

Kelompok 1

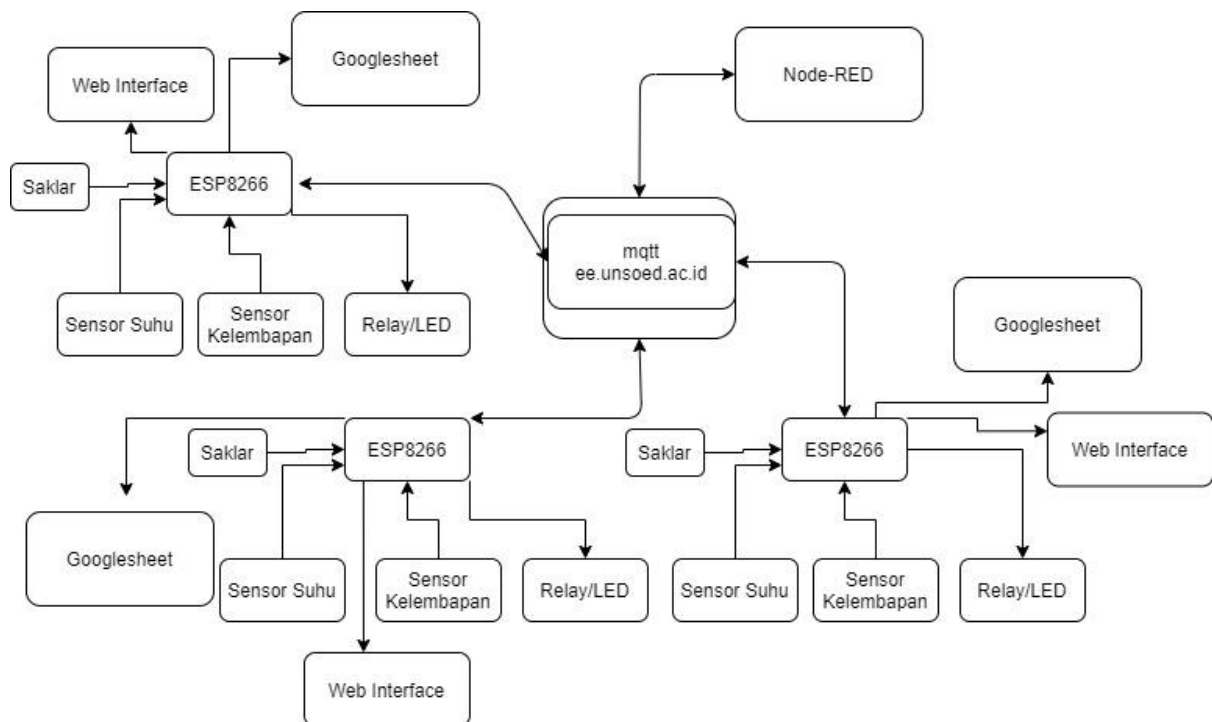
Anggota Kelompok :

- M. Sophian Alwi R. (H1A018008)
- Dinar Zidni Ilman (H1A018012)
- Haris Prasetyo (H1A018064)
- Muhammad Labib H. (H1A018091)

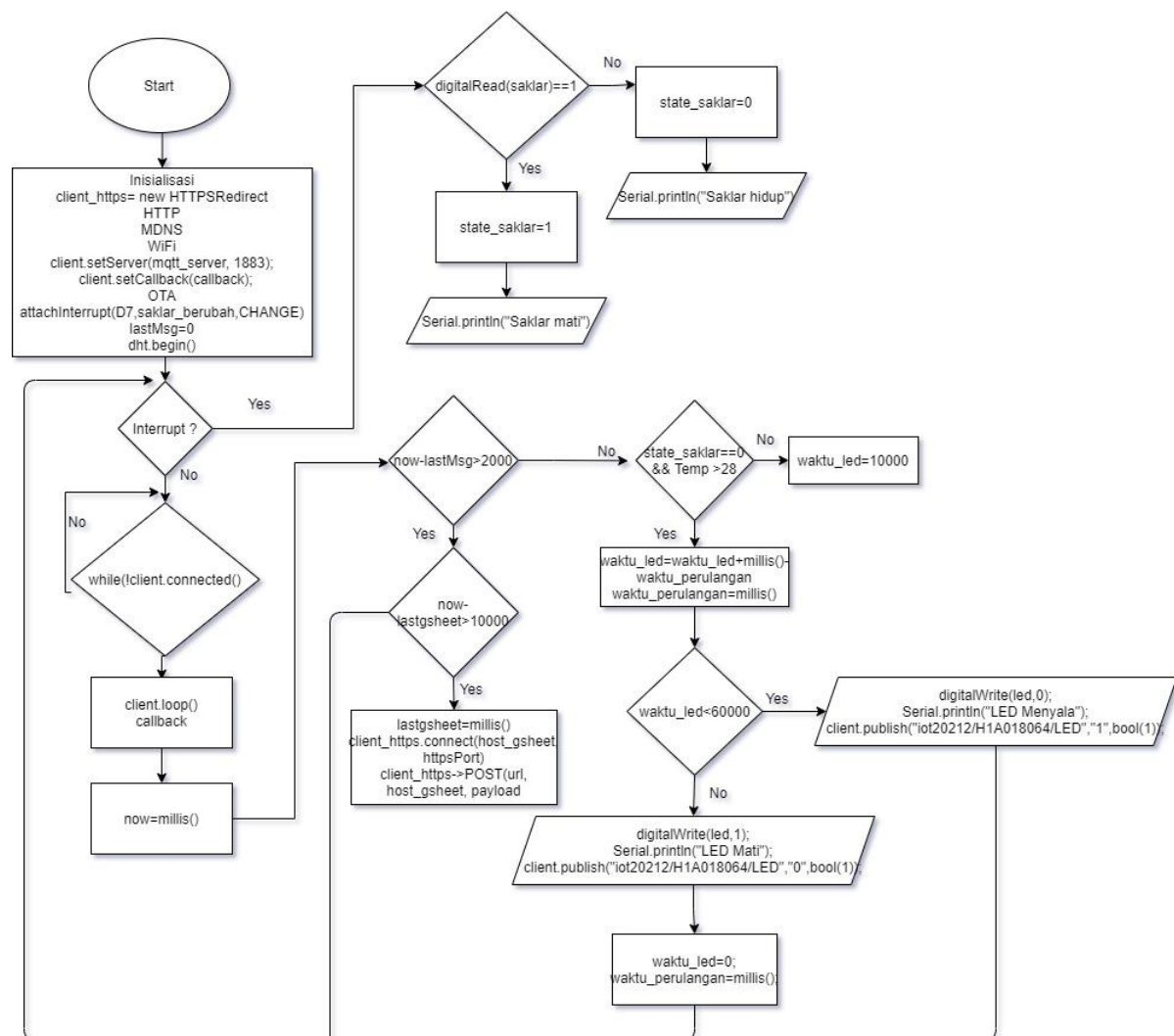
TUGAS BESAR

Pada tugas besar kali ini kelompok 1 yang beranggotakan Sophian, Haris, Dinar dan Labib. Dimana kelompok kami akan membuat sistem IoT dengan menggunakan 3 set perangkat IoT sekaligus dimana 1 set nya memiliki 1 esp8266, 1 sensor DHT11, dan 1 buah relay dual channel. Sehingga total perangkat yang ada yaitu 3 nodeMcu, 3 Sensor DHT11, dan 3 Relay dual channel. Setiap device terhubung dengan relay sebagai pengganti AC, sensor suhu dan sensor kelembaban. Data dari tiap device disimpan ke Googlesheet sehingga tiap device memiliki database masing - masing. Tiap device terhubung dengan Node-RED FRED dan mqtt sebagai middleware dan API yang terhubung dengan 'ee.unsoed.ac.id' sebagai servernya. Node-RED FRED untuk menampilkan grafik dari suhu yang terdeteksi dan MQTT untuk memonitor kondisi dari AC nya. Data yang diambil oleh Node-RED FRED dan MQTT bukan data langsung dari tiap devicenya melainkan dari googlesheet..

Arsitektur Sistem :



Flowchart :

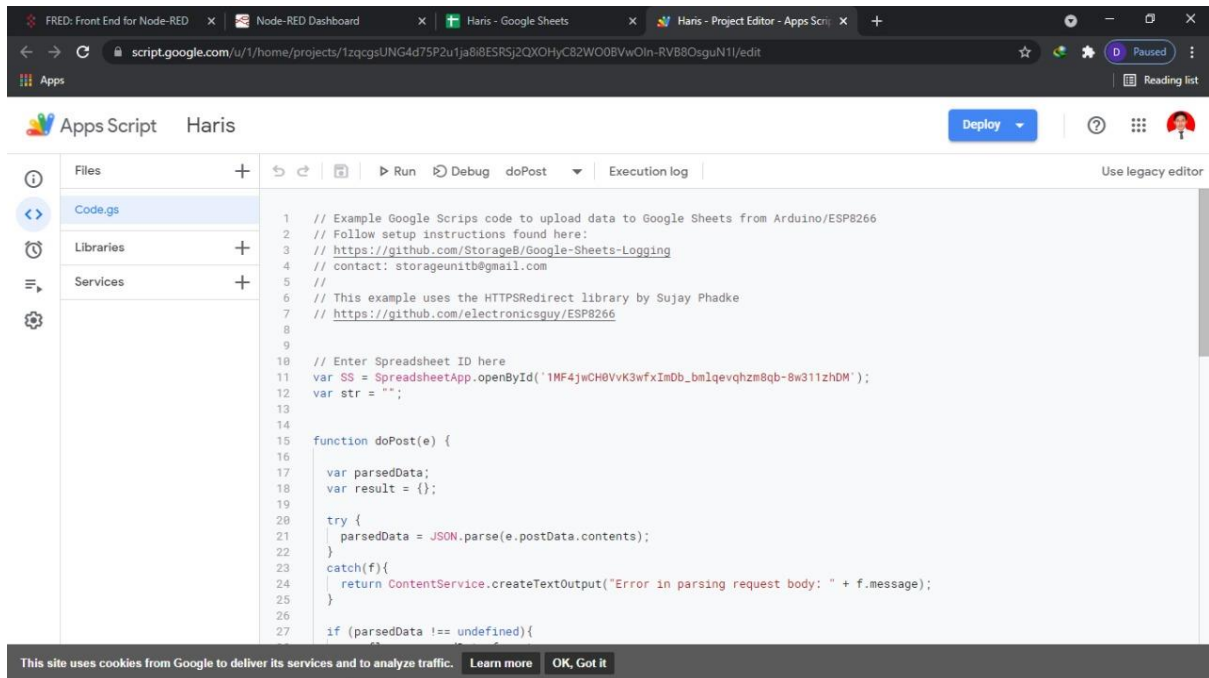


Disini kelompok kami mengumpamakan sebuah AC dengan LED. Alur dari ekosistem yang kami rencanakan adalah ketika AC di dalam sebuah ruangan mendeteksi suhu diatas 28 °C maka AC akan menyala selama 1 menit. Ketika AC sudah menyala selama 1 menit dan suhu ruangan sudah dibawah 28 °C maka AC akan langsung mati. Tetapi ketika AC sudah berjalan selama 1 menit dan suhu ruangan masih diatas 28 °C maka device tersebut akan meminta untuk dinyalakan kembali atau langsung dimatikan melalui <https://fred.sensetecnic.com/> . Ketika AC diperintahkan untuk menyala, maka AC akan menyala kembali selama 1 menit sampai AC sudah mendeteksi suhu dibawah 28 °C atau diminta untuk mati.

Cara Kerja :

Pertama kami menocba untuk mengitegrasikan esp8266 kami dengan google spreadsheet dengan cara:

- 1). Membuat spreadsheet yang baru
- 2). Pada menu tools masuk ke script editor, kemudian memasukan sumber kode berikut :



- 3). Kita mendeklarasikan header pada source code seperti berikut :

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
```

- 4). Setelah itu mendeklarasikan variabel agar terdeteksi ke google spreadsheet :

```
const char *GScriptId = "AKfycbwmO-Vpet0YmfHmsWZLLiZHQIFp7Y1orV5vsypDLDXWgezGG9zsokeaYaRhschw2tN";
String payload_base = "{\\command\\: \\\"insert_row\\\", \\\"sheet_name\\\": \\\"Sheet1\\\", \\\"values\\\": ";
String payload = "";
const char* host_gsheel = "script.google.com";
const int httpsPort = 443;
```

5). Menambahkan perintah agar terhubung ke google spreadsheet :

```
httpUpdater.setup(&server);
client_https = new HTTPSRedirect(httpsPort);
client_https->setInsecure();
client_https->setPrintResponseBody(true);
client_https->setContentTypeHeader("application/json");
Serial.print("Connecting to ");
Serial.println(host_gsheets);

bool flag = false;
for (int i=0; i<5; i++){
    int retval = client_https->connect(host_gsheets, httpsPort);
    if (retval == 1){
        flag = true;
        Serial.println("Connected");
        break;
    }
    else
        Serial.println("Connection failed. Retrying...");
}
```

6). Lalu menambahkan prosedur berupa saklar mekanik agar mengetahui saklar mekanik dalam keadaan on atau off :

```
void ICACHE_RAM_ATTR saklar_berubah(){
    if(digitalRead(saklar)==1){
        state_saklar=1;
        Serial.println("Saklar OFF");
    }
    else{
        Serial.println("Saklar ON");
        state_saklar=0;
    }
}
```

7). Lalu menambahkan perintah di prosedur callback dimana callback akan sesuai dengan mqtt dash :

```
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 1 was received as first character
    if ((char)payload[0] == '1') {
        //digitalWrite(, LOW);
        state_saklar=0;
    }
    else {
        //digitalWrite(led, HIGH);
        state_saklar=1;
    }
}
```

8). Langkah selanjutnya kami memodifikasi sumber kode untuk menyalakan relay ketika suhu diatas 28 °C dan ketika suhu dibawah 28 °C relay mati. Kami menambahkan pengkondisian pada bagian void loop seperti berikut :

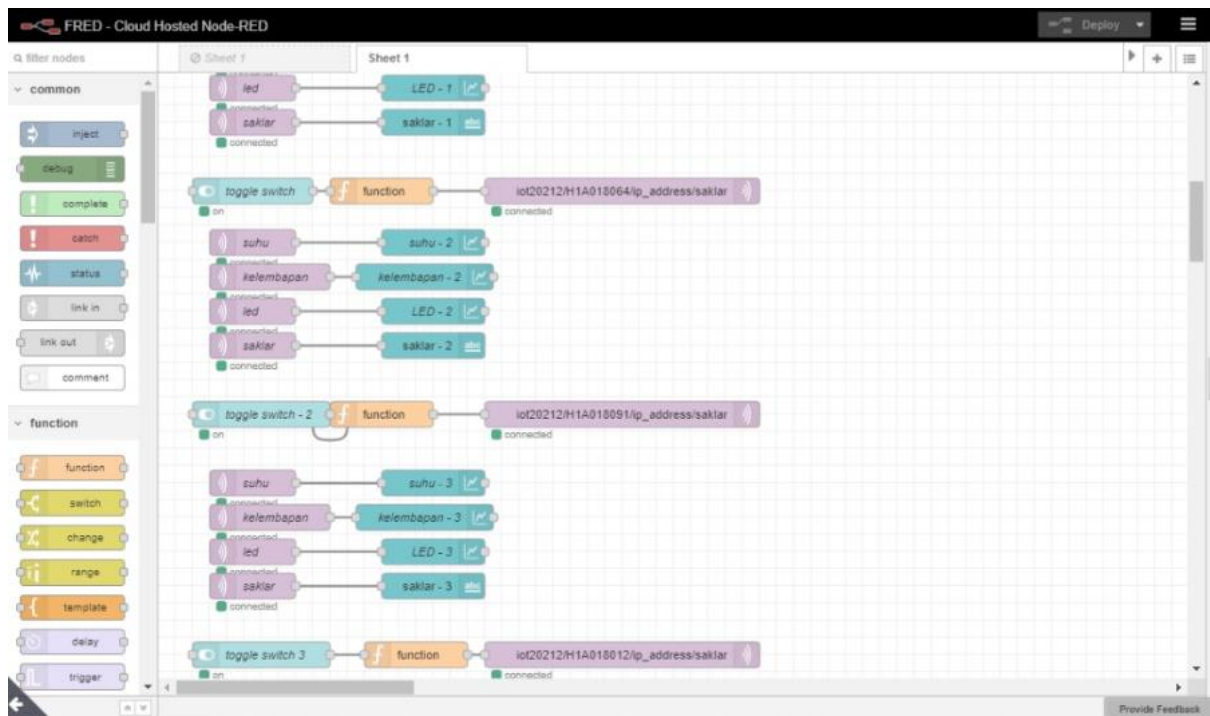
```
void lampu(){
  if(state_saklar==0&&Temp>28){
    waktu_led=waktu_led+millis()-waktu_perulangan;
    waktu_perulangan=millis();
  }
  else{
    waktu_led=100000;
  }
  if(waktu_led<60000){
    digitalWrite(led,0);
    Serial.println("LED Menyala");
    client.publish("iot20212/H1A018064/LED","1",bool(1));
  }
  if(waktu_led>60000){
    digitalWrite(led,1);
    Serial.println("LED Mati");
    client.publish("iot20212/H1A018064/LED","0",bool(1));
    waktu_led=0;
    waktu_perulangan=millis();
  }
}
```

Disini akan dijelaskan bahwa waktu nyala lampu tidak akan melebihi dari 1 menit sehingga apabila suhu yang dihasilkan pada waktu satu menit belum menghasilkan suhu dibawah 28 °C maka sistem akan mati sebentar dan akan menyalakan AC lagi hingga mendapatkan suhu dibawah 28°C .

9). Menambahkan prosedur perulangan pada sistem :

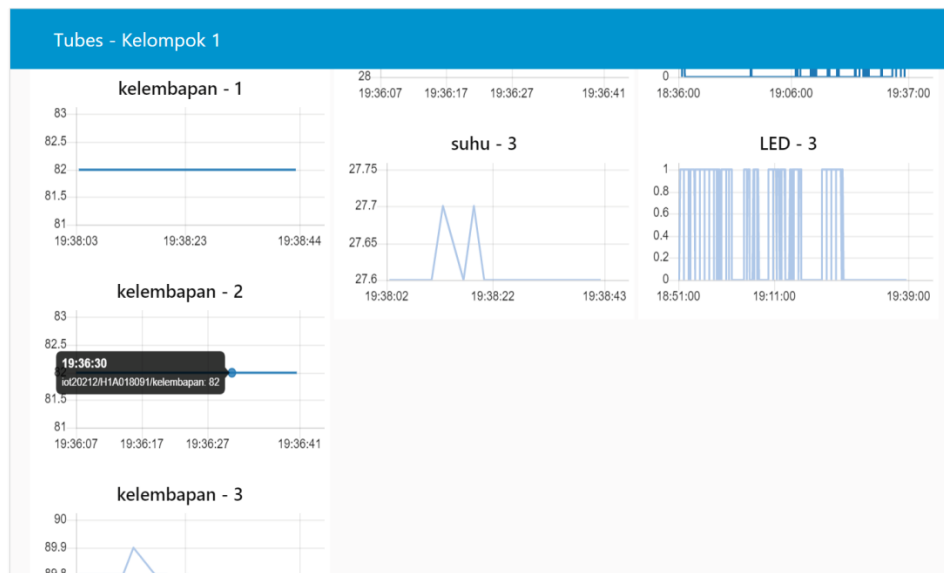
```
client.loop();
callback;
unsigned long now = millis();
if (now - lastMsg > 2000) {
  lampu();
  Serial.println(digitalRead(saklar));
  lastMsg = now;
  Temp=dht.readTemperature();
  Humidity=dht.readHumidity();
  snprintf (msg_LED, MSG_BUFFER_SIZE, "%d", !digitalRead(led));
  snprintf (msg_suhu, MSG_BUFFER_SIZE, "%f", Temp);
  snprintf (msg_kelembapan, MSG_BUFFER_SIZE, "%f", Humidity);
  client.publish("iot20212/H1A018064/ipaddress",msg_ipaddress,bool(1));
  client.publish("iot20212/H1A018064/chipid",msg_chipid,bool(1));
}
```

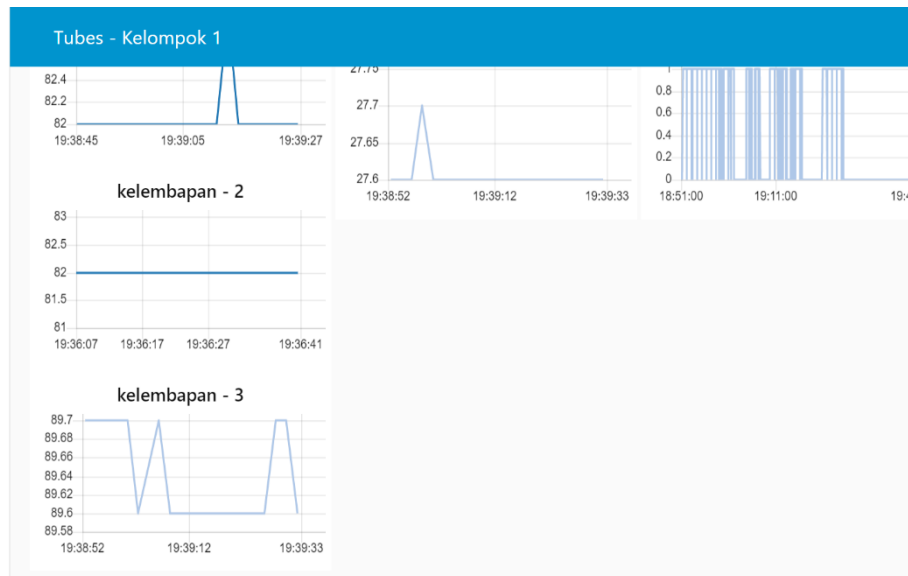
10). Untuk gambar Node red yang ditampilkan :



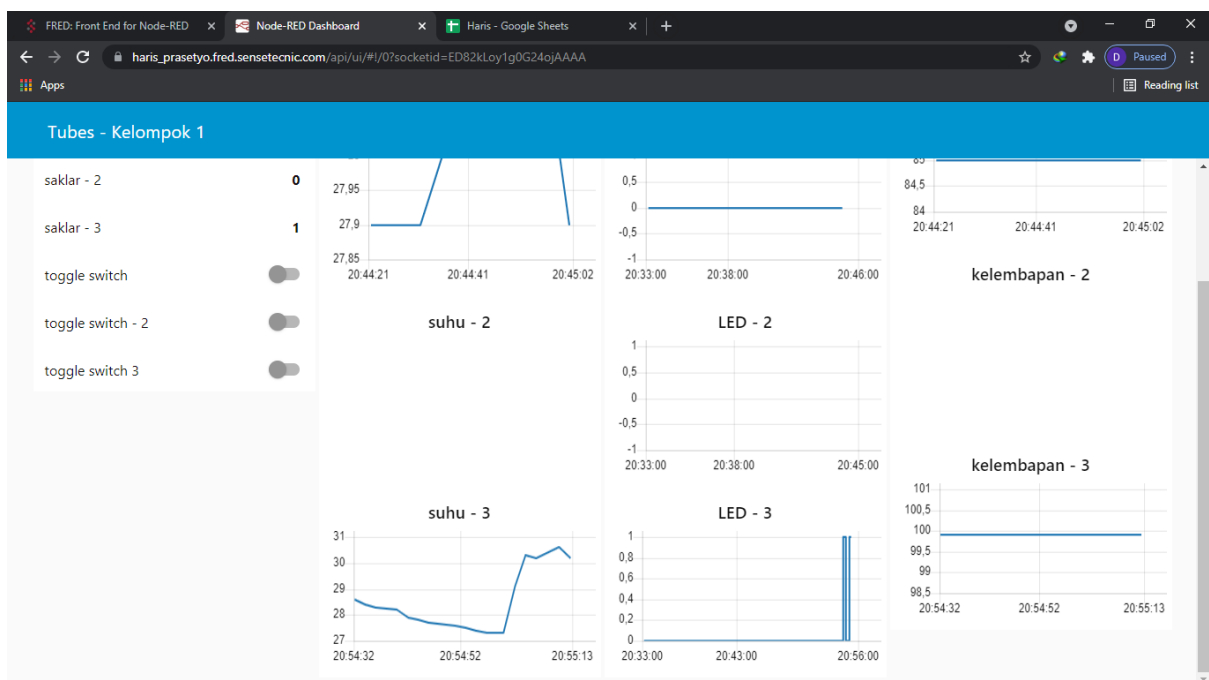
Hasil :

1). Tampilan pada interface WEB Browser.



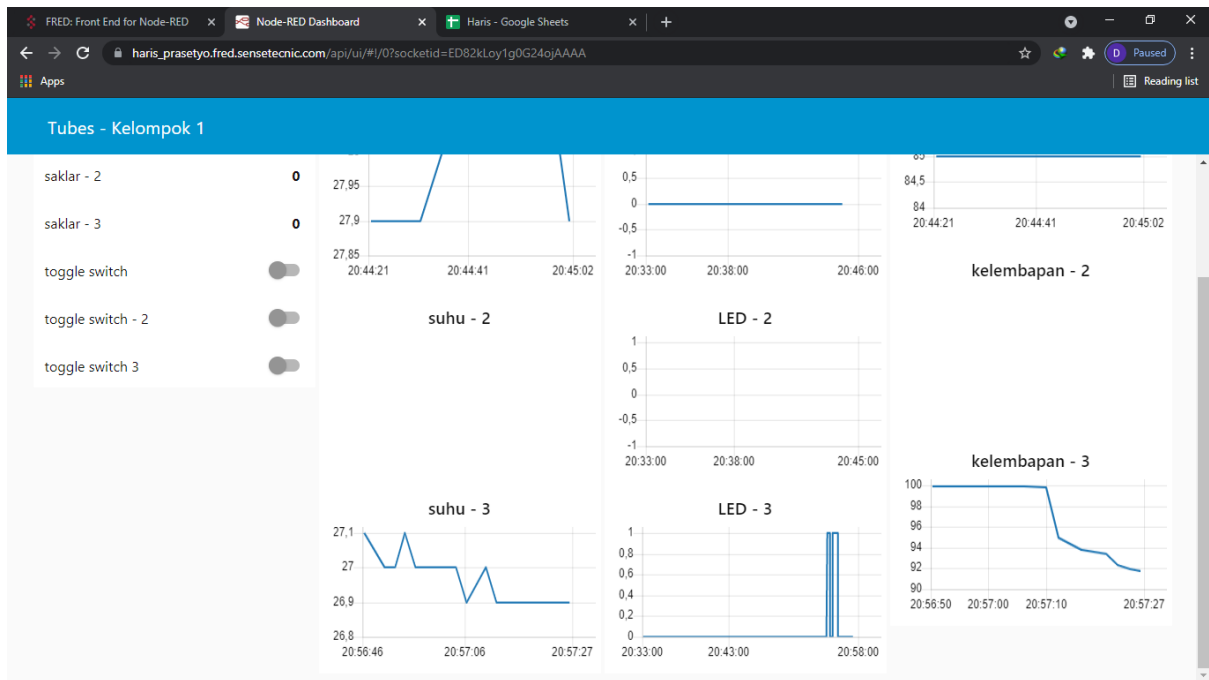


2). Tampilan pada interface WEB Browser ketika suhu di atas 28 °C maka ada informasi AC menyala.



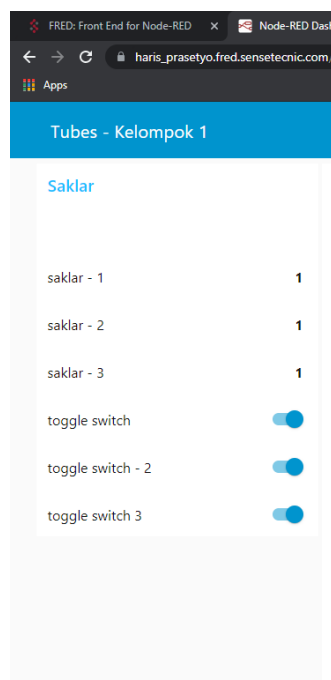
Pengujian di suhu 3 dan saklar 3 (ketika suhu 3 di atas 28°C saklar 3 nyala)

3). Tampilan pada interface WEB Browser ketika suhu bawah atau sama dengan 28 °C maka ada informasi AC mati.

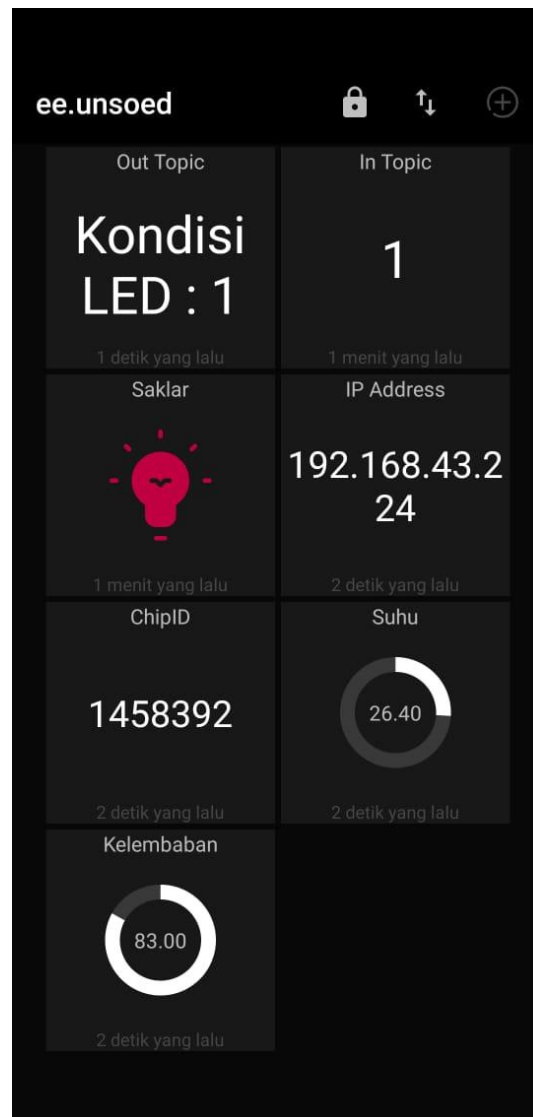


Pengujian di suhu 3 dan saklar 3 (ketika suhu di bawah 28°C saklar 3 mati)

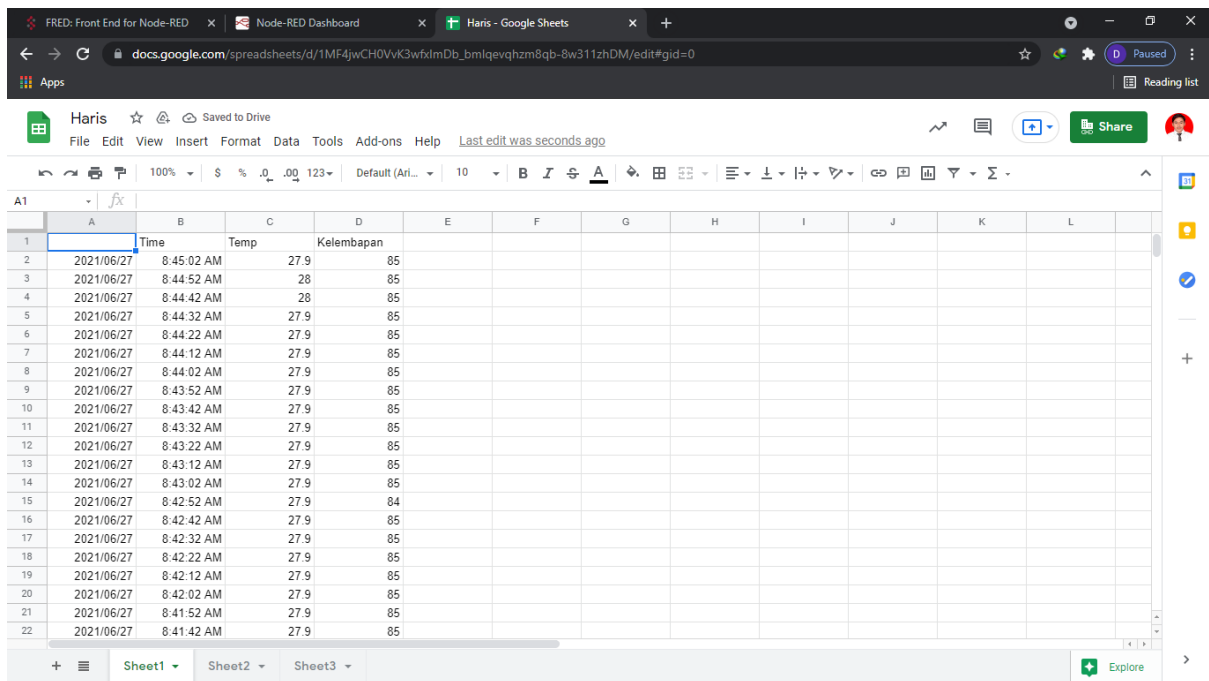
4). Tampilan pada interface WEB Browser Ketika masing – masing LED dinyalakan.



5). Tampilan pada interface MQTT Dash Ketika LED dinyalakan.

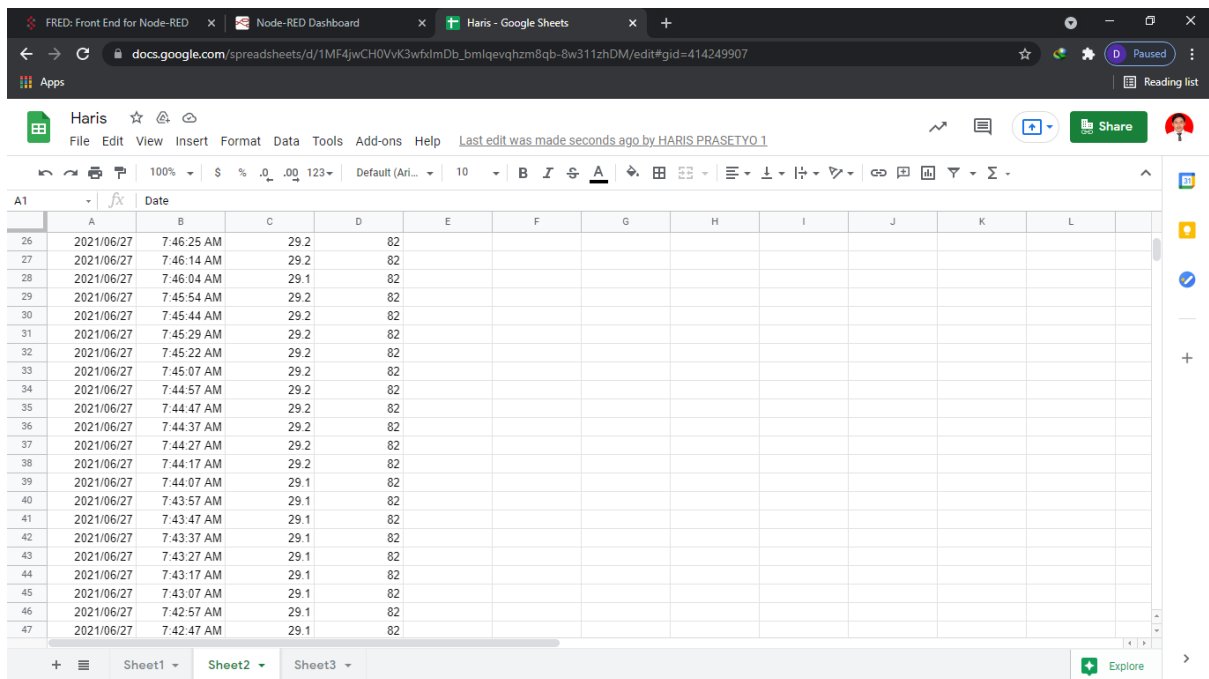


6). Tampilan pada Google spreadsheet.



A	B	C	D	E	F	G	H	I	J	K	L
1	Time	Temp	Kelembapan								
2	2021/06/27	8:45:02 AM	27.9	85							
3	2021/06/27	8:44:52 AM	28	85							
4	2021/06/27	8:44:42 AM	28	85							
5	2021/06/27	8:44:32 AM	27.9	85							
6	2021/06/27	8:44:22 AM	27.9	85							
7	2021/06/27	8:44:12 AM	27.9	85							
8	2021/06/27	8:44:02 AM	27.9	85							
9	2021/06/27	8:43:52 AM	27.9	85							
10	2021/06/27	8:43:42 AM	27.9	85							
11	2021/06/27	8:43:32 AM	27.9	85							
12	2021/06/27	8:43:22 AM	27.9	85							
13	2021/06/27	8:43:12 AM	27.9	85							
14	2021/06/27	8:43:02 AM	27.9	85							
15	2021/06/27	8:42:52 AM	27.9	84							
16	2021/06/27	8:42:42 AM	27.9	85							
17	2021/06/27	8:42:32 AM	27.9	85							
18	2021/06/27	8:42:22 AM	27.9	85							
19	2021/06/27	8:42:12 AM	27.9	85							
20	2021/06/27	8:42:02 AM	27.9	85							
21	2021/06/27	8:41:52 AM	27.9	85							
22	2021/06/27	8:41:42 AM	27.9	85							

Sheet1



A	B	C	D	E	F	G	H	I	J	K	L
26	2021/06/27	7:46:25 AM	29.2	82							
27	2021/06/27	7:46:14 AM	29.2	82							
28	2021/06/27	7:46:04 AM	29.1	82							
29	2021/06/27	7:45:54 AM	29.2	82							
30	2021/06/27	7:45:44 AM	29.2	82							
31	2021/06/27	7:45:29 AM	29.2	82							
32	2021/06/27	7:45:22 AM	29.2	82							
33	2021/06/27	7:45:07 AM	29.2	82							
34	2021/06/27	7:44:57 AM	29.2	82							
35	2021/06/27	7:44:47 AM	29.2	82							
36	2021/06/27	7:44:37 AM	29.2	82							
37	2021/06/27	7:44:27 AM	29.2	82							
38	2021/06/27	7:44:17 AM	29.2	82							
39	2021/06/27	7:44:07 AM	29.1	82							
40	2021/06/27	7:43:57 AM	29.1	82							
41	2021/06/27	7:43:47 AM	29.1	82							
42	2021/06/27	7:43:37 AM	29.1	82							
43	2021/06/27	7:43:27 AM	29.1	82							
44	2021/06/27	7:43:17 AM	29.1	82							
45	2021/06/27	7:43:07 AM	29.1	82							
46	2021/06/27	7:42:57 AM	29.1	82							
47	2021/06/27	7:42:47 AM	29.1	82							

Sheet 2

Lampiran :

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include "C:\Users\lenovo\Documents\PlatformIO\Projects\Tubes\.pio\libdeps\nodemcu2\DHT sensor library\DHT.h"
#include "C:\Users\lenovo\Documents\PlatformIO\Projects\Tubes\.pio\libdeps\nodemcu2\HTTPSRedirect\HTTPSRedirect.h"
#include <PubSubClient.h>
#include <WiFiClient.h>
#include "C:\Users\lenovo\Documents\PlatformIO\Projects\Tubes\.pio\libdeps\nodemcu2\DHT sensor library\DHT_U.h"
#define DHTTYPE DHT11
#include <ESP8266HTTPUpdateServer.h>

const char* ssid = "Mi Phone";
const char* password = "qwerty67";
const char* host = "esp8266";
const char* mqtt_server = "ee.unsoed.ac.id";
const char *GScriptId = "AKfycbwm0-Vpet0YmfHmsWZLLiZHQIFp7Y1orV5vsypDLDXWgezGG9zsokeaYaRhschw2tN";
String payload_base = "{\"command\": \"insert_row\", \"sheet_name\": \"Sheet1\", \"values\": \"\"";
String payload = "";
const char* host_gsheel = "script.google.com";
const int httpsPort = 443;
const char* fingerprint = "";
String url = String("/macros/s/") + GScriptId + "/exec?cal";
HTTPSRedirect* client_https = nullptr;
unsigned long waktu_led,waktu_perulangan;
float Temp;
float Humidity;

WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
unsigned long lastgsheet = 0;
#define MSG_BUFFER_SIZE (50)
char msg_LED[MSG_BUFFER_SIZE],msg_suhu[MSG_BUFFER_SIZE],msg_kelembapan[MSG_BUFFER_SIZE],msg_ipaddress[MSG_BUFFER_SIZE],msg_chipid[MSG_BUFFER_SIZE];

ESP8266HTTPUpdateServer httpUpdater;
DHT dht(GPIO_ID_PIN(D5),DHTTYPE);
ESP8266WebServer server(80);
```

```

int state_saklar;
const int saklar = D7;
const int led = LED_BUILTIN_AUX;
String message(float temp,float humid){
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=devicewidth, initial-
scale=1.0, user-scalable=no\">\n";
    ptr += "<title>Proyek dengan ESP8266</title>\n";
    ptr += "<style>html { font-
family: Helvetica; display: inlineblock; margin: 0px auto; text-
align: center;}\n";
    ptr += "body{margin-
top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} \n";
    ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;} \n";
    ptr += "</style>\n";
    ptr += "</head>\n";
    ptr += "<body>\n";
    ptr += "<div id=\"identitas\">\n";
    ptr += "<h1>Haris Prasetyo</h1>\n";
    ptr += "<p>NIM: H1A018064 </p>";
    ptr += "<p>Jurusan Teknik Elektro FT Unsoed </p>";
    ptr += "</div>\n";
    ptr += "<div id=\"webpage\">\n";
    ptr += "<h1>Pembacaan suhu dan kelembaban</h1>\n";
    ptr += "<p>Suhu: ";
    ptr += temp;
    ptr += " <sup>o</sup>C</p>";
    ptr += "<p>Kelembaban: ";
    ptr += humid;
    ptr += " %</p>";
    ptr += "</div>\n";
    ptr += "<div id=\"saklar\">\n";
    ptr += "<h1>LED : ";
    if (digitalRead(led)==0){
        ptr+="Menyala</h1>\n";
    }
    else{
        ptr+="Mati</h1>\n";
    }
    ptr += "<a href=\"/led_hidup\"><button>ON</button></a>";
    ptr += "<a href=\"/led_mati\"><button>OFF</button></a>";
    ptr += "</div>\n";
    ptr += "</body>\n";
    ptr += "</html>\n";
    return ptr;
}

```

```

void ICACHE_RAM_ATTR saklar_berubah(){
    if(digitalRead(saklar)==1){
        state_saklar=1;
        Serial.println("Saklar OFF");
    }
    else{
        Serial.println("Saklar ON");
        state_saklar=0;
    }
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Create a random client ID
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        // Attempt to connect
        if (client.connect(clientId.c_str())) {
            Serial.println("connected MQTT");
            // Once connected, publish an announcement...
            client.publish("iot20212/H1A018064/OutTopic", "hello world");
            // ... and resubscribe
            client.subscribe("iot20212/H1A018064/ip_address/saklar");
        }
        else {
            Serial.print("failed, rc=");
            Serial.print(client.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 1 was received as first character
    if ((char)payload[0] == '1') {
        //digitalWrite(LED_BUILTIN, LOW);
        state_saklar=0;
    }
}

```

```

    }
    else {
        //digitalWrite(led, HIGH);
        state_saklar=1;
    }
}

void handleRoot() {
    float Temp=dht.readTemperature();
    float Humidity=dht.readHumidity();
    server.send(200,"text/html",message(Temp,Humidity));
    delay(2000);
}

void handleNotFound(){
    delay(2000);
    digitalWrite(led, 1);
    String message = "File Not Found\n\n";
    message += "URI: ";
    message += server.uri();
    message += "\nMethod: ";
    message += (server.method() == HTTP_GET)? "GET": "POST";
    message += "\nArguments: ";
    message += server.args();
    message += "\n";
    for (uint8_t i=0; i<server.args(); i++){
        message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
    }
    server.send(404, "text/plain", message);
    digitalWrite(led, 0);
}

void lampu(){
    if(state_saklar==0&&Temp>28){
        waktu_led=waktu_led+millis()-waktu_perulangan;
        waktu_perulangan=millis();
    }
    else{
        waktu_led=100000;
    }
    if(waktu_led<60000){
        digitalWrite(led,0);
        Serial.println("LED Menyala");
        client.publish("iot20212/H1A018064/LED","1",bool(1));
    }
    if(waktu_led>60000){
        digitalWrite(led,1);
        Serial.println("LED Mati");
    }
}

```



```

        client.publish("iot20212/H1A018064/LED","0",bool(1));
        waktu_led=0;
        waktu_perulangan=millis();
    }
}

void setup(void){
    Serial.begin(115200);
    pinMode(saklar,INPUT);
    attachInterrupt(digitalPinToInterrupt(saklar), saklar_berubah, CHANGE);
    //attachInterrupt(saklar,saklar_berubah,CHANGE);
    dht.begin();
    pinMode(led, OUTPUT);
    digitalWrite(led, 0);
    state_saklar=digitalRead(D7);
    WiFi.mode(WIFI_AP_STA);
    WiFi.begin(ssid, password);
    Serial.println("");
    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    if (MDNS.begin(host)) {
        Serial.println("\nMDNS responder started");
    }
    httpUpdater.setup(&server);
    client_https = new HTTPSRedirect(httpsPort);
    client_https->setInsecure();
    client_https->setPrintResponseBody(true);
    client_https->setContentTypeHeader("application/json");
    Serial.print("Connecting to ");
    Serial.println(host_gsheets);

    bool flag = false;
    for (int i=0; i<5; i++){
        int retval = client_https->connect(host_gsheets, httpsPort);
        if (retval == 1){
            flag = true;
            Serial.println("Connected");
            break;
        }
        else
            Serial.println("Connection failed. Retrying...");
    }
    if (!flag){
        Serial.print("Could not connect to server: ");
        Serial.println(host_gsheets);
    }
}

```

```

    return;
}
delete client_https;    // delete HTTPSRedirect object
client_https = nullptr; // delete HTTPSRedirect object

MDNS.addService("http", "tcp", 80);
Serial.printf("HTTPUpdateServer ready! Open http://%s.local/update in your browser\n", host);
ArduinoOTA.onStart([]() {
    String type;
    if (ArduinoOTA.getCommand() == U_FLASH) {
        type = "sketch";
    }
    else { // U_FS
        type = "filesystem";
    }
    Serial.println("Start updating " + type);
});
ArduinoOTA.onEnd([]() {
    Serial.println("\nEnd");
});
ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
});
ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) {
        Serial.println("Auth Failed");
    }
    else if (error == OTA_BEGIN_ERROR) {
        Serial.println("Begin Failed");
    }
    else if (error == OTA_CONNECT_ERROR) {
        Serial.println("Connect Failed");
    }
    else if (error == OTA_RECEIVE_ERROR) {
        Serial.println("Receive Failed");
    }
    else if (error == OTA_END_ERROR) {
        Serial.println("End Failed");
    }
});
ArduinoOTA.begin();
Serial.println("Ready");
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");

```

```

Serial.println(WiFi.localIP());
server.on("/", handleRoot);

server.on("/inline", [](){
    server.send(200, "text/plain", "this works as well");
});
server.on("/led_hidup", [](){
    server.send(200, "text/plain", "Led Hidup");
    digitalWrite(led,0);
    Serial.println("Led Hidup");
    if(digitalRead(led)==0){
        client.publish("iot20212/H1A018064/LED","1");
    }
    if(digitalRead(led)==1){
        client.publish("iot20212/H1A018064/LED","0");
    }
});
server.on("/led_mati", [](){
    server.send(200, "text/plain", "Led Padam");
    digitalWrite(led,1);
    Serial.println("Led Padam");
    if(digitalRead(led)==0){
        client.publish("iot20212/H1A018064/LED","1");
    }
    if(digitalRead(led)==1){
        client.publish("iot20212/H1A018064/LED","0");
    }
});
server.onNotFound(handleNotFound);
server.begin();
Serial.println("HTTP server started");
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
snprintf (msg_chipid, MSG_BUFFER_SIZE, "Chip ID : %d", ESP.getChipId());
Serial.println(msg_chipid);
String myip = WiFi.localIP().toString();
myip.toCharArray(msg_ipaddress,MSG_BUFFER_SIZE);
client.publish("iot20212/H1A018064/ipaddress",msg_ipaddress,bool(1));
//client.publish("iot20212/H1A018064/chipid",msg_chipid,bool(1));
}

int i=0;

void loop(void){
    server.handleClient();
    ArduinoOTA.handle();
    MDNS.update();
    if (!client.connected()) {

```

```

    reconnect();
}
client.loop();
callback;
unsigned long now = millis();
if (now - lastMsg > 2000) {
    lampu();
    Serial.println(digitalRead(saklar));
    lastMsg = now;
    Temp=dht.readTemperature();
    Humidity=dht.readHumidity();
    snprintf (msg_LED, MSG_BUFFER_SIZE, "%d", !digitalRead(led));
    snprintf (msg_suhu, MSG_BUFFER_SIZE, "%f", Temp);
    snprintf (msg_kelembapan, MSG_BUFFER_SIZE, "%f", Humidity);
    client.publish("iot20212/H1A018064/ipaddress",msg_ipaddress,bool(1));
    client.publish("iot20212/H1A018064/chipid",msg_chipid,bool(1));
    /*if(digitalRead(led)==0){
        client.publish("iot20212/H1A018064/LED","1",bool(1));
    }
    if(digitalRead(led)==1){
        client.publish("iot20212/H1A018064/LED","0",bool(1));
    }*/
    Serial.print("\n\nPublish message: \n");
    Serial.println(msg_LED);
    Serial.println(msg_suhu);
    Serial.println(msg_kelembapan);
    //client.publish("")
    client.publish("iot20212/H1A018064/suhu",msg_suhu);
    client.publish("iot20212/H1A018064/kelembapan",msg_kelembapan);

}
if(now-lastgsheet>10000){
    lastgsheet = millis();
    static bool flag = false;
    if (!flag){
        client_https = new HTTPSRedirect(httpsPort);
        client_https->setInsecure();
        flag = true;
        client_https->setPrintResponseBody(true);
        client_https->setContentTypeHeader("application/json");
    }
    if (client_https != nullptr){
        if (!client_https->connected()){
            client_https->connect(host_gsheel, httpsPort);
        }
    }
    else{
        Serial.println("Error creating client object!");
    }
}

```

```
}

payload = payload_base + "\"" + Temp + "," + Humidity + "\"";
Serial.println("Publishing data...");
Serial.println(payload);
if(client_https->POST(url, host_gsheat, payload)){
    // do stuff here if publish was successful
}
else{
    // do stuff here if publish was not successful
    Serial.println("Error while connecting");
}
}
}
```