LAMPIRAN

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <WiFiUdp.h>
#include <NTPClient.h>

//untuk ambil data ke firebase
// #include <Firebase_ESP_Client.h>
#include <FirebaseESP8266.h>

#include <addons/TokenHelper.h>
#include "DHT.h"
const char* ssid = "iPhone";
const char* password = "senyumdulu";
#define BOTtoken "6782600378:AAFuQekVR9FfXvEcRkbEDcmf-9QKIvtDWQ0"
#define CHAT_ID "5182511361"

const long utcOffsetInSeconds = 25200;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, "asia.pool.ntp.org",utcOffsetInSeconds);


//PIN DI NODEMCU
#define led     D4
#define led1    D1
#define led2    D2
#define flame   D6
#define mq2     A0

#define DHTPIN  D5      // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

// Insert Firebase project API Key
#define API_KEY "AIzaSyARbnYScMz1aB8rZiG_0qr3Weu7CyQi2Dw"
#define DATABASE_URL "https://monitoringserver-f0680-default-rtdb.firebaseio.com"
// Insert Authorized Email and Corresponding Password
#define USER_EMAIL "aboysetiawan13@gmail.com"
#define USER_PASSWORD "enggataulupa13"

float h,t;
int api,asap;
```

```
String statusAsap, statusApi, statusServer,statusSuhu,statusLedMerah,statusLedKuning;

unsigned long previousMillis  = 0;

const long interval = 1000;



X509List cert(TELEGRAM_CERTIFICATE_ROOT);

WiFiClientSecure client;

UniversalTelegramBot  bot(BOTtoken, client);


// Checks for new messages every 1 second.

int botRequestDelay = 1000;

unsigned long lastTimeBotRan;

unsigned long lastFirebaseUpdate;


const int ledPin = 2;

bool ledState = LOW;

bool isTelegram  = false;


//Define Firebase  Data objects

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig configF;
```

```
FirebaseJson json;


String suhuPath = "/suhu";
String kelembabanPath = "/kelembaban";
String statusAsapPath = "/status_asap";
String statusApiPath = "/status_api";
String ledMerahPath = "/led_merah";
String ledKuningPath = "/led_kuning";
String statusSuhuPath = "/status_suhu";
String statusServerPath = "/status_server";


// Handle what happens when you receive new messages
void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));


  for (int i=0; i<numNewMessages; i++) {
    // Chat id of the requester
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }
```

```
// Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);


    String from_name = bot.messages[i].from_name;


    if (text == "/start") {
      String welcome = "Welcome, " + from_name + ".\n";
      welcome += "Ketik perintah dibawah ini untuk melihat kondisi Server.\n\n";
      welcome += "/check-server \n";
      bot.sendMessage(chat_id, welcome, "");
    }


    if(text == "/check-server"){
      // isTelegram  = true;
      String dataServer = "";
      dataServer += "Suhu Server : " + String(t) + " °C" + "\n";
      dataServer += "Tingkat Kelembaban : " + String(h) + " %" + "\n";
      dataServer += statusSuhu + ".\n";
      dataServer += statusAsap + ".\n";
      dataServer += statusApi + ".\n";
      dataServer += statusServer + ".\n";
      bot.sendMessage(chat_id, dataServer,"");
```

```
  }
 }
}

void setup() {
 configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
 client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

 pinMode(led, OUTPUT);
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(flame, INPUT);
 Serial.begin(115200);
 Serial.println("starting");
 dht.begin();

 timeClient.begin();

 digitalWrite(led1,1);
 digitalWrite(led2,1);
 delay(1000);
 digitalWrite(led1,0);
 digitalWrite(led2,0);
```

```
  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  // Print ESP32 Local IP Address
  Serial.println(WiFi.localIP());
  initFirebase();
}


void initFirebase(){
  //Firebase
  // Assign the api key
  configF.api_key = API_KEY;
   /* Assign the RTDB URL (required) */
  configF.database_url = DATABASE_URL;
  //Assign the user sign in credentials
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  //Assign the callback function for the long running token generation task
  configF.token_status_callback = tokenStatusCallback; //see
addons/TokenHelper.h
```

```
  Firebase.begin(&configF, &auth);
  Firebase.reconnectWiFi(true);
  // fbdo.setResponseSize(4096);

}

String getCurrentDate(){
  unsigned long epochTime = timeClient.getEpochTime();
  time_t rawtime = epochTime;
  struct tm *ptm = gmtime(&rawtime);
  int monthDay = ptm->tm_mday;
  int currentMonth = ptm->tm_mon+1;
  int currentYear = ptm->tm_year+1900;

  int hours = timeClient.getHours();
  int minutes = timeClient.getMinutes();

  String currentDate = String(currentYear) + "-" + String(currentMonth) + "-" +
String(monthDay);
  String currentTime = String(hours) + ":" + String(minutes) + ":" + "00";
  String currentDateTime = currentDate + " "+ currentTime;
  return currentDateTime;
 }
```

```
void sendData(String dateTime){
  String parentPath = "/data/" + dateTime + "/result";
  String result = String(t) + "|" + String(h) + "|" + String(statusAsap) + "|" + String(statusApi);
  Firebase.setString(fbdo, parentPath, result);
}

void getData() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    digitalWrite(led,digitalRead(led)^1);

    h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    t = dht.readTemperature();
    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t)) {
      Serial.println("dht sensor error!");
    }
    api = digitalRead(flame);
    asap = analogRead(mq2);
```

```
if(api==0){
    digitalWrite(led1,1);
    statusApi = "Ada Api !";
    statusServer = "Waspada Kondisi Server !";
    statusLedMerah = "Nyala";
  }
  else{
    digitalWrite(led1,0);
    statusApi = "Tidak Ada Api";
    statusServer = "Kondisi Server Aman";
    statusLedMerah = "Mati";
  }
  if(asap>350){  // batas asap
    digitalWrite(led2,1);
    statusAsap = "Terdeteksi Kemunculan Asap !";
    statusLedKuning = "Nyala";
  }
  else{
    digitalWrite(led2,0);
    statusAsap ="Tidak ada Asap berbahaya";
    statusLedKuning = "Mati";
  }
```

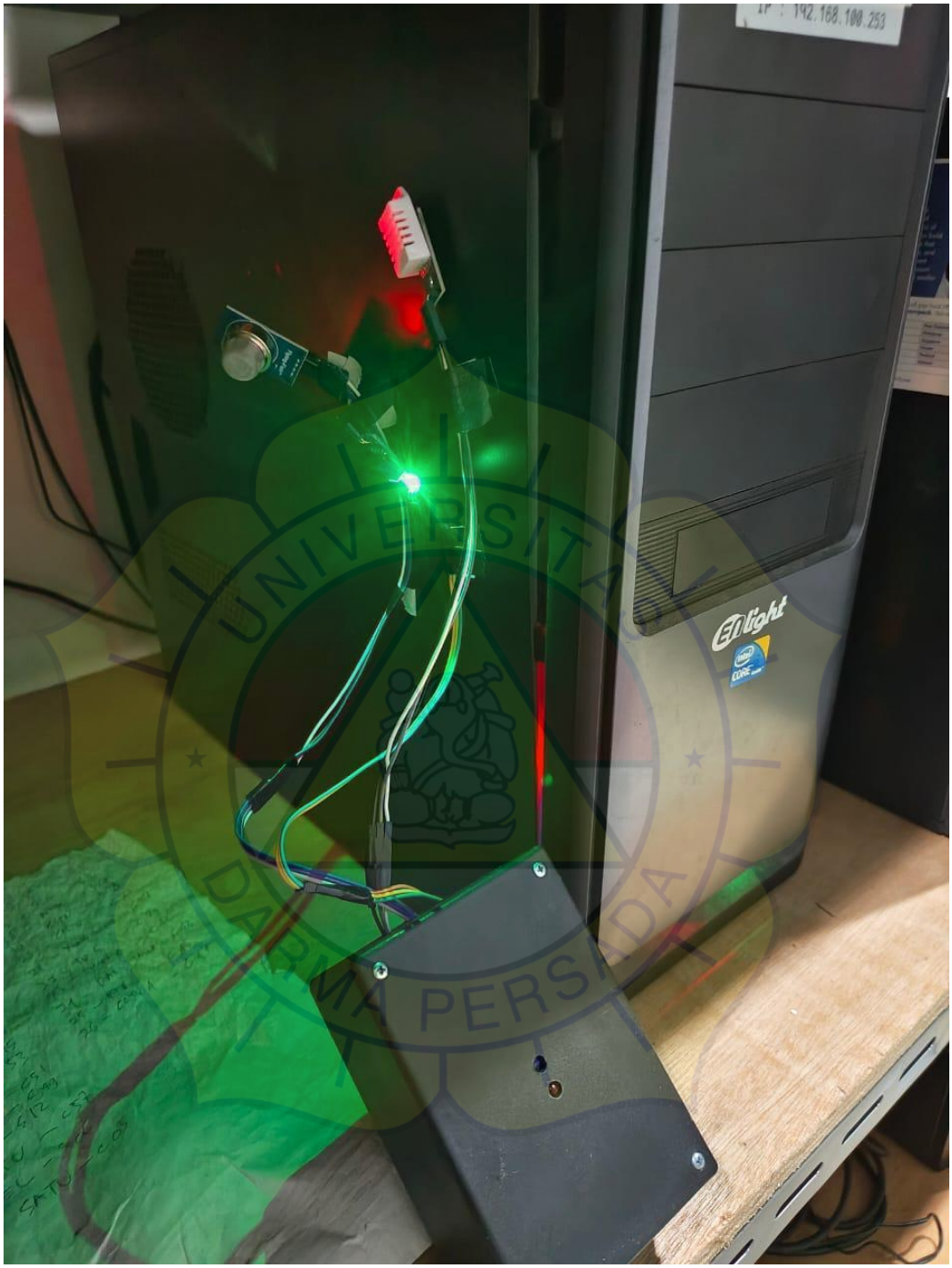
```
if(t>35){ // batas suhu
    // Serial.println("SUHU NAIK");
    statusSuhu = "Suhu Naik !";
   }
   else{
    // Serial.println("SUHU NORMAL");
    statusSuhu = "Suhu Rendah";
   }

    Firebase.setString(fbdo, "/dashboard/suhu", String(t)); // Asumsi ini adalah non-blocking
    Firebase.setString(fbdo, "/dashboard/kelembaban", String(h)); // Asumsi ini adalah non-blocking
    Firebase.setString(fbdo, "/dashboard/status_asap", String(asap)); // Asumsi ini adalah non-blocking
    Firebase.setString(fbdo, "/dashboard/status_api", String(api)); // Asumsi ini adalah non-blocking

  }
}
void loop() {
  timeClient.update();

  getData();
```

```
  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  Serial.println(numNewMessages);
  if (numNewMessages > 0) {
    handleNewMessages(numNewMessages);
    // numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    numNewMessages = 0;
    Serial.println("Send  to Telegram");
  }else{
    if (millis()  - lastFirebaseUpdate > 10000) {
      String parentPath = "/data/" + getCurrentDate() + "/result";
      String result = String(t) + "|" + String(h) + "|" + String(api) + "|" + String(asap);
      Firebase.setString(fbdo, parentPath, result); // Asumsi ini adalah non-blocking
      Serial.println(result);
      lastFirebaseUpdate = millis();
    }
  }
}
```


192.168.100.253
EDlight


SERVER 3 "PRINT SERVER"
IP : 192.168.100.253