

MANUAL BOOK
“SISTEM MONITORING DAN KONTROL ALARM KEBAKARAN RUANGAN
DENGAN SENSOR API DAN SENSOR GERAK (PIR)”

Projek Akhir Praktikum Internet of Things



Disusun Oleh : Kelompok 4 / IOT B

NAMA

Reihan Al Sya'ban
Anhar Khoirun Najib
M. Ranga Yaqub Wijaya
Auzan Hawaari Fahrany

NIM

2109106051
2109106081
2109106104
2109106127

Asisten :

Didi Nur Rahmad
2009106117

Alan Nuzulan
2009106032

Indro Dwi Saputro
2009106099

INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN

2024

DAFTAR ISI

A. LATAR BELAKANG SISTEM	3
B. FUNGSI SISTEM.....	3
C. KONSEP YANG DIGUNAKAN.....	3
D. BOARD SCHEMATIC	4
E. TAHAPAN PERANCANGAN SISTEM.....	5

A. LATAR BELAKANG SISTEM

Pentingnya keselamatan dan keamanan di lingkungan yang rentan terhadap risiko kebakaran, seperti gedung perkantoran, rumah tinggal, atau fasilitas publik. Kebakaran dapat menimbulkan kerugian besar, termasuk kehilangan nyawa, kerusakan properti, dan gangguan operasional. Oleh karena itu, deteksi dengan respons cepat terhadap kebakaran sangat penting untuk mengurangi dampak negatif yang diakibatkannya.

Sistem Monitoring dan Kontrol Alarm Kebakaran Ruangan dengan Sensor Api dan Sensor Gerak, merupakan suatu sistem yang dirancang untuk mendeteksi dan memberikan peringatan dini terhadap kebakaran dalam ruangan. Sistem ini menggunakan sensor api dan sensor gerak yang terhubung dengan jaringan Internet menggunakan broker MQTTX untuk memonitor kondisi ruangan secara *real-time* menggunakan aplikasi android yang dibuat menggunakan *tools* milik kodular.io.

Sistem akan mendeteksi adanya keberadaan api dalam ruangan menggunakan sensor api dan segera memberikan informasi kepada pengguna aplikasi dan menyalakan alarm kebakaran sebagai peringatan terdeteksinya api dalam ruangan. Sistem juga dapat memberikan informasi mengenai keberadaan seseorang dalam ruangan tersebut melalui sensor gerak (PIR). Informasi tersebut akan sangat berguna untuk tim penyelamat agar mengetahui kondisi ruangan yang telah terjadi kebakaran.

B. FUNGSI SISTEM

1. Monitoring kebakaran ruangan
2. Monitoring adanya keberadaan manusia dalam ruangan
3. Memberikan peringatan berupa alarm kebakaran jika terdeteksi api
4. Memberikan informasi berupa LED jika terdapat keberadaan manusia dalam ruangan
5. Kontrol alarm kebakaran jika terjadi kesalahan deteksi api, misalnya ada seseorang yang menyalakan korek, atau semacamnya.

C. KONSEP YANG DIGUNAKAN

1. MQTT

MQTT digunakan untuk komunikasi antar *node*. Kedua node terkoneksi pada server **broker.hivemq.com** port **1883** dengan *topic* **PA/IOT/API** dan **PA/IOT/PIR**. *Topic* **PA/IOT/API** akan memberikan informasi status dari sensor api, yang selanjutnya akan

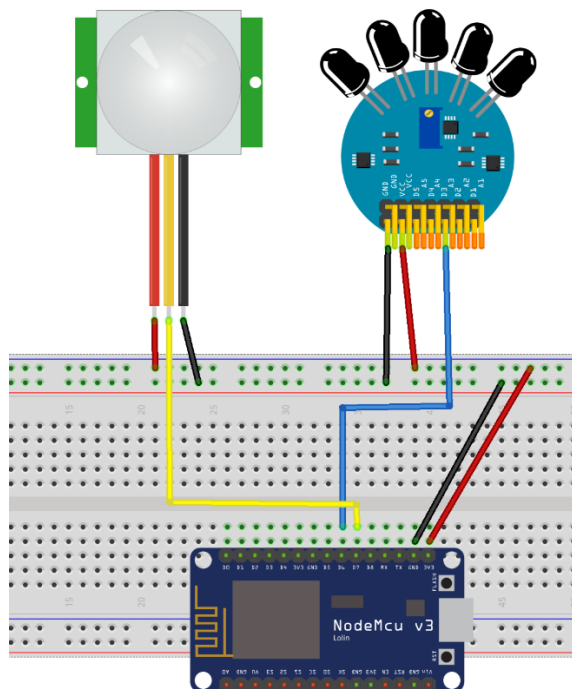
digunakan sebagai indikator alarm kebakaran. *Publisher* akan mengirimkan informasi status dari sensor api ke *topic* tersebut. Ketika *payload* dari *topic* tersebut bernilai *true* atau 1, maka akan mengindikasikan terdeteksinya api dalam ruangan dan menyalakan alarm kebakaran (*buzzer*) pada *node subscriber*. *Topic PA/IOT/PIR* akan memberikan informasi status dari sensor gerak, yang selanjutnya akan digunakan sebagai indikator manusia dalam ruangan. *Publisher* akan mengirimkan informasi status dari sensor gerak ke *topic* tersebut. Ketika *payload* dari *topic* tersebut bernilai *true* atau 1, maka akan mengindikasikan terdeteksinya manusia dalam ruangan dan menyalakan lampu LED pada *node subscriber*.

2. Kodular

Kodular digunakan sebagai *tools* untuk membuat *front-end* aplikasi android dari sistem monitoring dan kontrol kami. Kami menggunakan ekstensi UrsPahoMqttClient untuk menghubungkan antara broker MQTT dengan aplikasi android dengan *topic PA/IOT/API* dan *PA/IOT/PIR*. Pada aplikasi tersebut dapat memonitor status dari sensor api dan pir pada *node publisher*, serta mengontrol alarm kebakaran pada *node subscriber*.

D. BOARD SCHEMATIC

1. Node Publisher

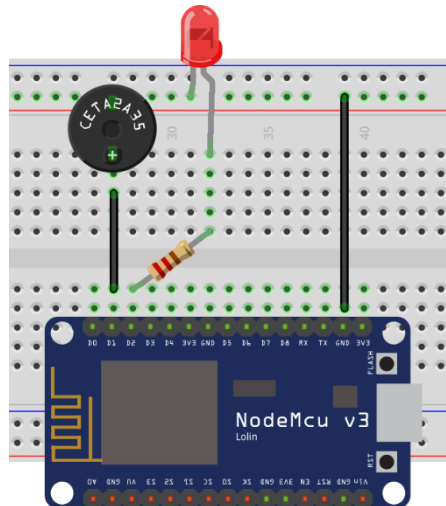


Gambar 1 Board Schematic Node Publisher

Komponen yang digunakan antara lain:

1. NodeMCU x 1
2. Sensor Api 5 Channel x 1
3. Sensor Gerak (PIR) x 1
4. Kabel Jumper Male - Male x 2
5. Kabel Jumper Male – Female x 6

2. Node Subscriber



Gambar 2 Board Schematic Node Subscriber

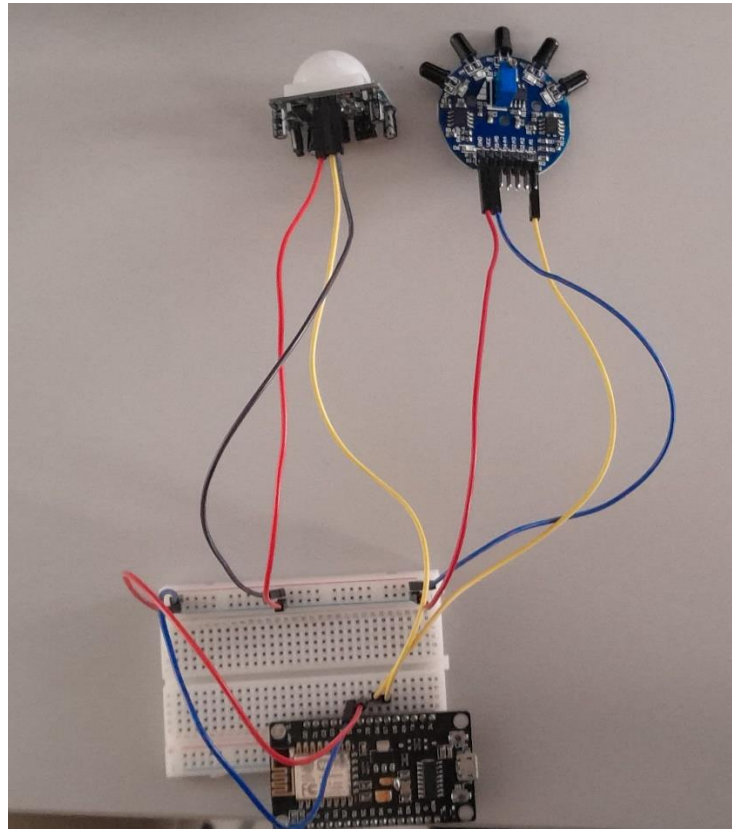
Komponen yang digunakan antara lain:

1. NodeMCU x 1
2. Piezo Buzzer x 1
3. LED x 1
4. Kabel Jumper Male – Male x 2
5. Resistor x 1

E. TAHAPAN PERANCANGAN SISTEM

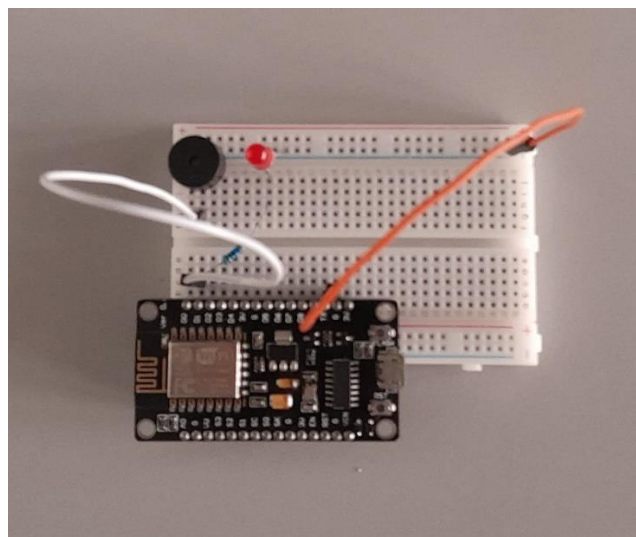
Berikut adalah cara merancang sistem monitoring dan kontrol alarm kebakaran ruangan dengan sensor api dan sensor gerak (PIR). Perancangan sistem terdiri dari tahap merangkai komponen elektronik, persiapan platform IoT, perancangan program Node MCU, dan pengujian sistem.

1. Merangkai Komponen Elektronik



Gambar 3 Rangkaian *Node Publisher*

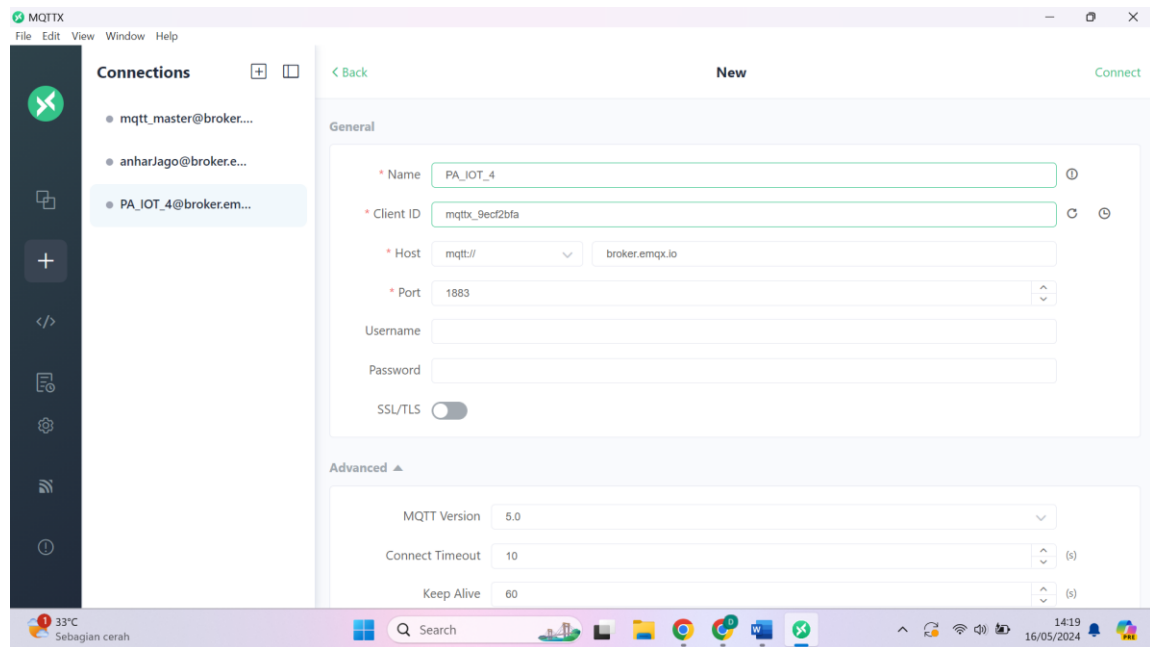
Rangkaian *node publisher* dengan sensor api dan sensor gerak yang telah dihubungkan dengan *Node MCU* oleh kabel *jumper* dan dirangkai pada *breadboard*. Rangkaian ini akan mendeteksi adanya kebakaran ruangan dan mendeteksi keberadaan manusia dalam ruangan. Selanjutnya, nilai yang didapat dari kedua sensor tersebut akan dikirimkan ke broker MQTT dan selanjutnya akan diterima oleh *subscriber*.



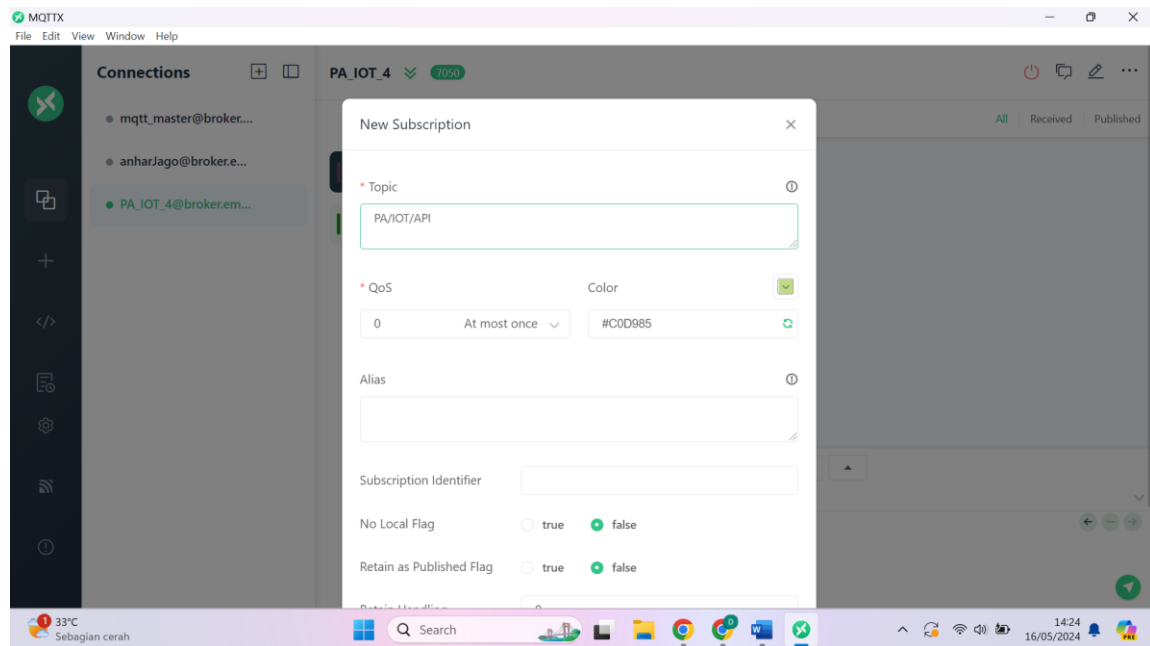
Gambar 4 Rangkaian *Node Subscriber*

Rangkaian *node subscriber* dengan menggunakan *piezo buzzer* dan lampu LED yang dihubungkan ke *Node MCU* melalui kabel jumper dan resistor yang dirangkai pada *breadboard*. Rangkaian ini akan berperan sebagai alarm kebakaran berdasarkan data sensor yang telah diterima dari broker MQTT yang dikirim oleh *node publisher*.

2. Persiapan Platform MQTT dan Kodular

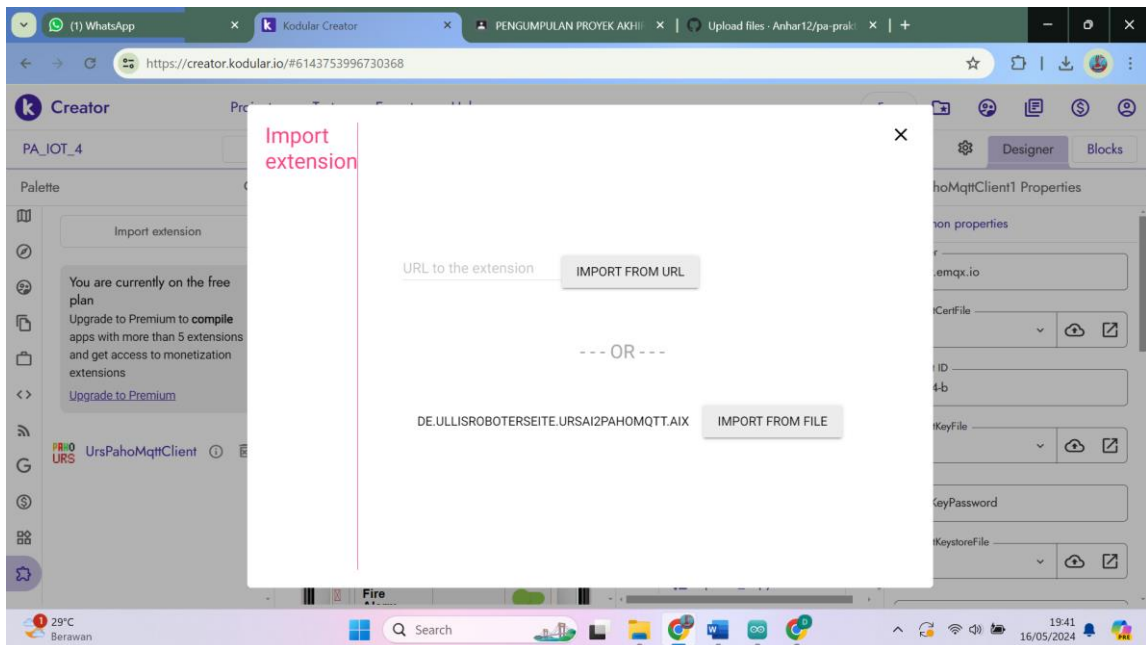


Gambar 5 Membuat Koneksi MQTTX



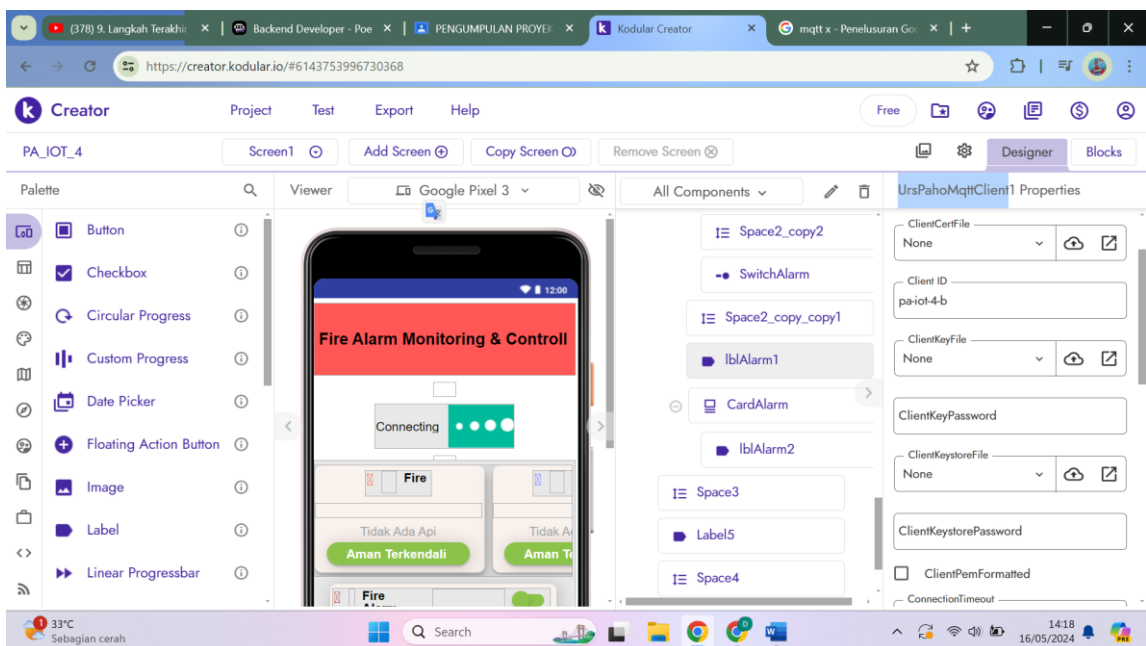
Gambar 6 Membuat Topic Subscription Baru

Membuat koneksi baru pada MQTTX dan membuat *topic subscription* baru dengan *topic* PA/IOT/API dan PA/IOT/PIR.



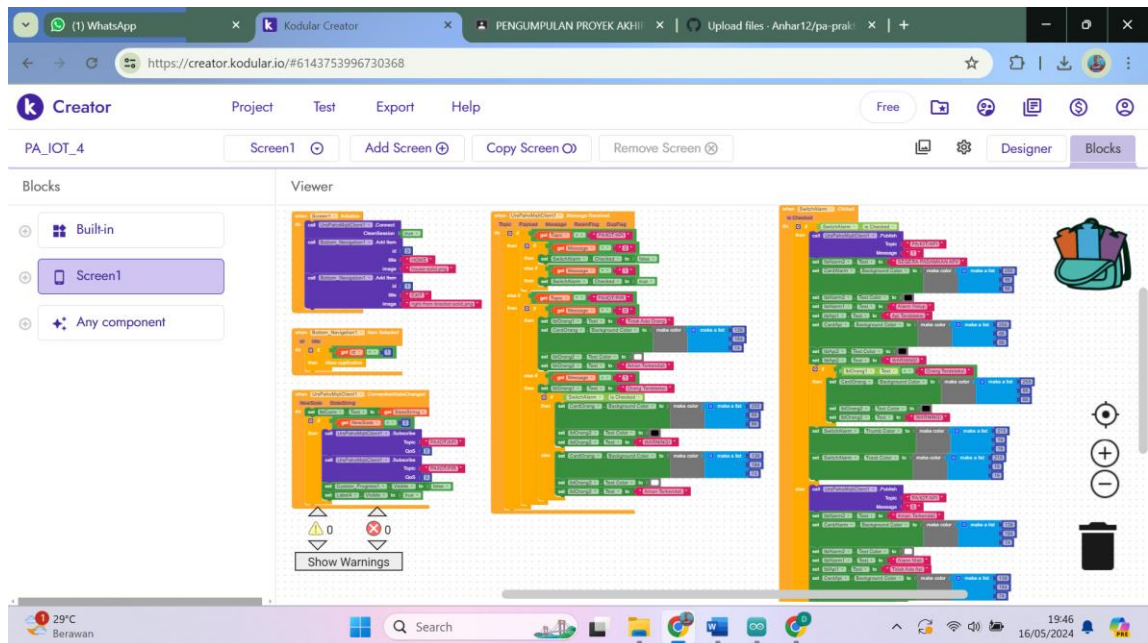
Gambar 7 Import Extension MQTT Pada Kodular

Menambahkan ekstensi untuk dapat menghubungkan antara aplikasi yang dibuat dengan Kodular dengan broker MQTT.



Gambar 8 Membuat Desain Aplikasi Pada Kodular

Membuat desain dari aplikasi yang akan digunakan untuk monitoring dan kontrol sistem alarm kebakaran ruangan.



Gambar 9 Desain *Blocks* Aplikasi Pada Kodular

Membuat desain *blocks* untuk aplikasi pada Kodular, agar dapat menerima pesan dan mengirimkan pesan ke broker MQTT.

3. Perancangan Program pada Arduino IDE

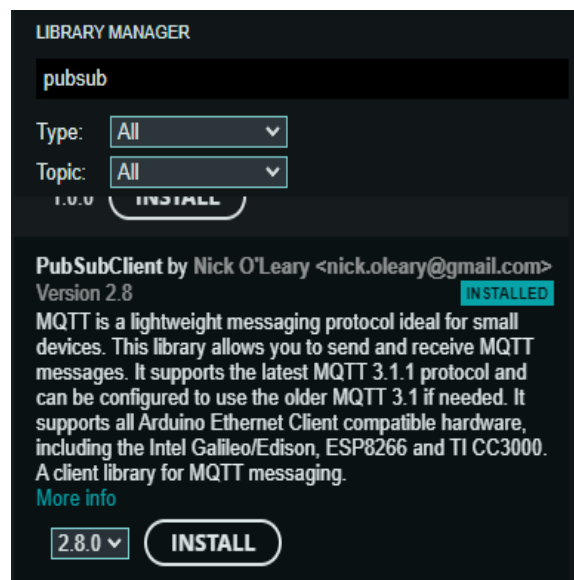
Source code dapat diakses pada link dibawah.

Node Publisher:

<https://github.com/Anhar12/pa-praktikum-iot-unmul-b4/blob/main/Publisher.ino>

Node Subscriber:

<https://github.com/Anhar12/pa-praktikum-iot-unmul-b4/blob/main/Subscriber.ino>



Gambar 10 Instalasi *Library* MQTT

Agar dapat menggunakan protokol MQTT untuk mengirim pesan, pastikan sudah menginstall library **PubSubClient** dari **Nick O’Leary**.

Tabel 1 Set Up Wifi & MQTT

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

#define BUZZ D1
#define LED D2

const char* ssid = "YourWifi";
const char* password = "YourPassword";
const char* mqtt_server = "broker.emqx.io";
const int mqtt_port = 1883;
const char* mqtt_topic1 = "PA/IOT/API";
const char* mqtt_topic2 = "PA/IOT/PIR";

WiFiClient espClient;
PubSubClient client(espClient);
```

Tabel 1 adalah *set up* pin perangkat IoT, kemudian dilanjutkan dengan *set up* wifi yang akan digunakan oleh *Node* MCU, dan *set up* agar dapat terhubung ke broker MQTT. Proses ini digunakan pada kedua *node* yang digunakan dalam sistem ini.

Tabel 2 Set Up Koneksi Wifi

```
void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
```

Tabel 2 menunjukkan proses untuk menghubungkan *Node* MCU dengan *wifi*. Proses ini digunakan untuk kedua *node* yang digunakan dalam sistem ini.

Tabel 3 *Send Nilai Dari Sensor Api dan PIR*

```
void sendSensorApi() {
    bool fire = digitalRead(fireSensor);
    String sensorData = String(fire);
    client.publish(topic1.c_str(), sensorData.c_str());
}

void sendSensorPir() {
    bool pir = digitalRead(pirSensor);
    String sensorData = String(pir);
    client.publish(topic2.c_str(), sensorData.c_str());
}
```

Tabel 3 menunjukkan fungsi untuk mengirimkan pesan ke broker mengenai nilai dari sensor api dan sensor gerak. Fungsi tersebut digunakan oleh *node publisher*.

Tabel 4 *Read Oleh Node Subscriber*

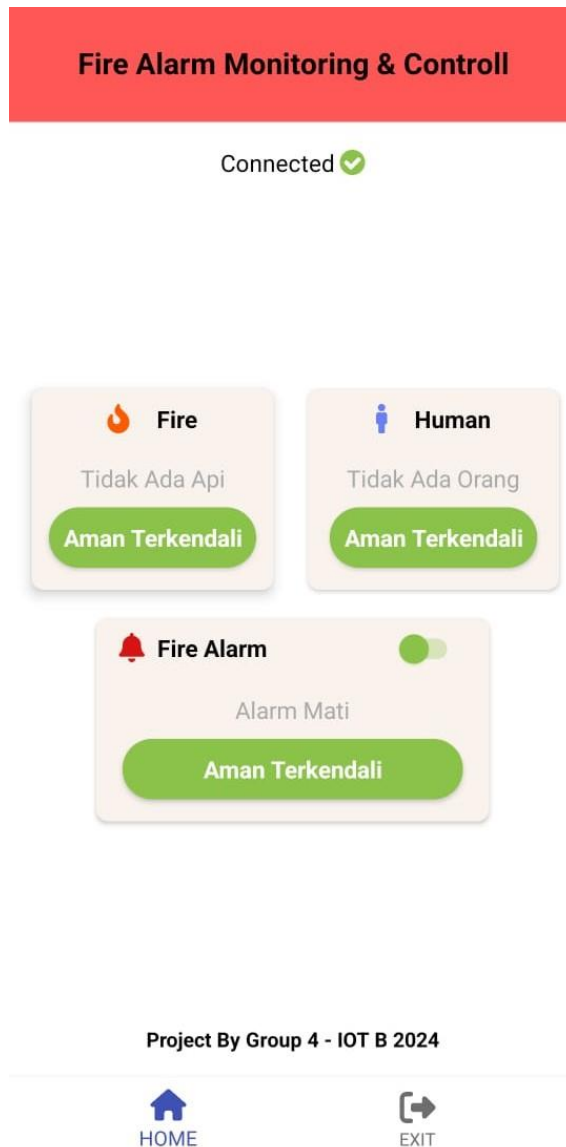
```
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Received message on topic: ");
    Serial.println(topic);
    Serial.print("Payload: ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }

    String payloadString;
    for (int i = 0; i < length; i++) {
        payloadString += (char)payload[i];
    }

    if (String(topic) == mqtt_topic1){
        if (payloadString == "1"){
            tone(BUZZ, 10);
        } else {
            noTone(BUZZ);
        }
    } else if (String(topic) == mqtt_topic2){
        if (payloadString == "1"){
            digitalWrite(LED, 1);
        } else {
            digitalWrite(LED, 0);
        }
    }
}
```

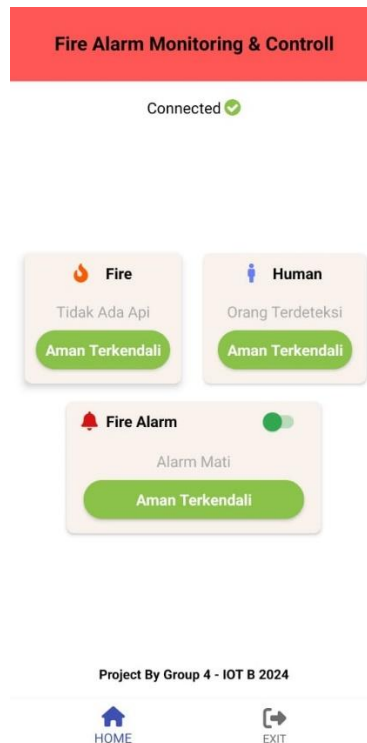
Tabel 4 menunjukkan fungsi yang digunakan oleh *node subscriber* untuk menerima data dari broker MQTT. Data tersebut selanjutnya yang akan menentukan hidup atau tidaknya alarm kebakaran ruangan.

4. Pengujian Sistem



Gambar 11 Tampilan Aplikasi Pada Kondisi Pertama

Gambar 11 menunjukkan tampilan aplikasi jika tidak ada api yang terdeteksi pada ruangan dan tidak ada terdeteksi manusia dalam ruangan.



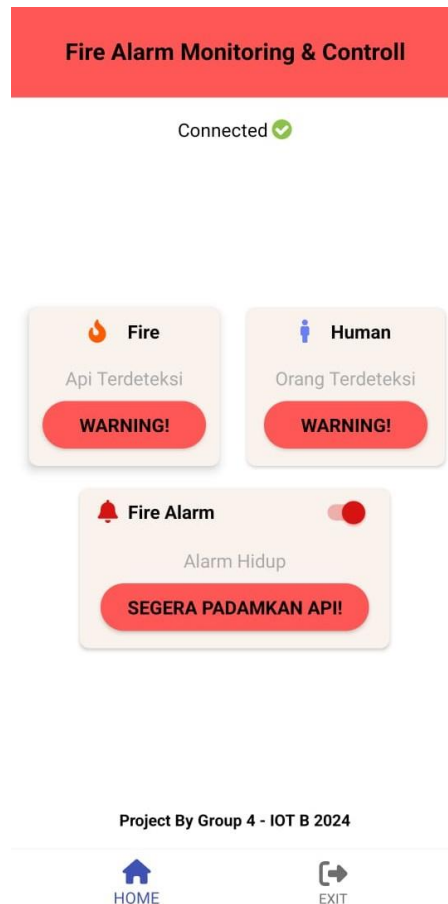
Gambar 12 Tampilan Pada Kondisi Ketiga

Gambar 12 menunjukkan tampilan aplikasi jika tidak ada api yang terdeteksi pada ruangan dan ada terdeteksi manusia dalam ruangan.



Gambar 13 Tampilan Pada Kondisi Ketiga

Gambar 13 menunjukkan tampilan aplikasi jika ada api yang terdeteksi pada ruangan dan tidak ada terdeteksi manusia dalam ruangan.



Gambar 14 Tampilan Pada Kondisi Keempat

Gambar 14 menunjukkan tampilan aplikasi jika ada api yang terdeteksi pada ruangan dan ada terdeteksi manusia dalam ruangan.