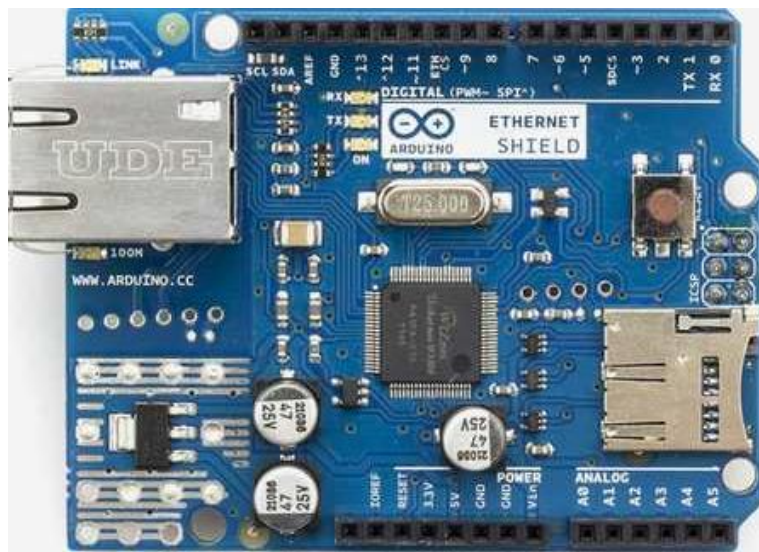


Gambar 3. *Logo Virtuino*

(Sumber: https://play.google.com/store/apps/details?id=com.virtuino.virtuino_iot&hl=in&gl=US)

Virtuino merupakan platform HMI (*Human Machine Interface*) untuk platform IoT, Arduino, ESP8266, ESP32 dan board lain yang sejenis, broker MQTT, serta modul PLC dan Modbus [2]. Virtuino dapat membantu dalam memvisualisasikan atau mengontrol berbagai sistem automasi. Aplikasi ini mendukung koneksi MQTT, MQTT5, MODBUS, Soket Web & HTTP [3]. Selain itu, aplikasi ini digunakan untuk membuat layar menakjubkan untuk memvisualisasikan atau mengontrol perangkat Arduino atau platform IoT [4].

C. Arduino Ethernet Shield



Gambar 4. *Arduino Ethernet Shield*

(Sumber: <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>)

Arduino Ethernet Shield menghubungkan Arduino ke internet hanya dalam hitungan menit. Untuk menggunakan Arduino Ethernet Shield cukup dengan mencolokkan modul ini ke board Arduino. Kemudian, sambungkan ke jaringan dengan menggunakan kabel RJ45. Arduino Ethernet Shield memungkinkan board Arduino terhubung ke internet. Modul ini didasarkan pada chip Wiznet W5100 ethernet. Wiznet W5100 menyediakan tumpukan jaringan (IP) yang mampu melakukan TCP dan UDP sehingga mendukung hingga empat koneksi socket secara bersamaan. Untuk menulis sketsa yang terhubung ke internet menggunakan shield diperlukan library ethernet. Ethernet shield terhubung ke board Arduino menggunakan header bungkus kawat panjang yang memanjang melalui shield sehingga menjaga tata letak pin tetap utuh dan memungkinkan shield lain ditumpuk di atasnya [5].

D. Sensor Ultrasonik



Gambar 5. Sensor Ultrasonik

(Sumber: <https://www.arduinoindonesia.id/2022/10/pengertian-dan-cara-kerja-sensor-ultrasonik-HC-SR04.html>)

Sensor ultrasonik, juga dikenal sebagai sensor jarak, adalah perangkat elektronik yang digunakan untuk mengukur jarak atau jarak relatif antara sensor dan objek di depannya. Sensor ini menggunakan gelombang suara ultrasonik untuk mengukur jarak dan sangat umum digunakan dalam berbagai aplikasi, termasuk dalam proyek-projek Internet of Things (IoT), otomatisasi, kendaraan otonom, dan banyak lagi. Cara kerja dari sensor ultrasonik yaitu dengan mengirimkan pulsa gelombang suara ultrasonik ke objek yang ingin diukur jaraknya. Gelombang suara ini kemudian dipantulkan kembali dari objek dan diterima oleh sensor. Waktu yang dibutuhkan oleh gelombang suara untuk pergi dari sensor ke objek dan kembali digunakan untuk menghitung jarak antara sensor dan objek [6].

E. Buzzer



Gambar 6. Buzzer

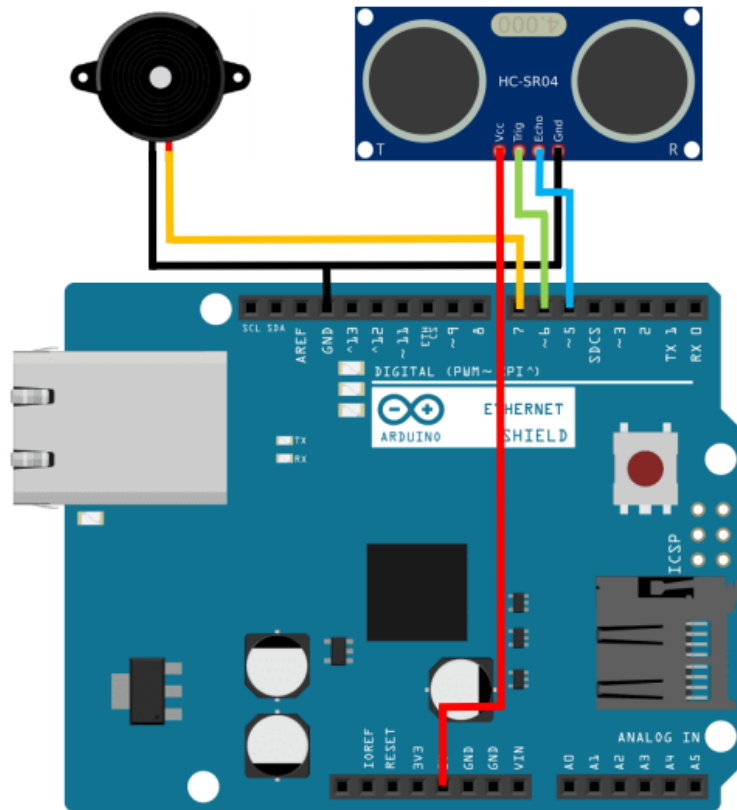
(Sumber: <http://syarifahmuthiaputri.blog.uma.ac.id/2021/07/05/buzzer-and-electro/>)

Buzzer merupakan komponen elektronika yang dapat menghasilkan suara dari membran yang terdapat kumparan. Buzzer mengubah sinyal listrik menjadi sinyal suara. Dibandingkan dengan speaker yang bekerja pada tegangan AC, buzzer bekerja pada tegangan DC. Spesifikasi buzzer bermacam-macam berdasarkan tegangan kerjanya, bervariasi mulai dari 5 Volt, 9 Volt, 12 Volt, 24 Volt, dan lain-lain [7]. Buzzer terdiri dari sebuah kumparan yang dialiri dengan arus listrik sehingga menghasilkan elektromagnetik. Kumparan didorong masuk atau keluar bergantung pada aliran arus listrik dan arah dari kutub-kutub magnet. Setiap gerakan bolak-balik dari kumparan yang menempel pada diafragma menyebabkan udara bergetar dan menghasilkan suara [8].

BAB III

HASIL DAN PEMBAHASAN

A. Implementasi



Gambar 7. Skematik Rangkaian Sistem
(Sumber: Dokumentasi Pribadi)

Konfigurasi pin pada sensor dan aktuator dengan modul ethernet ditunjukkan pada tabel berikut.

Tabel 1. Konfigurasi Pin Buzzer

Ethernet	Buzzer
7	+
GND	-

Tabel 2. Konfigurasi Pin Sensor Ultrasonik

Ethernet	Ultrasonik
5V	Vcc
GND	Gnd
5	Echo
6	Trig

B. Cara Kerja

Sistem monitoring jarak dan kontrol buzzer berbasis IoT yang dibuat menggunakan modul ethernet shield chip W5100 untuk menghubungkan board Arduino UNO ke internet. Protokol yang digunakan adalah MQTT dan aplikasi yang digunakan untuk monitoring dan kontrol adalah Virtuino IoT. Pada protokol MQTT menggunakan istilah publish dan subscribe dimana keduanya merupakan klien MQTT. Selain itu, juga terdapat istilah topik. Topik harus memiliki nama yang unik atau spesifik jika menggunakan broker publik karena broker hanya berfungsi sebagai penyalur. Sehingga jika ada kesamaan nama topik, maka subscriber bisa saja menerima data dari klien lain yang tidak diinginkan.

Pada sistem monitoring, sensor ultrasonik berperan sebagai publisher dan aplikasi virtuino berperan sebagai subscriber. Sensor ultrasonik mengambil data dari lingkungan kemudian data tersebut akan diproses oleh Arduino sehingga didapatkan nilai jarak. Nilai jarak yang didapatkan selanjutnya akan dipublish pada topik yang telah ditentukan. Broker akan menyalurkan nilai tersebut ke Virtuino yang telah melakukan subscribe pada topik yang sama. Setelah itu, pengguna dapat memonitoring nilai jarak melalui Virtuino.

Sementara itu pada sistem kontrol, Arduino berperan sebagai subscriber yang nantinya akan memberikan perintah pada buzzer berdasarkan pesan yang masuk dari publisher dan Virtuino berperan sebagai publisher. Virtuino akan mempublish nilai 1 atau 0 pada topik yang telah ditentukan. Topik yang digunakan berbeda dengan topik yang digunakan untuk mempublish nilai jarak. Nilai 1 atau 0 tersebut kemudian disalurkan broker ke Arduino yang telah melakukan subscribe pada topik yang sama. Ketika Arduino menerima nilai 1, maka Arduino akan memberi perintah pada buzzer untuk menyala dan ketika Arduino menerima nilai 0 maka Arduino akan memberi perintah untuk menonaktifkan buzzer.

C. Tutorial

1. Arduino IDE

- a. Gunakan listing program berikut untuk disalin pada sketch Arduino.

```
#include <SPI.h>
#include <Ethernet.h>      // Library ethernet w5100
#include <PubSubClient.h>  // Library MQTT

const char* mqtt_server = "test.mosquitto.org"; // URL broker mqtt yang
digunakan
const int mqtt_port = 1883; // port mqtt publik, tidak memerlukan nama
dan username
```

```

EthernetClient ethClient;
PubSubClient client(ethClient);

byte mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };

#define buzzer 7
#define echoPin 5
#define trigPin 6

long duration;
int distance;
unsigned long lastMsg = 0;

int sensor1;    // nilai sensor ultrasonik
int command1 = 0;    // buzzer

const char* sensor1_topic= "IOT_Ultrasonik";
const char* command1_topic= "IOT_Buzzer";

void setup() {
    Serial.begin(9600);
    Serial.println("Attempting to connect...");
    while (!Serial) delay(1);
    Ethernet.begin(mac);
    pinMode(buzzer, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    //--- inisialisasi mqtt client
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(callback);
    randomSeed(micros());
    String clientId = "EthernetClient-";    // Membuat ID client random
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
        Serial.println("connected");
        client.subscribe(command1_topic);    // subscribe topik
    }
}

```



```

else {
    Serial.print("\nfailed, rc=");
    Serial.println(client.state());
}
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.034 / 2;
    delay(500);

    client.loop();
    unsigned long now = millis();
    if (now - lastMsg > 500) { //mem-publish nilai sensor setiap 0.5 detik
        lastMsg = now;
        sensor1 = distance;
        publishMessage(sensor1_topic, String(sensor1), true); // publish
nilai sensor1 ke broker
    }
}

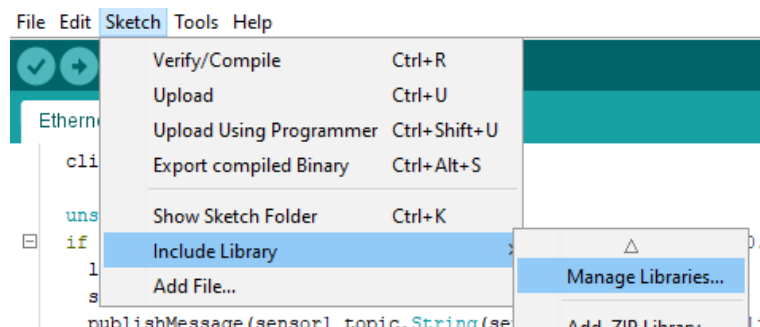
// --- fungsi yang dipanggil setiap ada pesan dari broker
void callback(char* topic, byte* payload, unsigned int length) {
    String incommingMessage = "";
    for (int i = 0; i < length; i++) incommingMessage+=(char)payload[i];
    Serial.println("Message arrived ["+String(topic)+"]"+
incommingMessage);

    //--- cek pesan yang masuk
    if( strcmp(topic,command1_topic) == 0){
        if (incommingMessage.equals("1")) digitalWrite(buzzer, HIGH);
        else digitalWrite(buzzer, LOW);
    }
}

```

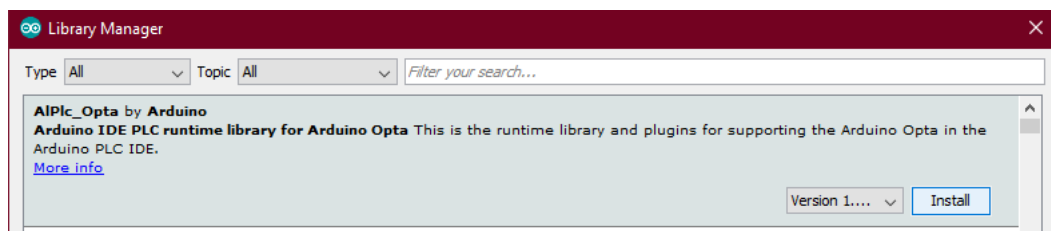
```
//---- mempublish sebagai string
void publishMessage(const char* topic, String payload , boolean
retained){
    if (client.publish(topic, payload.c_str(), true))
        Serial.println("Message publised ["+String(topic)+"]: "+payload);
}
```

- b. Install library yang dibutuhkan.
- c. Pilih menu “Sketch” → “Include Library” → “Manage Libraries...”



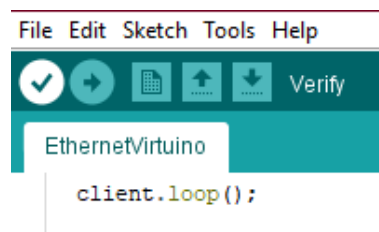
Gambar 8. Tampilan Menu Sketch Arduino IDE
(Sumber: Dokumentasi Pribadi)

- d. Setelah muncul tampilan Library Manager, cari library yang ingin diinstall kemudian klik “Install”.



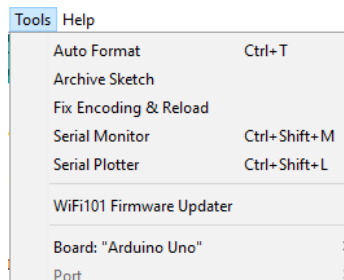
Gambar 9. Tampilan Library Manager
(Sumber: Dokumentasi Pribadi)

- e. Setelah terinstall, close menu library manager dan tekan menu “Verify”.



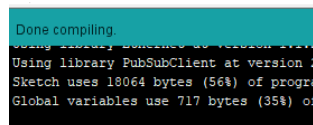
Gambar 10. Tombol Verify
(Sumber: Dokumentasi Pribadi)

- f. Pilih menu “Tools” untuk memastikan bahwa board yang dipilih sudah benar.



Gambar 11. Tampilan Menu Tools Arduino IDE
(Sumber: Dokumentasi Pribadi)

- g. Pastikan bahwa program sudah tidak ada yang error dan program bisa diupload ke board. Jika masih ada error maka periksa kembali bagian mana yang error.



Gambar 12. Tampilan setelah Tidak Terdapat Error pada Program
(Sumber: Dokumentasi Pribadi)

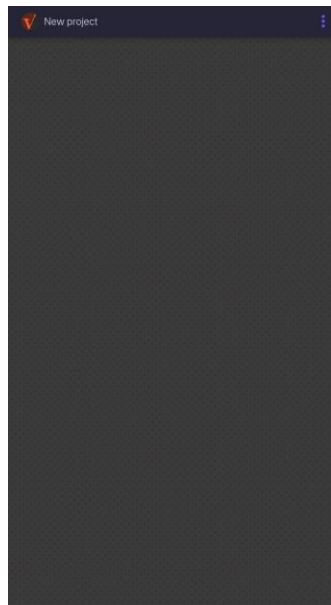
2. Virtuino IoT

- a. Install aplikasi Virtuino IoT. Bisa melalui play store, app store, atau windows



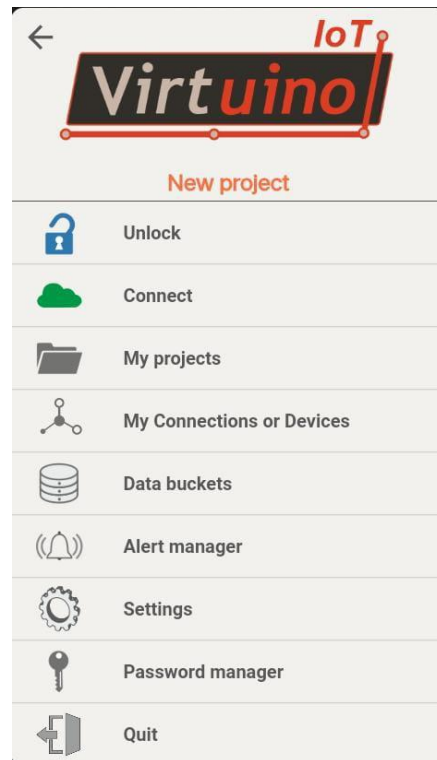
Gambar 13. Aplikasi Virtuino IoT di Play Store
(Sumber: Dokumentasi Pribadi)

- b. Buka aplikasi, maka akan muncul tampilan awal halaman kosong seperti berikut.



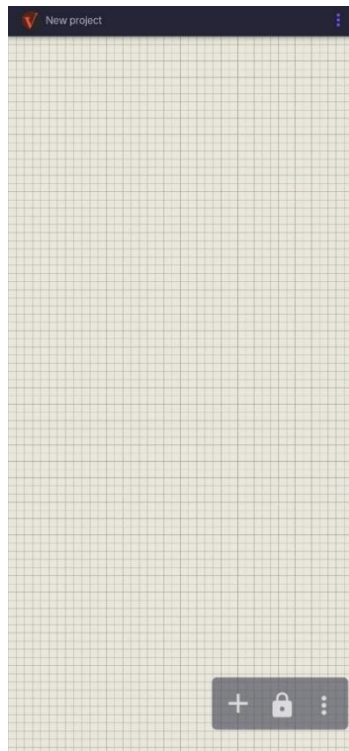
Gambar 14. Tampilan Awal Aplikasi Virtuino IoT
(Sumber: Dokumentasi Pribadi)

- c. Klik titik tiga di kanan atas, maka akan muncul menu seperti berikut. Pilih “Unlock”.



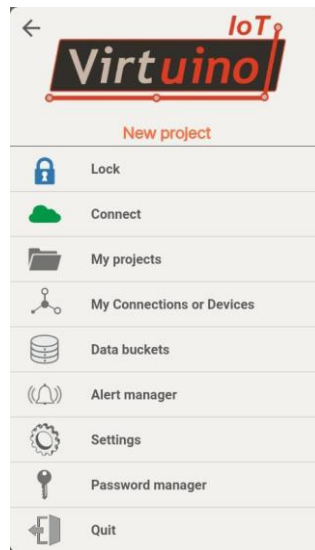
Gambar 15. Tampilan Pilihan Menu Virtuino IoT
(Sumber: Dokumentasi Pribadi)

- d. Maka tampilan dashboard akan berubah menjadi seperti berikut.



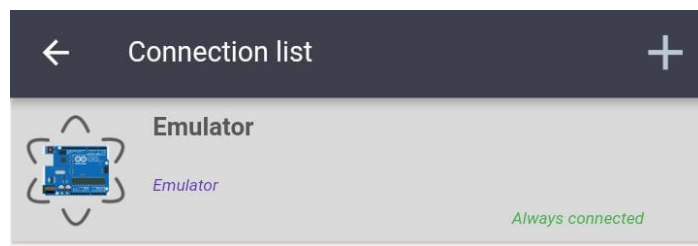
Gambar 16. Tampilan Dashboard Aplikasi setelah di-Unlock
(Sumber: Dokumentasi Pribadi)

- e. Klik titik tiga di kanan atas, maka akan muncul menu seperti berikut. Pilih “My Connections or Devices”.



Gambar 17. Tampilan Pilihan Menu Virtuino IoT
(Sumber: Dokumentasi Pribadi)

- f. Setelah muncul tampilan seperti berikut, klik tanda “+”.



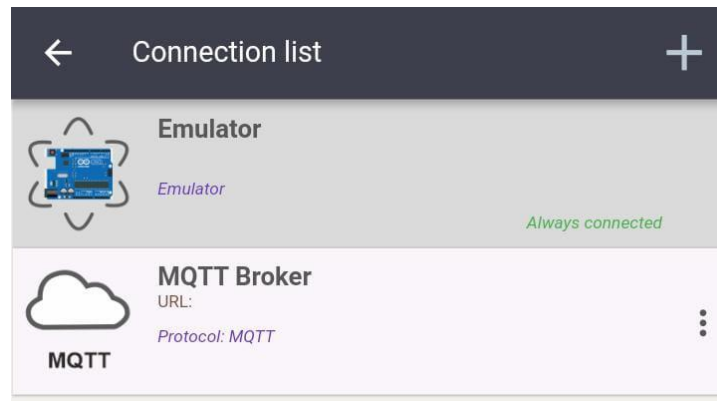
Gambar 18. Tampilan Menu Connection List
(Sumber: Dokumentasi Pribadi)

- g. Jika telah muncul tampilan seperti berikut, pilih “MQTT Connection” untuk menggunakan protokol MQTT.



Gambar 19. Tampilan Pilihan Menu Koneksi
(Sumber: Dokumentasi Pribadi)

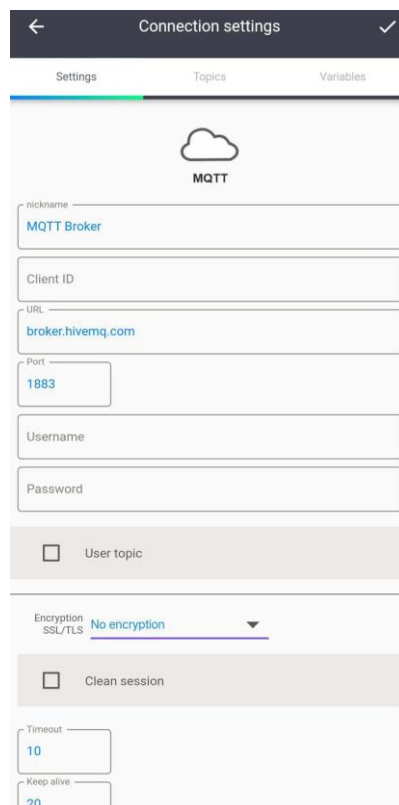
- h. Setelah menambahkan koneksi MQTT akan muncul tampilan seperti berikut. Klik koneksi tersebut untuk melakukan setting koneksi.



Gambar 20. Tampilan Menu Connection List
(Sumber: Dokumentasi Pribadi)

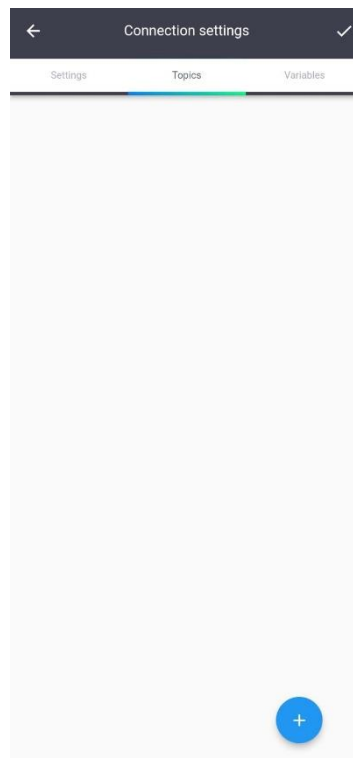
- i. Setelah masuk ke tampilan Connection Settings, pada menu “Settings” ganti URL dan Port sesuai program yang ada, yaitu pada baris

```
const char* mqtt_server = "test.mosquitto.org"; // URL broker mqtt yang
digunakan
const int mqtt_port = 1883; // port mqtt publik, tidak memerlukan nama
dan username
```



Gambar 21. Tampilan Menu Settings pada Connection Settings
(Sumber: Dokumentasi Pribadi)

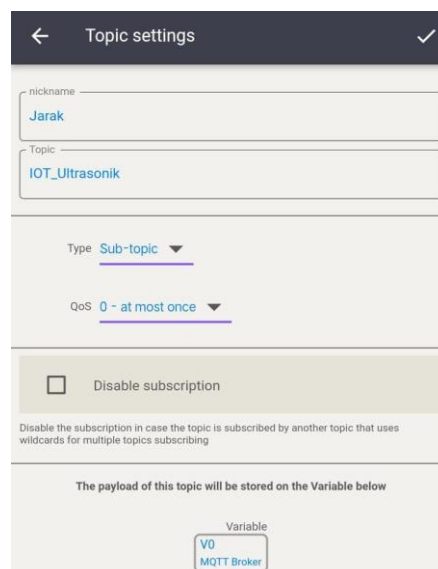
- j. Setelah itu, ganti ke menu “Topics” dan klik tanda “+” untuk menambahkan topik.



Gambar 22. Tampilan Menu Topics
(Sumber: Dokumentasi Pribadi)

- k. Setting topik sesuai dengan program yang sudah ada yaitu pada baris

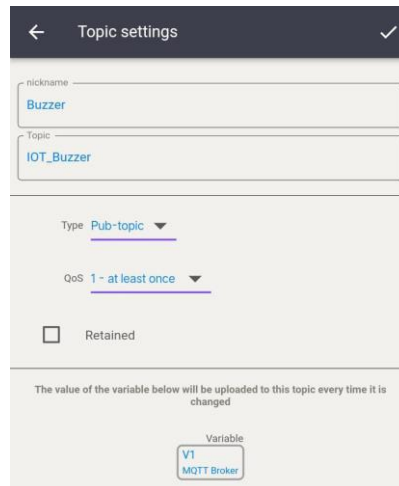
```
const char* sensor1_topic= "IOT_Ultrasonik";  
const char* command1_topic= "IOT_Buzzer";
```



Gambar 23. Tampilan Menu Topic Settings untuk Sensor Jarak
(Sumber: Dokumentasi Pribadi)

Berikan nickname apapun, namun topic disesuaikan dengan program. Karena topik yang dimasukkan adalah topik sensor ultrasonik, maka type diatur sebagai Sub-topic dan menggunakan QoS 0. Untuk Variable bisa memilih variabel manapun. Setelah selesai klik tanda “centang”.

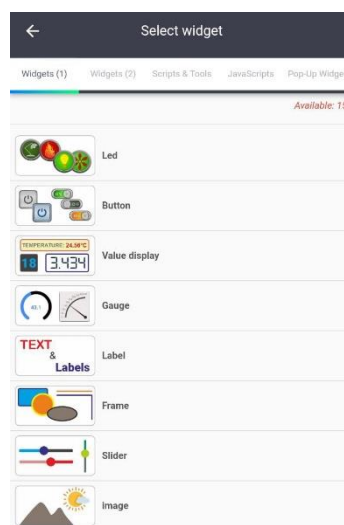
- l. Selanjutnya menambahkan topik untuk kontrol buzzer.



Gambar 24. Tampilan Menu Topic Settings untuk Buzzer
(Sumber: Dokumentasi Pribadi)

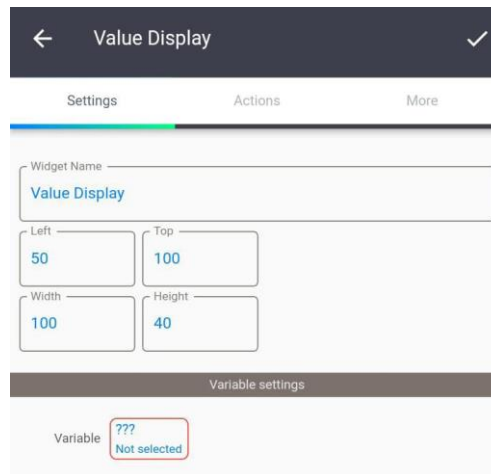
Karena buzzer akan digunakan sebagai output dari kontrol, maka untuk type pada topik buzzer diatur sebagai Pub dan menggunakan QoS 1 untuk mencegah hilangnya message. Untuk Variable bisa memilih variabel manapun. Setelah selesai klik tanda “centang”.

- m. Setelah selesai mengatur menu “Settings” dan “Topics”, klik tanda centang. Kemudian kembali ke dashboard untuk menambahkan widget dengan meng-klik tanda “+” dan akan muncul tampilan Select widget.



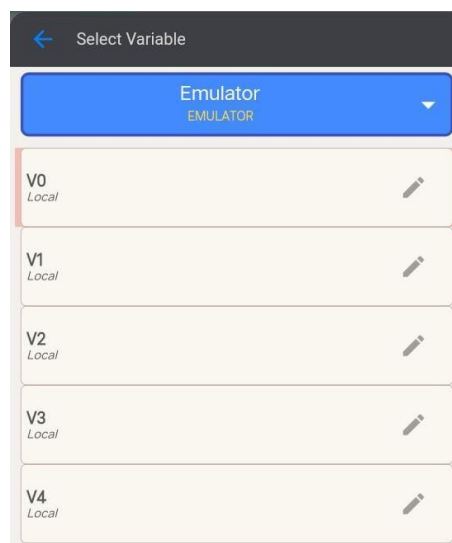
Gambar 25. Tampilan Select Widget
(Sumber: Dokumentasi Pribadi)

- n. Untuk monitoring jarak, pilih widget Value display. Maka akan muncul tampilan berikut.



Gambar 26. Tampilan Menu Settings pada Value Display
(Sumber: Dokumentasi Pribadi)

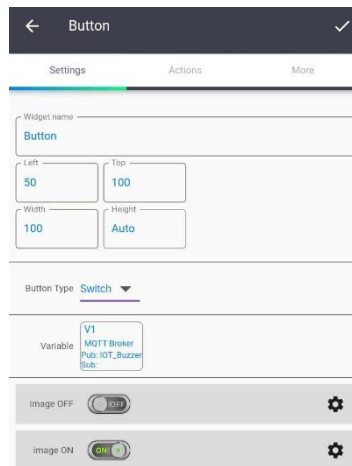
- o. Pilih pada bagian Variable hingga muncul tampilan seperti berikut. Pada bagian emulator, ganti ke MQTT Broker dan pilih variabel yang digunakan untuk topik sensor ultrasonik. Jika ingin menambahkan kata-kata, bisa menuliskan di bagian Suffix dan Prefix. Setelah selesai klik tanda “centang”.



Gambar 27. Tampilan untuk Memilih Variabel
(Sumber: Dokumentasi Pribadi)

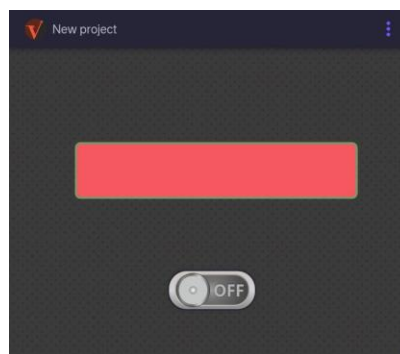
Atur tata letak widget pada dashboard dengan cara menekan lama widget.

- p. Ulangi langkah sebelumnya untuk menambahkan widget kontrol On Off Buzzer. Untuk kontrol buzzer bisa menggunakan widget Button. Cara mengatur widget juga sama seperti sebelumnya yaitu pilih pada bagian Variable kemudian pada bagian emulator, ganti ke MQTT Broker dan pilih variabel yang digunakan untuk topik buzzer.



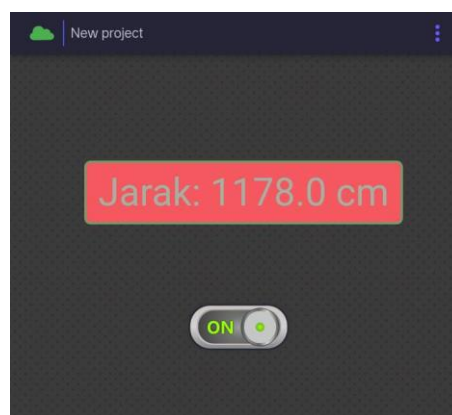
Gambar 28. Tampilan Menu Settings untuk Button
(Sumber: Dokumentasi Pribadi)

- q. Setelah mengatur semua widget yang dibutuhkan. Kembali ke dashboard, klik titik tiga di kanan atas kemudian pilih “Lock”. Maka tampilan dashboard akan menjadi seperti berikut.



Gambar 29. Tampilan Dashboard setelah ditambah Widget
(Sumber: Dokumentasi Pribadi)

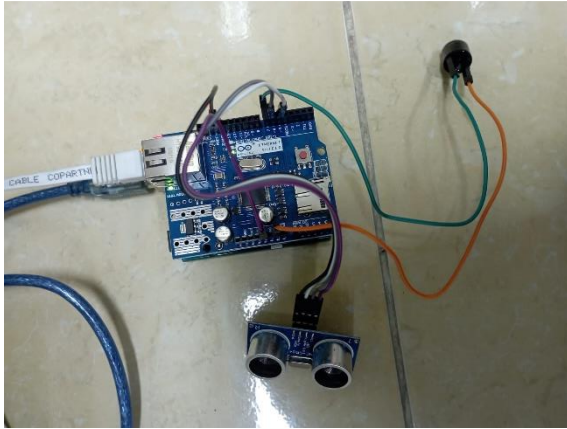
- r. Ulangi lagi klik titik tiga kemudian pilih “Connect”. Maka Virtuino IoT sudah siap digunakan untuk monitoring jarak dan kontrol buzzer.



Gambar 30. Tampilan Dashboard ketika sudah Connect
(Sumber: Dokumentasi Pribadi)

BAB IV

LAMPIRAN



DAFTAR PUSTAKA

- [1] "MQTT: The Standard for IoT Messaging," MQTT, 2022. [Online]. Available: <https://mqtt.org/>. [Accessed 23 December 2023].
- [2] "Virtuino," Virtuino, 2023. [Online]. Available: <https://virtuino.com/>. [Accessed 23 December 2023].
- [3] "Virtuino IoT - Aplikasi di Google Play," Google Play, 2 November 2023. [Online]. Available: https://play.google.com/store/apps/details?id=com.virtuino.virtuino_iot&hl=in&gl=US. [Accessed 24 December 2023].
- [4] "Virtuino IoT on the App Store," Apple, 12 November 2023. [Online]. Available: <https://apps.apple.com/id/app/virtuino-iot/id1624114498>. [Accessed 24 December 2023].
- [5] "Arduino Ethernet Shield | Arduino Documentation," Arduino, 13 December 2023. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>. [Accessed 24 December 2023].
- [6] E. A. Prastyo, "Pengertian dan Cara Kerja Sensor Ultrasonik HC-SR04," Arduino Indonesia, 31 October 2022. [Online]. Available: <https://www.arduinoindonesia.id/2022/10/pengertian-dan-cara-kerja-sensor-ultrasonik-HC-SR04.html>. [Accessed 24 December 2023].
- [7] S. Irjayanto and A. N. N. Chamim, "Prototipe Kotak Peningat Minum Obat (Prototype of Medication Reminder Box)," *Jurnal Ilmiah Semesta Teknik*, vol. XVIII, no. 2, pp. 182-189, 2015.
- [8] I. Inayatillah, R. Munandi and A. I. Irawan, "Automatic Cat Feeding Monitoring System with The Arduino Mega 2560-Using Hc-Sr04 Sensor Based Internet of Things (IoT)," 2022.