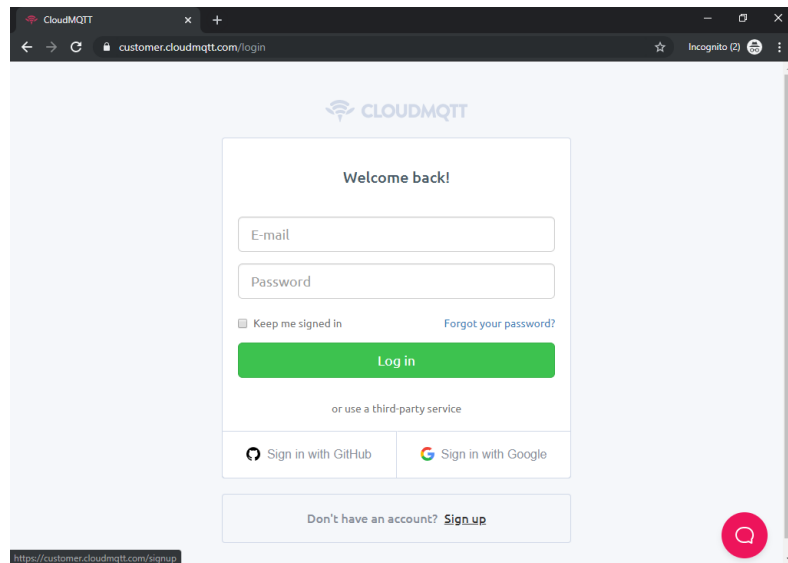


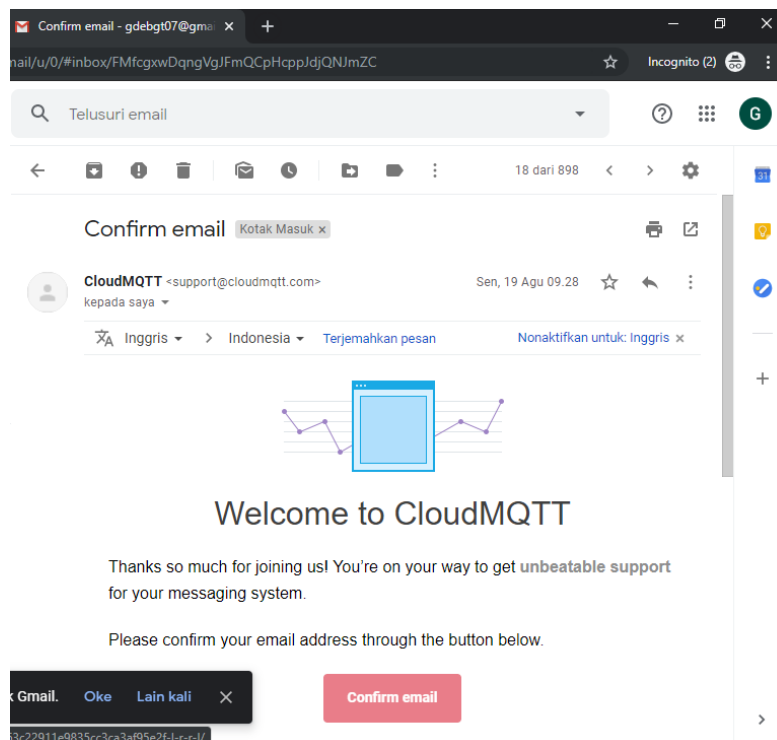
Langkah-langkah implementasi penggunaan Publisher (esp32) dan subscriber (laptop atau hp) menggunakan CloudMQTT sebagai Broker

Setting Broker

1. Buka website CloudMQTT di <https://www.cloudmqtt.com/>
2. Daftar dengan menggunakan email yang anda miliki di menu Login, lalu sign up



3. Buka email anda, lalu lakukan konfirmasi email



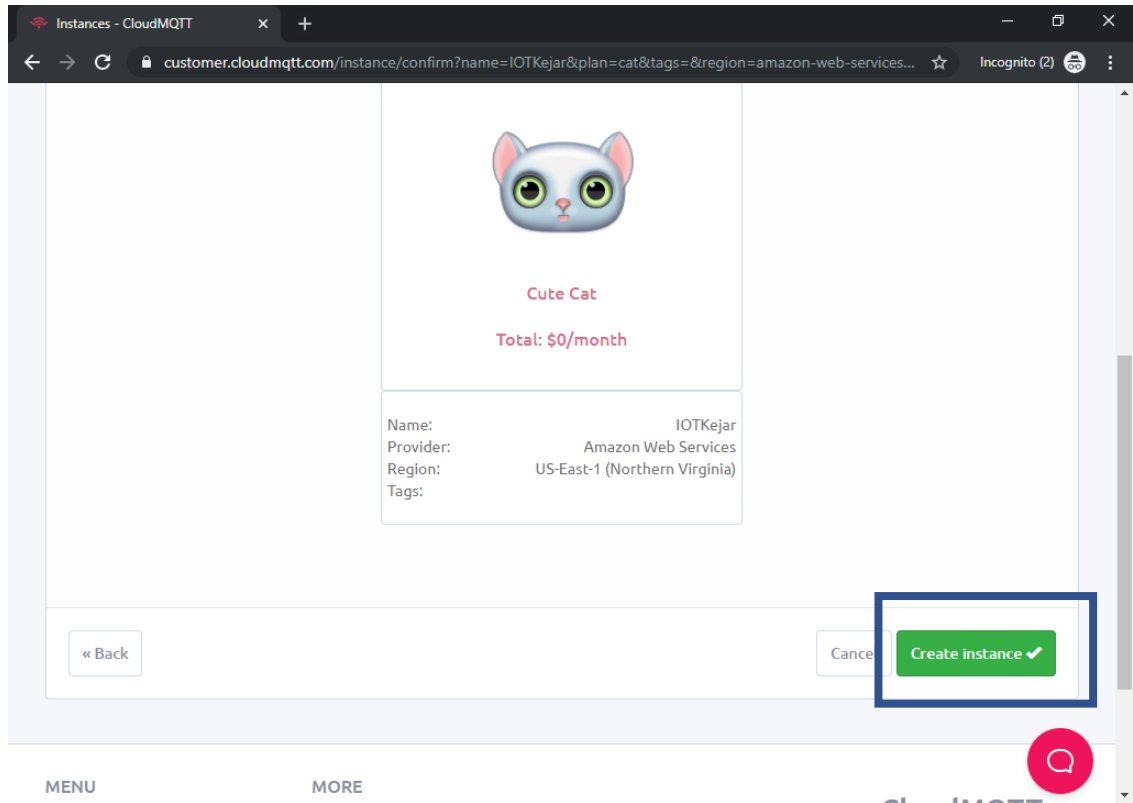
4. Login dengan menggunakan account yang telah anda buat
5. Klik Create New Instance
6. Masukkan nama instance yang diinginkan, lalu klik select regional

The screenshot shows the 'Create new instance' page in the CloudMQTT web interface. The browser address bar shows 'customer.cloudmqtt.com/instance/create'. The page has a progress bar at the top with four steps: Plan (active), Region, Configure (Dedicated plans only), and Confirm. The main content area is titled 'Select a plan and name - Step 1 of 4'. It contains three input fields: 'Name' with the value 'IOTKejar', 'Plan' with a dropdown menu showing 'Cute Cat (Free)', and 'Tags' which is empty. Below the 'Tags' field, there is explanatory text: 'Tags are used to separate your instances between projects. This is primarily used in the project listing view for easier navigation and access control. Tags allow admins to manage team members access to different groups of instances.' To the right of the input fields, there is a 'Plan' section featuring a cat icon and the text 'Cute Cat'. Below this, it says 'See the plan page to learn about the different plans.' At the bottom right of the form, there are two buttons: 'Cancel' and 'Select Region', with the 'Select Region' button highlighted by a blue box. A red speech bubble icon is visible in the bottom right corner of the page.

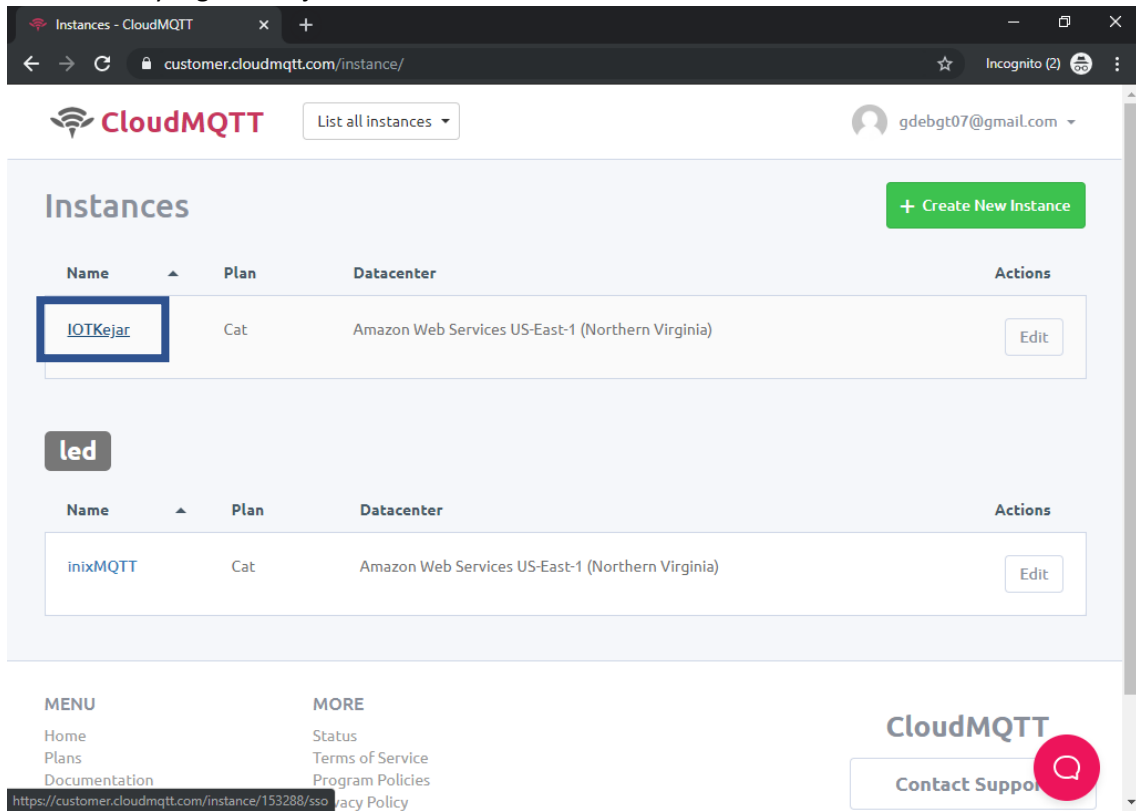
7. Lalu Klik Review

The screenshot shows the 'Create new instance' page in the CloudMQTT web interface, specifically Step 2 of 4: 'Select a region and data center'. The browser address bar shows 'customer.cloudmqtt.com/instance/region?name=IOTKejar&plan=cat&tags='. The page has a progress bar at the top with four steps: Plan, Region (active), Configure (Dedicated plans only), and Confirm. The main content area is titled 'Select a region and data center - Step 2 of 4'. It contains a 'Data center' dropdown menu with the value 'US-East-1 (Northern Virginia)'. Below the dropdown menu, there is the AWS logo. At the bottom left of the form, there is a '« Back' button. At the bottom right, there are two buttons: 'Cancel' and 'Review', with the 'Review' button highlighted by a blue box. A red speech bubble icon is visible in the bottom right corner of the page.

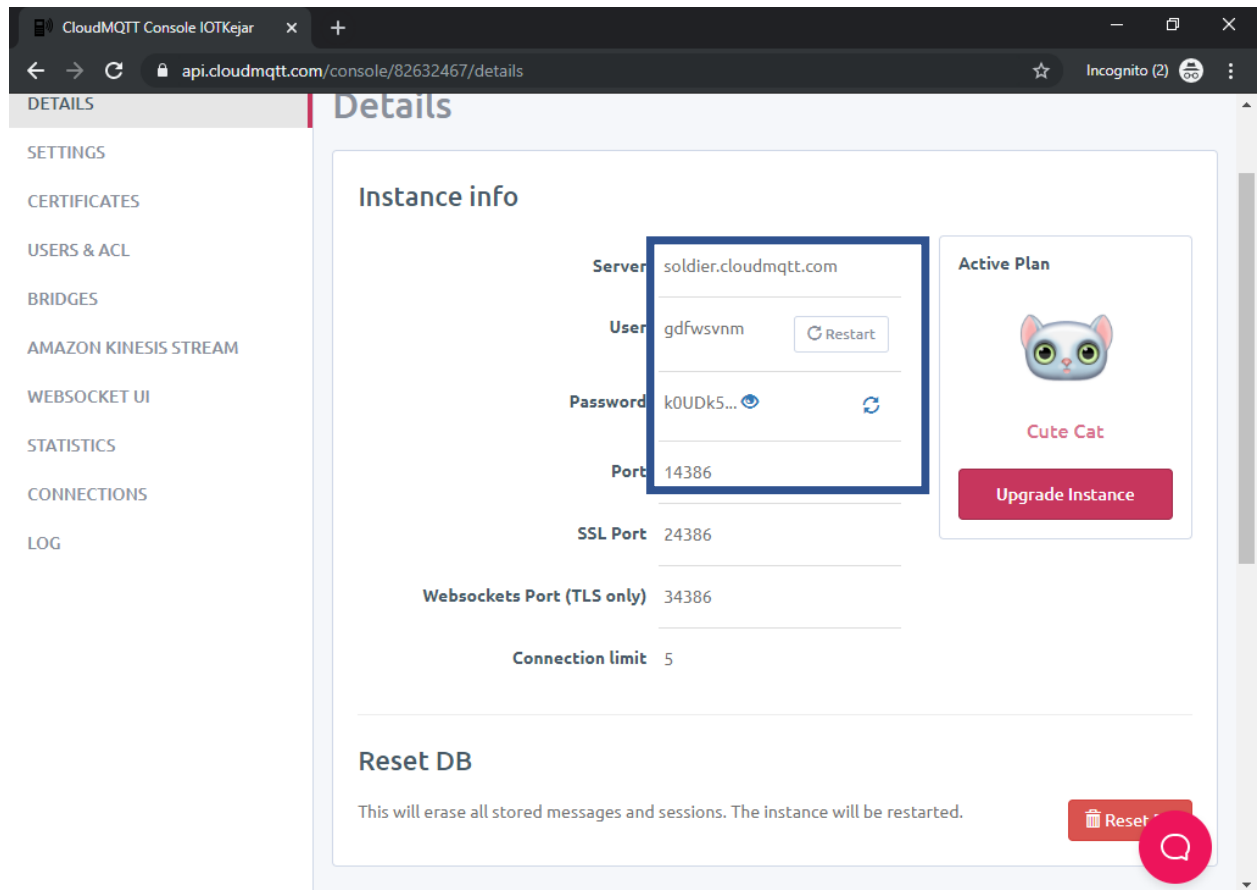
8. Lalu klik Create Instance



9. Klik instance yang baru saja di buat



10. Catat user, password, port dan alamat mqtt broker yang akan digunakan



The screenshot shows the CloudMQTT Console IOTKejar interface. The browser address bar displays `api.cloudmqtt.com/console/82632467/details`. The left sidebar contains a menu with the following items: DETAILS (selected), SETTINGS, CERTIFICATES, USERS & ACL, BRIDGES, AMAZON KINESIS STREAM, WEBSOCKET UI, STATISTICS, CONNECTIONS, and LOG.

The main content area is titled "Details" and features a section for "Instance info". A blue rectangular box highlights the following fields:

- Server: `soldier.cloudmqtt.com`
- User: `gdfwsvnm` (with a "Restart" button)
- Password: `k0UDk5...` (with an eye icon and a refresh icon)
- Port: `14386`

Below the highlighted fields, the following information is displayed:

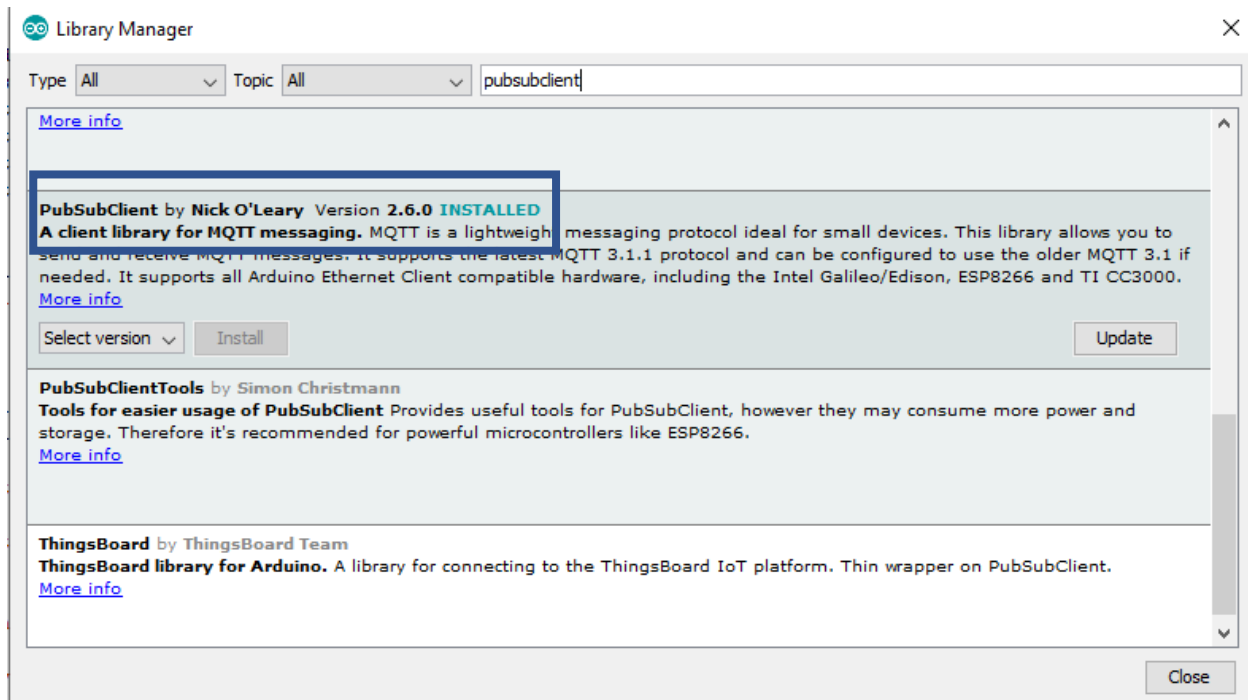
- SSL Port: `24386`
- Websockets Port (TLS only): `34386`
- Connection limit: `5`

To the right of the instance info, there is an "Active Plan" section featuring a cat icon, the text "Cute Cat", and an "Upgrade Instance" button.

At the bottom of the page, there is a "Reset DB" section with the text: "This will erase all stored messages and sessions. The instance will be restarted." and a "Reset" button.

Setting Publisher

1. Buatlah rangkaian IoT Device dengan salah satu sensor (misalkan dalam percobaan ini digunakan sensor LDR)
2. Install library berikut



3. Masukkan sketch berikut

```
#include <WiFi.h>
```

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

```
const char* ssid = "di isi dengan nama wifi";  
const char* password = "di isi dengan password wifi";  
const char* mqttServer = "di isi sesuai mqtt broker";  
const int  mqttPort = di isi sesuai mqtt broker;  
const char* mqttUser = " di isi sesuai mqtt broker ";  
const char* mqttPassword = " di isi sesuai mqtt broker ";  
int sensor = 33;
```

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

```
void setup() {
```

```
  Serial.begin(115200);  
  WiFi.begin(ssid, password);
```

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("Connecting to WiFi..");
}

Serial.println("Connected to the WiFi network");

client.setServer(mqttServer, mqttPort);

while (!client.connected()) {
  Serial.println("Connecting to MQTT...");

  if (client.connect("ESP32Client", mqttUser, mqttPassword )) {

    Serial.println("connected");

  } else {

    Serial.print("failed with state ");
    Serial.print(client.state());
    delay(2000);

  }
}
pinMode(sensor,INPUT);

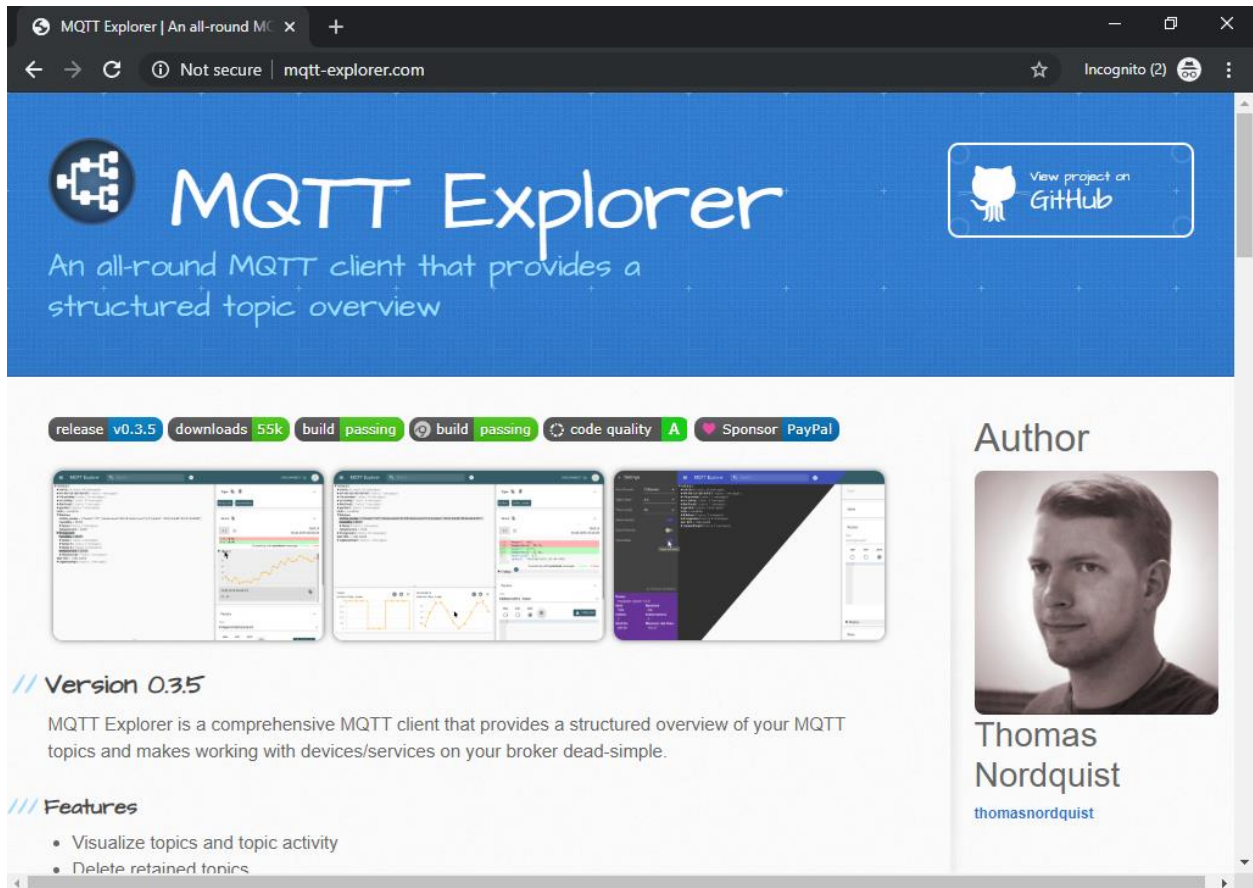
}

void loop() {
  client.loop();
  int SensorLDR=analogRead(sensor);
  char hasil[4];
  dtostrf(SensorLDR, 4, 0, hasil);
  Serial.println(SensorLDR);
  client.publish("cahaya", hasil);
  delay (5000);
}

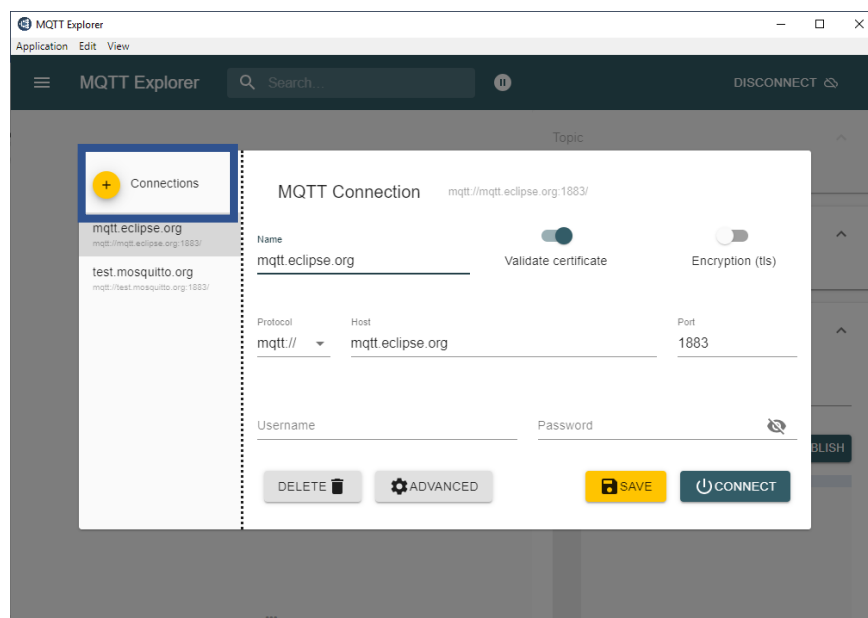
```

Setting Subcsriber di Laptop

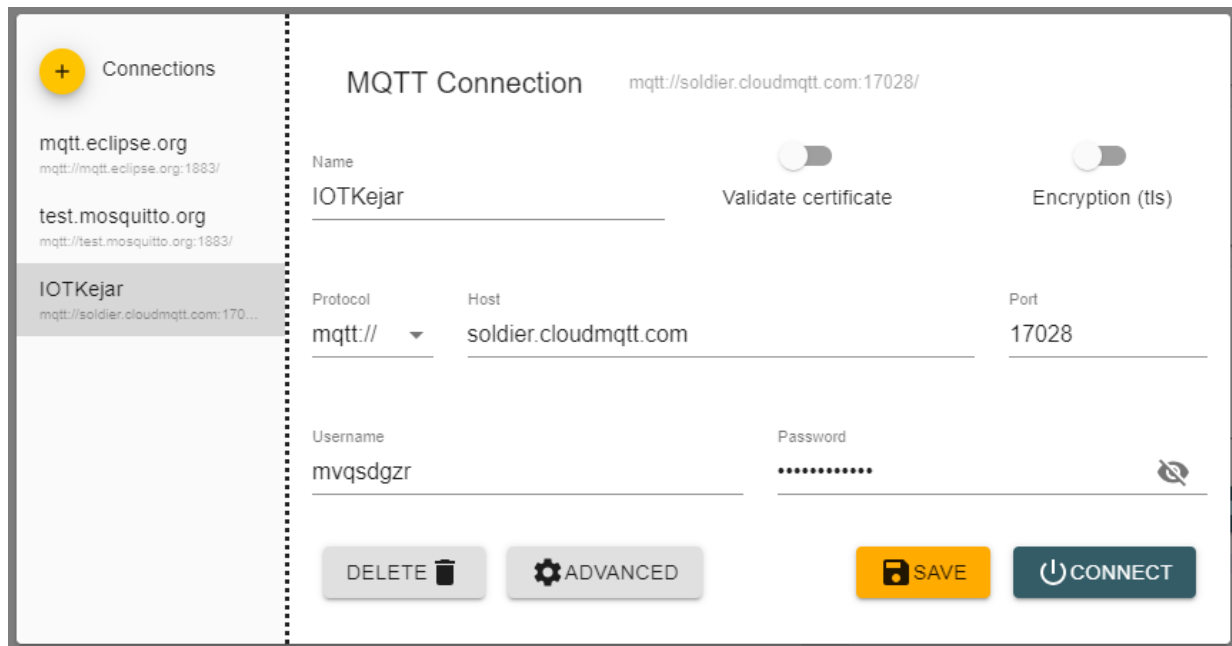
1. Download dan install aplikasi MQTTExplorer dari halaman web <http://mqtt-explorer.com/>



2. Tambahkan koneksi baru

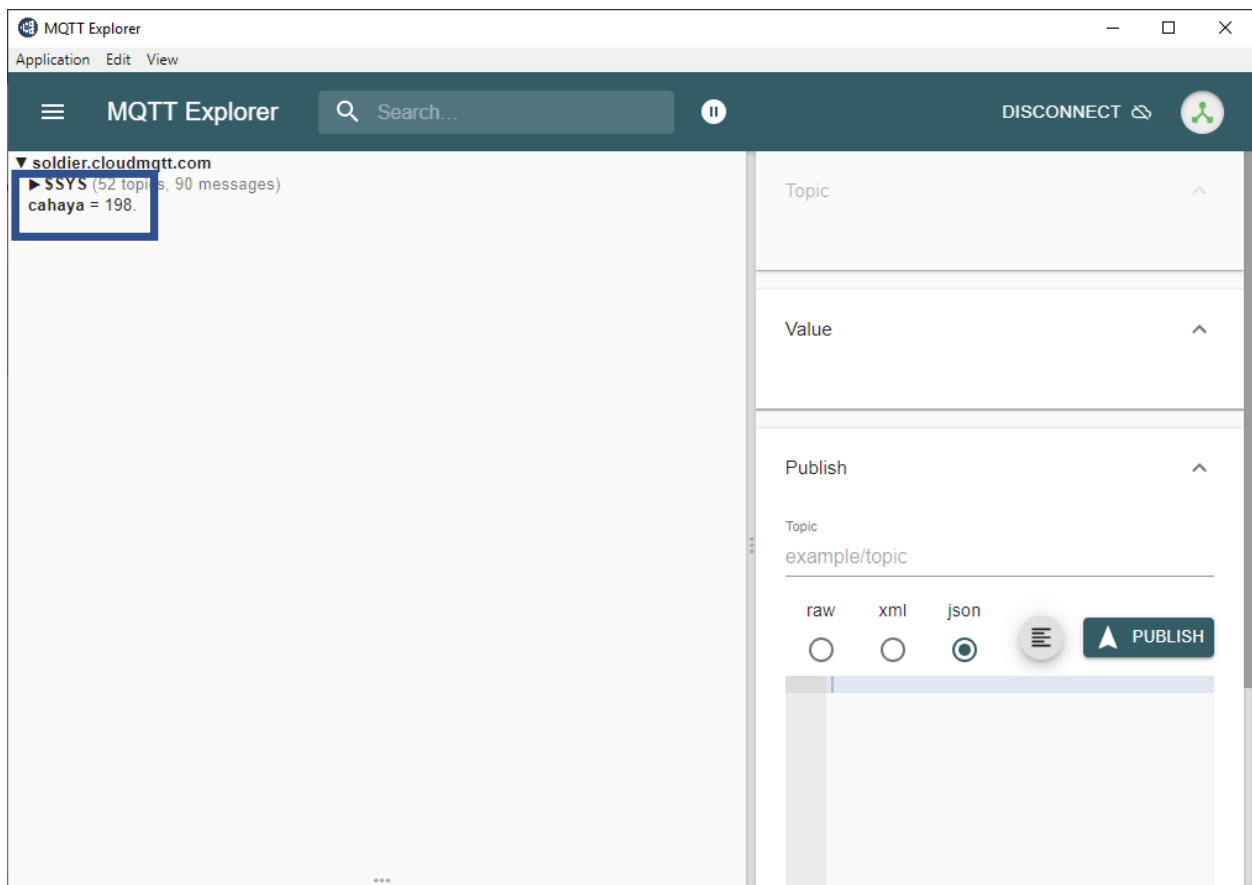


3. Masukkan semua informasi yang di butuhkan, lalu klik connect



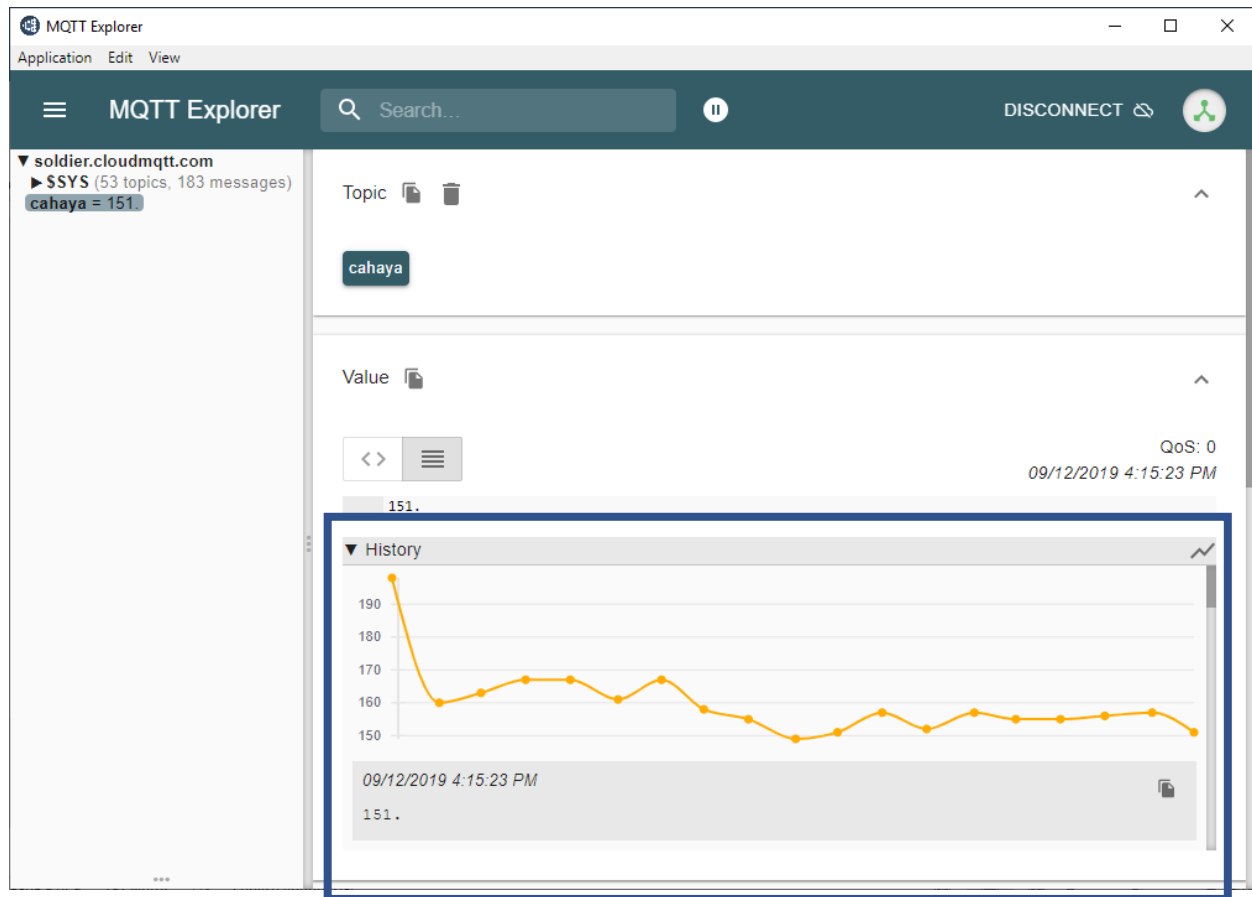
The image shows the 'MQTT Connection' configuration window. On the left, a sidebar lists connections: 'mqtt.eclipse.org', 'test.mosquitto.org', and 'IOTKejar' (selected). The main area is titled 'MQTT Connection' with the address 'mqtt://soldier.cloudmqtt.com:17028/'. It contains fields for 'Name' (IOTKejar), 'Protocol' (mqtt://), 'Host' (soldier.cloudmqtt.com), 'Port' (17028), 'Username' (mvqsdgxr), and 'Password' (masked with dots). There are toggle switches for 'Validate certificate' and 'Encryption (tls)'. At the bottom, there are buttons for 'DELETE', 'ADVANCED', 'SAVE', and 'CONNECT'.

4. Klik topik yang kita telah buat



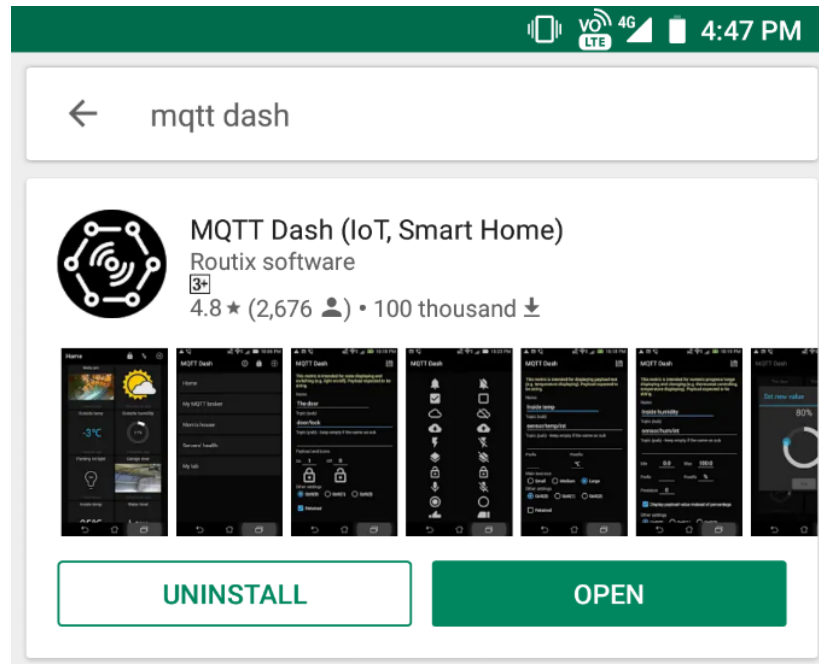
The image shows the 'MQTT Explorer' application window. The top bar includes a menu icon, the title 'MQTT Explorer', a search bar, and a 'DISCONNECT' button. The main area is divided into two panels. The left panel shows a tree view of the MQTT broker 'soldier.cloudmqtt.com' with a topic 'SSYS (52 topics, 90 messages)' and a message 'cahaya = 198'. The right panel shows the details of the selected topic, including a 'Topic' field with the value 'example/topic', a 'Value' field, and a 'Publish' section with a 'PUBLISH' button.

5. Jika ingin melihat hasil dan grafiknya, klik history

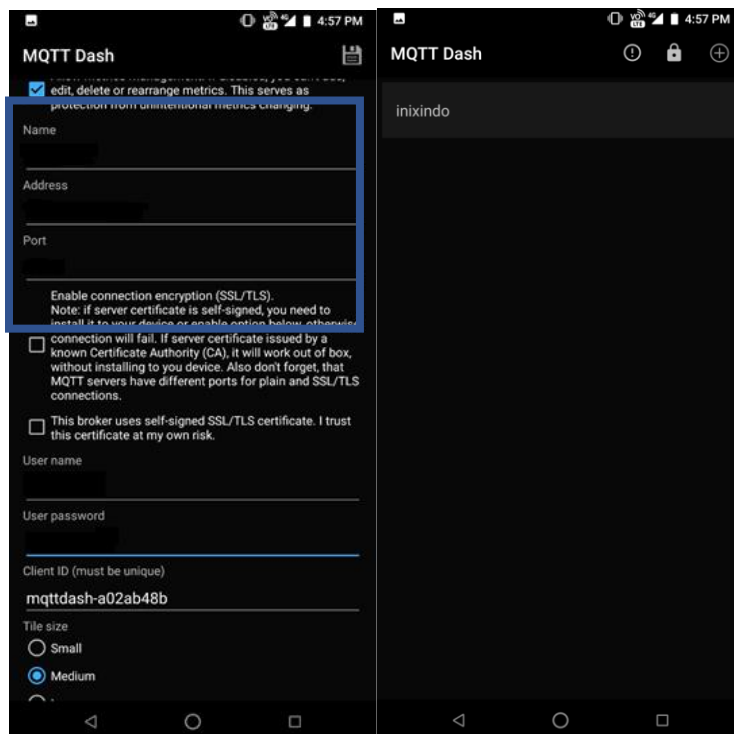


Setting Subcsriber di Handphone

1. Install MQTT Client (dalam lab ini yang akan digunakan adalah MQTTDash)



2. Masukkan informasi MQTTBroker (nama bisa di isi bebas)



3. Tambahkan topik dengan tipe text, tulis nama (bebas) dan topik (sesuai topik yang di buat), lalu save

ESP 32 sebagai Publisher

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

const char* ssid = "di isi dengan ssid wifi";
const char* password = "diisi dengan password wifi";
const char* mqtt_server = "diisi dengan alamat broker";
const int  mqtt_port  = <portmqtt>;
const char* mqtt_user  = "username mqtt";
const char* mqtt_pass  = "password mqtt";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

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(".");
  }

  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  if (strcmp(topic, "led1")==0){
    for (int i=0;i<length;i++) {
      char receivedChar = (char)payload[i];
      if (receivedChar == '1')
        digitalWrite(BUILTIN_LED, HIGH);
    }
  }
}
```

```

    if (receivedChar == '0')
        digitalWrite(BUILTIN_LED, LOW);
    }
}

void reconnect() {

    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");

        String clientId = "ESP32Client-";
        clientId += String(random(0xffff), HEX);

        if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {
            Serial.println("connected");

            client.subscribe("led1");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");

            delay(5000);
        }
    }
}

void setup() {
    pinMode(BUILTIN_LED, OUTPUT);
    Serial.begin(9600);
    setup_wifi();
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(callback);
}

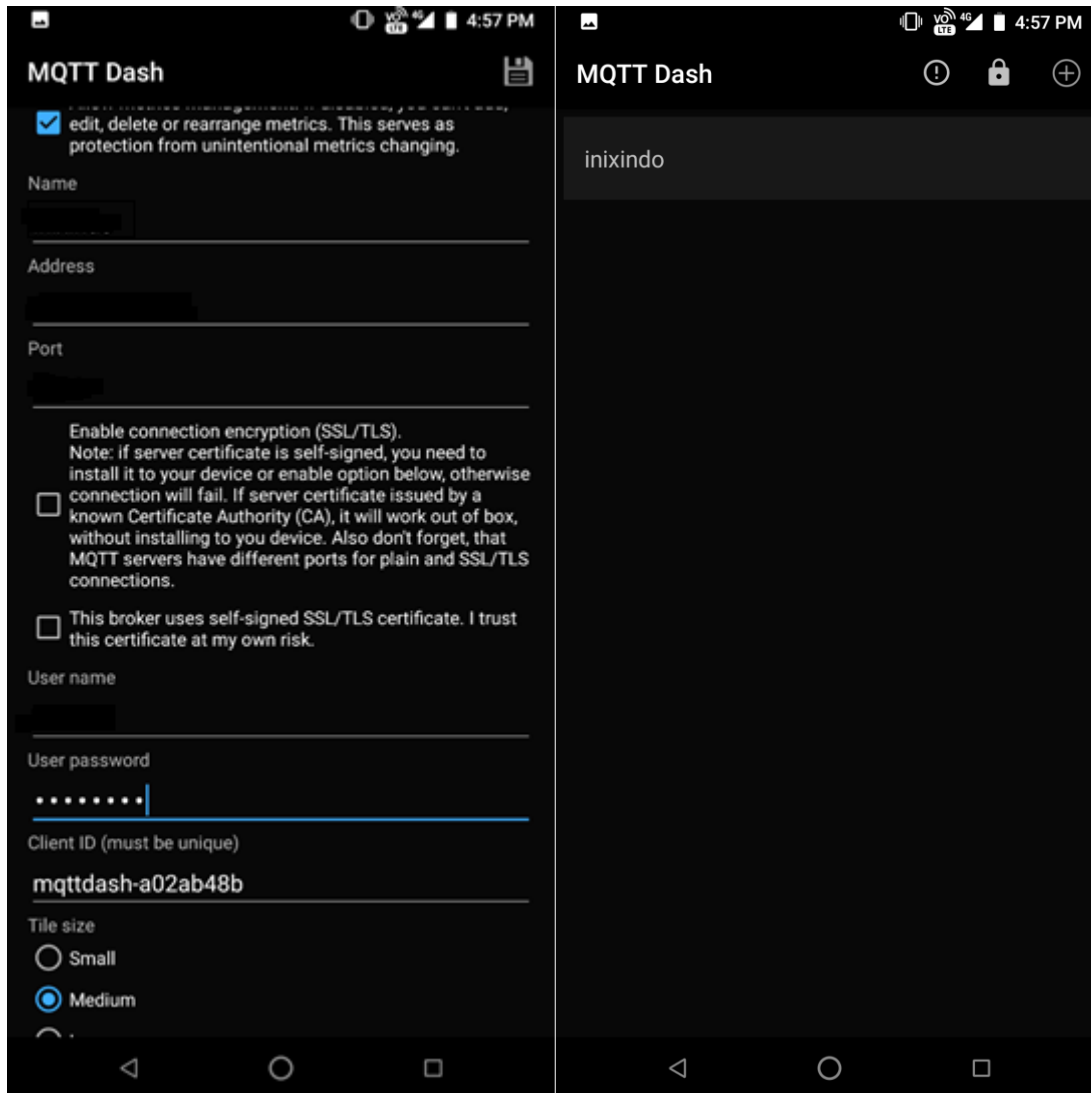
void loop() {

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

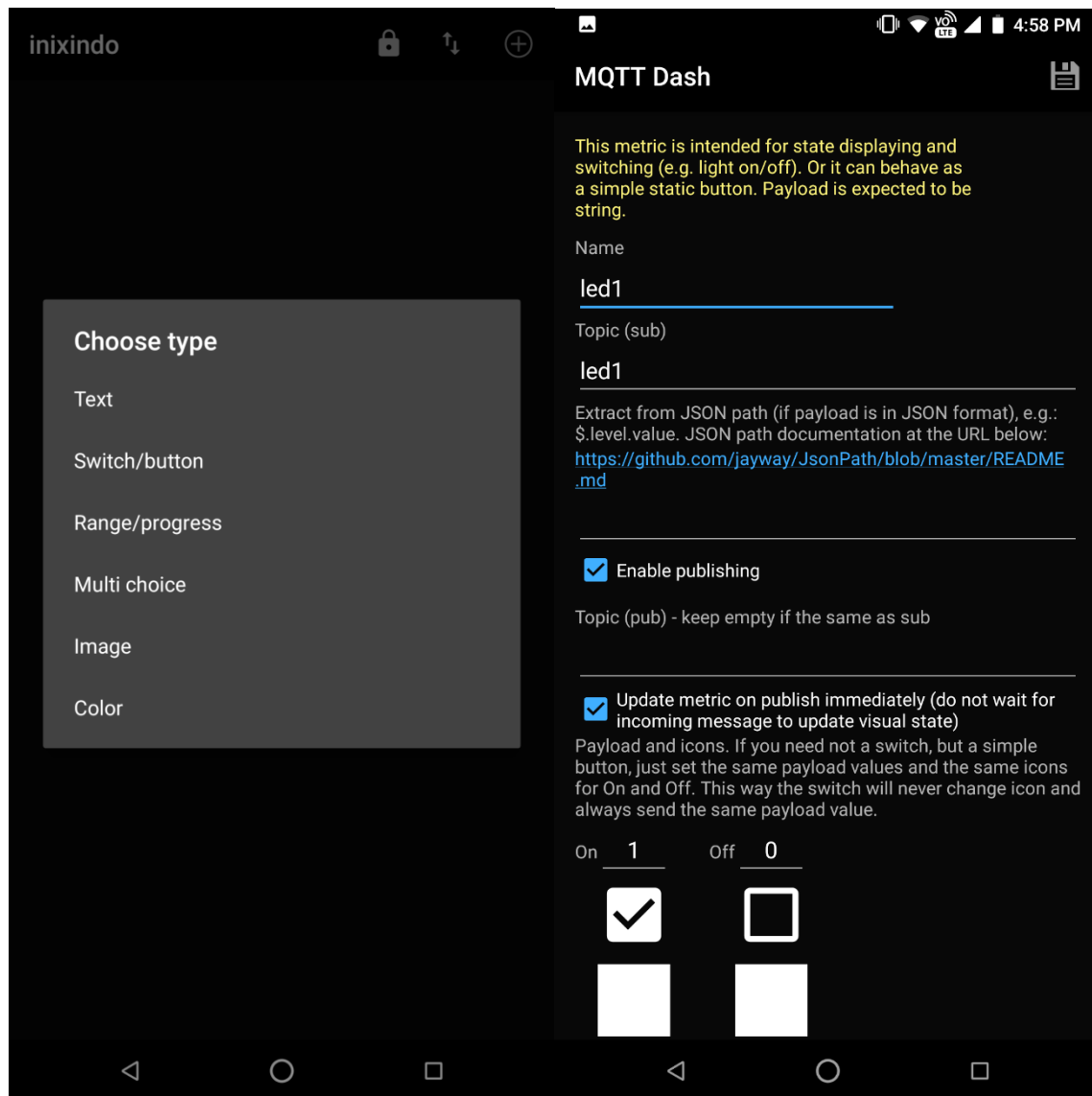
```

Setting MQTTDash

- Masukkan alamat broker, username dan password



- Tambahkan topik sesuai dengan jenis topik yang akan di gunakan



- Test

inixindo



inixindo



led1

led2



led1

led2



6 seconds ago

2 seconds ago

