

## LAMPIRAN

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
#include <MFRC522.h>
#include <ArduinoJson.h>
#include <BlynkSimpleEsp8266.h> //---ini kebutuhan blynk
#include "CTBot.h" //---kebutuhan tele

#include "Adafruit_Fingerprint.h"
#include "SoftwareSerial.h"
SoftwareSerial mySerial(15, 14);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

#define SS_PIN D2 // sda pin D2
#define RST_PIN D1 // RST (flash) pin D2

#define relay D4 //Relay Pin D4
#define buzzer D3 //buzzer pin D8
#define LED_R D0 //LED R (RGB)
#define LED_G 3 //LED G (RGB) (pin RX0)
#define LED_B D8 //LED B (RGB)
//byte buzzer = 2; //pin D0
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
MFRC522::MIFARE_Key key;

const char* ssid = "iPhone Asrawan"; //Your Wifi SSID
const char* password = "bambangsutisna"; //Wifi Password

//BLYNK CONFIG
#define BLYNK_TEMPLATE_ID "TMPLaqn2_Oek"
#define BLYNK_DEVICE_NAME "Open Door"
#define BLYNK_AUTH_TOKEN "UE9LVo_gk3rhwKuYkwdfZ56S4X5UeQnC"
#define BLYNK_PRINT Serial
char authb[] = BLYNK_AUTH_TOKEN;
char ssidb[] = "iPhone Asrawan";
char passb[] = "bambangsutisna";
//BLYNK CONFIG

//TELE CONFIG
CTBot myBot;
String ssidt = "iPhone Asrawan";
String passt = "bambangsutisna";
String tokent = "5851001257:AAF7e_4krf3xFFIGqSihgTdrdUtKhLVfEwv";
const int idt = 1722529437;
//TELE CONFIG;

WiFiClientSecure client;
HTTPClient sender;
void(* ku_reset) (void) = 0;

byte readCard[4];
uint8_t successRead;
String UIDCard;
String IDFinger;
```

```

int inProcess=0;
byte i;
String ID_TAG;

String paramLogic;

void Red_ON(){
  digitalWrite(LED_R,HIGH);
  digitalWrite(LED_G,LOW);
  digitalWrite(LED_B,LOW);
}
//=====================================================
void Green_ON(){
  digitalWrite(LED_R,LOW);
  digitalWrite(LED_G,HIGH);
  digitalWrite(LED_B,LOW);
}
//=====================================================
void Blue_ON(){
  digitalWrite(LED_R,LOW);
  digitalWrite(LED_G,LOW);
  digitalWrite(LED_B,HIGH);
}
}
BLYNK_WRITE(V0){
  Serial.print("Done Blynk.... ");
  Green_ON();
  digitalWrite(relay,LOW);
  delay(5000);
  digitalWrite(relay,HIGH);
  mfr522.PCD_Init();
  if(param.asInt() == 1){
    Blynk.virtualWrite(V0, 0);
  } else {
  }
}
}
void setup() {
  Serial.begin(115200);
  finger.begin(57600);
  delay(200);
  pinMode(buzzer,OUTPUT);
  pinMode(relay,OUTPUT);//relay
  pinMode(LED_R,OUTPUT);
  pinMode(LED_G,OUTPUT);
  pinMode(LED_B,OUTPUT);
  digitalWrite(relay,HIGH);
  digitalWrite(LED_R,LOW);
  digitalWrite(LED_G,LOW);
  digitalWrite(LED_B,LOW);
  delay(1000);
  mfr522.PCD_Init();
  Serial.println("Baca RFID Card");
  Serial.println("Tap kartu RFID ...");
  ConnectWIFI();
  ConnectTELE();
  myBot.sendMessage(idt, "Telegram sudah terkoneksi");
  Serial.println("Pesan Terkirim");
  postMessage("Alat Siaap");
}

```

```

    Blynk.begin(authb,ssidb,passb,"blynk.cloud",80);
}

void loop() {
  //delay(0);
  Blynk.run();
  Blue_ON();
  if(!mfr522.PICC_IsNewCardPresent() || !mfr522.PICC_ReadCardSerial()){

    return;
  }
  getID();
  delay(1000);
}
//-----RFID-----//
void getID() {

  UIDCard = "";
  Serial.println("Scanned PICC's UID:");

  for ( uint8_t i = 0; i < mfr522.uid.size; i++) {
    UIDCard += String(mfr522.uid.uidByte[i], HEX);
  }
  UIDCard.toUpperCase(); //Capital
  Serial.print("UID:");
  Serial.println(UIDCard);
  //Serial.println(F("End Reading"));
  digitalWrite(buzzer, HIGH);delay(200);
  digitalWrite(buzzer, LOW);delay(200);
  digitalWrite(buzzer, HIGH);delay(200);
  digitalWrite(buzzer, LOW);
  getParameter();
  storeData(); //store data to DB
  //delay(2000);

}
void postMessage(String message){
  ConnectWiFi(); //check wifi connection

  String address, first_name;
  String payload;
  client.setInsecure();
  client.connect("https://myskripsi.pancaran-group.co.id/api/access/set-message", 443);

  Serial.print("Client Connect ... api ");
  Serial.println("https://myskripsi.pancaran-group.co.id/api/access/set-message");

  if (sender.begin(client, "https://myskripsi.pancaran-group.co.id/api/access/set-
message"))
  {
    Serial.print("Client Connected ");

    // Serialize JSON document
    sender.addHeader("Content-Type", "application/x-www-form-urlencoded");
    String httpRequestData = "message="+message;

    int httpCode = sender.POST(httpRequestData);

```

```

if (httpCode > 0)
{
  if (httpCode == HTTP_CODE_OK)
  {
    payload = sender.getString();
    Serial.println("\nHTTP_CODE_OK");
    Serial.println(payload);
  }
  else
  {
    Serial.printf("HTTP-Error: ");
  }
}
}
else
{
  Serial.printf("Other HTTP error");
  sender.end();
}

const size_t capacity = JSON_OBJECT_SIZE(10) + 100; //simulate your JSON data
https://arduinojson.org/v6/assistant/
DynamicJsonDocument doc(capacity);

// Deserialize the JSON document
DeserializationError error = deserializeJson(doc, payload);

// Test if parsing succeeds.
if (error) {
  Serial.print(F("deserializeJson() failed: "));
  Serial.println(error.c_str());
}
}

void storeData(){
  ConnectWiFi(); //check wifi connection

  String address, first_name;

  String payload;
  client.setInsecure();
  client.connect("https://myskripsi.pancaran-group.co.id/api/access/create-rfid", 443);

  Serial.print("Client Connect create rfid ... ");
  Serial.println(" https://myskripsi.pancaran-group.co.id/api/access/create-
rfid?rfid="+UIDCard);

  if (sender.begin(client, "https://myskripsi.pancaran-group.co.id/api/access/create-
rfid?rfid="+UIDCard))
  {
    Serial.print("Client Connected");

    // Serialize JSON document
    int httpCode = sender.GET();

    if (httpCode > 0)

```

```

{
  if (httpCode == HTTP_CODE_OK)
  {
    payload = sender.getString();
    Serial.println("\nHTTP_CODE_OK");
    Serial.println(payload);
  }
  else
  {
    Serial.printf("HTTP-Error: ");
  }
}
}
else
{
  Serial.printf("Other HTTP error");
  sender.end();
}

const size_t capacity = JSON_OBJECT_SIZE(10) + 100; //simulate your JSON data
https://arduinojson.org/v6/assistant/
DynamicJsonDocument doc(capacity);

// Deserialize the JSON document
DeserializationError error = deserializeJson(doc, payload);

// Test if parsing succeeds.
if (error) {
  Serial.print(F("deserializeJson() failed: "));
  Serial.println(error.c_str());
  Red_ON();
  digitalWrite(buzzer, HIGH);
  delay(500);
  digitalWrite(buzzer, LOW);
  ConnectTELE();
  myBot.sendMessage(idt, "Error Koneksi");
  return;
}

String message_res = doc["message"];
int status_res = doc["resultCode"];
boolean priority = doc["priority"];

//Print Data
if (status_res == 0){

  Serial.print("Gagal Masuk1");
  //myBot.sendMessage(idt, message_res);
  postMessage(message_res);
  Red_ON();
  digitalWrite(buzzer, HIGH);
  delay(500);
  digitalWrite(buzzer, LOW);
  //ConnectTELE();
}else if (status_res == -1){
  Serial.print("Gagal Masuk2");
  postMessage(message_res);
}

```

```

Red_ON();
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
//ConnectTELE();
}else{
  if (status_res == 1){
    Serial.print("Masuk1");
    if(paramLogic=="Banyak" && priority){
      //myBot.sendMessage(idt, message_res);
      postMessage(message_res);
      Green_ON();
      digitalWrite(relay,LOW);
      delay(5000);
      digitalWrite(relay,HIGH);
      mfrc522.PCD_Init();
    } else if (paramLogic=="Banyak" && !priority) {
      Serial.print("Gagal Masuk2");
      //myBot.sendMessage(idt, "Ruangan Sudah Melebihi Kuota");
      postMessage(message_res+" Tapi Ruangan Sudah Melebihi Limit");
      Red_ON();
      digitalWrite(buzzer, HIGH);
      delay(500);
      digitalWrite(buzzer, LOW);
    } else if(paramLogic=="Sedang"){
      //myBot.sendMessage(idt, message_res);
      postMessage(message_res);
      Green_ON();
      digitalWrite(relay,LOW);
      delay(5000);
      digitalWrite(relay,HIGH);
      mfrc522.PCD_Init();
    } else if(paramLogic=="Sedikit"){
      //myBot.sendMessage(idt, message_res);
      postMessage(message_res);
      Green_ON();
      digitalWrite(relay,LOW);
      delay(5000);
      digitalWrite(relay,HIGH);
      mfrc522.PCD_Init();
    } else if(paramLogic=="Kosong"){
      //myBot.sendMessage(idt, message_res);
      postMessage(message_res);
      Green_ON();
      digitalWrite(relay,LOW);
      delay(5000);
      digitalWrite(relay,HIGH);
      mfrc522.PCD_Init();
    }
  }

} else{
  Serial.print("Masuk2");
  Green_ON();
  digitalWrite(relay,LOW);
  delay(5000);
  digitalWrite(relay,HIGH);
}
}
}

```

```

//display.display();
//delay(3000);

}
void getParameter(){
  ConnectWiFi(); //check wifi connection

  String address, first_name;
  String payload;
  client.setInsecure();
  client.connect("https://myskripsi.pancaran-group.co.id/api/access/get-parameter", 443);

  Serial.print("Client Connect ...");

  if (sender.begin(client, "https://myskripsi.pancaran-group.co.id/api/access/get-
parameter"))
  {
    Serial.print("Client Connected");

    // Serialize JSON document
    int httpCode = sender.GET();

    if (httpCode > 0)
    {
      if (httpCode == HTTP_CODE_OK)
      {
        payload = sender.getString();
        Serial.println("\nHTTP_CODE_OK");
        Serial.println(payload);
      }
      else
      {
        Serial.printf("HTTP-Error: ");
      }
    }
  }
  else
  {
    Serial.printf("Other HTTP error");
    sender.end();
  }

  const size_t capacity = JSON_OBJECT_SIZE(4) + 70; //simulate your JSON data
  https://arduinojson.org/v6/assistant/
  DynamicJsonDocument doc(capacity);

  // Deserialize the JSON document
  DeserializationError error = deserializeJson(doc, payload);

  // Test if parsing succeeds.
  if (error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.c_str());
    Red_ON();
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    ConnectTELE();
  }
}

```

```

    myBot.sendMessage(idt, "Error Koneksi");
    return;
}

int status_res = doc["resultCode"];
int max = doc["maxOrang"];
int access = doc["banyakOrang"];

//Print Data
Serial.print(access,max);
if (status_res == 1){
    if(access>=max){
        paramLogic = "Banyak";
    } else if (access<max && access>=(max+1)/2){
        paramLogic = "Sedang";
    } else if (access<(max+1)/2 && access >= 1){
        paramLogic = "Sedikit";
    } else {
        paramLogic="Kosong";
    }

    Serial.print("Parameter");
    Serial.println(paramLogic);
}
}
//-----CONNECT WIFI-----//
void ConnectWIFI(){
    if(WiFi.status() != WL_CONNECTED){
        Serial.print("Attempting to connect to SSID: ");
        Serial.println(ssid);
        WiFi.begin(ssid, password);
        int i=0;
        int a=0;
        while(WiFi.status() != WL_CONNECTED){
            Serial.print(".");
            if (a==0){
                a=1;
            }
            }else{
                a=0;
            }
            Red_ON();
            delay(1000);
            ++i;
            if (i==30){
                i=0;
                Red_ON();
                Serial.println("\n Failed to Connect.");
                break;
            }
        }
        Serial.println("\n Connected!");
        Green_ON();
        delay(2000);
    }
}
void ConnectTELE(){

```

```
myBot.wifiConnect(ssid, passt);  
myBot.setTelegramToken(token);  
  
if (myBot.testConnection()) {  
  Serial.println("Koneksi Bagus");  
} else {  
  Serial.println("Koneksi Jelek");  
}  
}
```

