TUGAS SIB ARKATAMA MULTI SOLUSINDO  
IOT PRAKTIK

ID Kegiatan : 7582873

Nama : Evy Nur Imamah

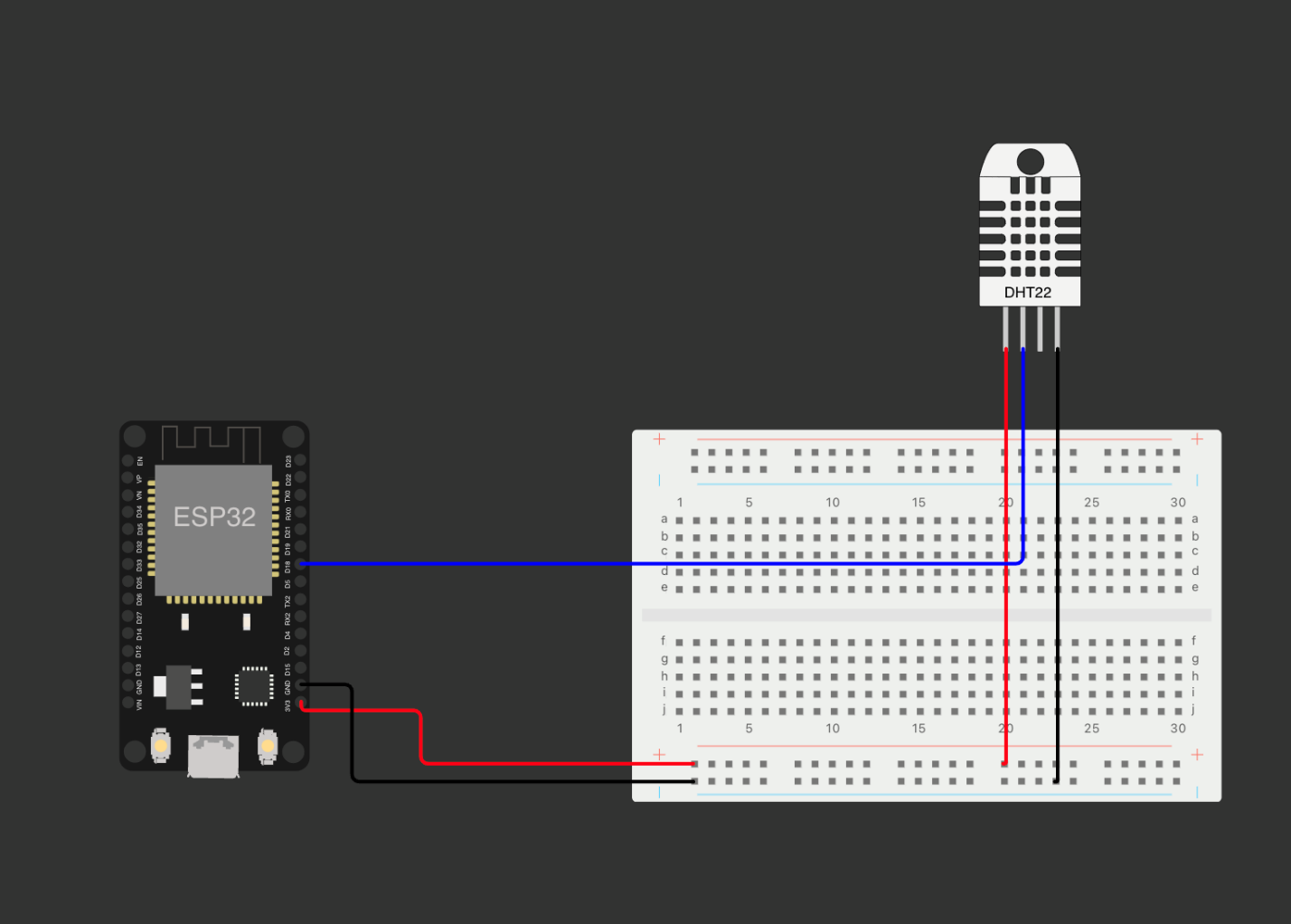
Kelas : IoT1

**TOOLS!**

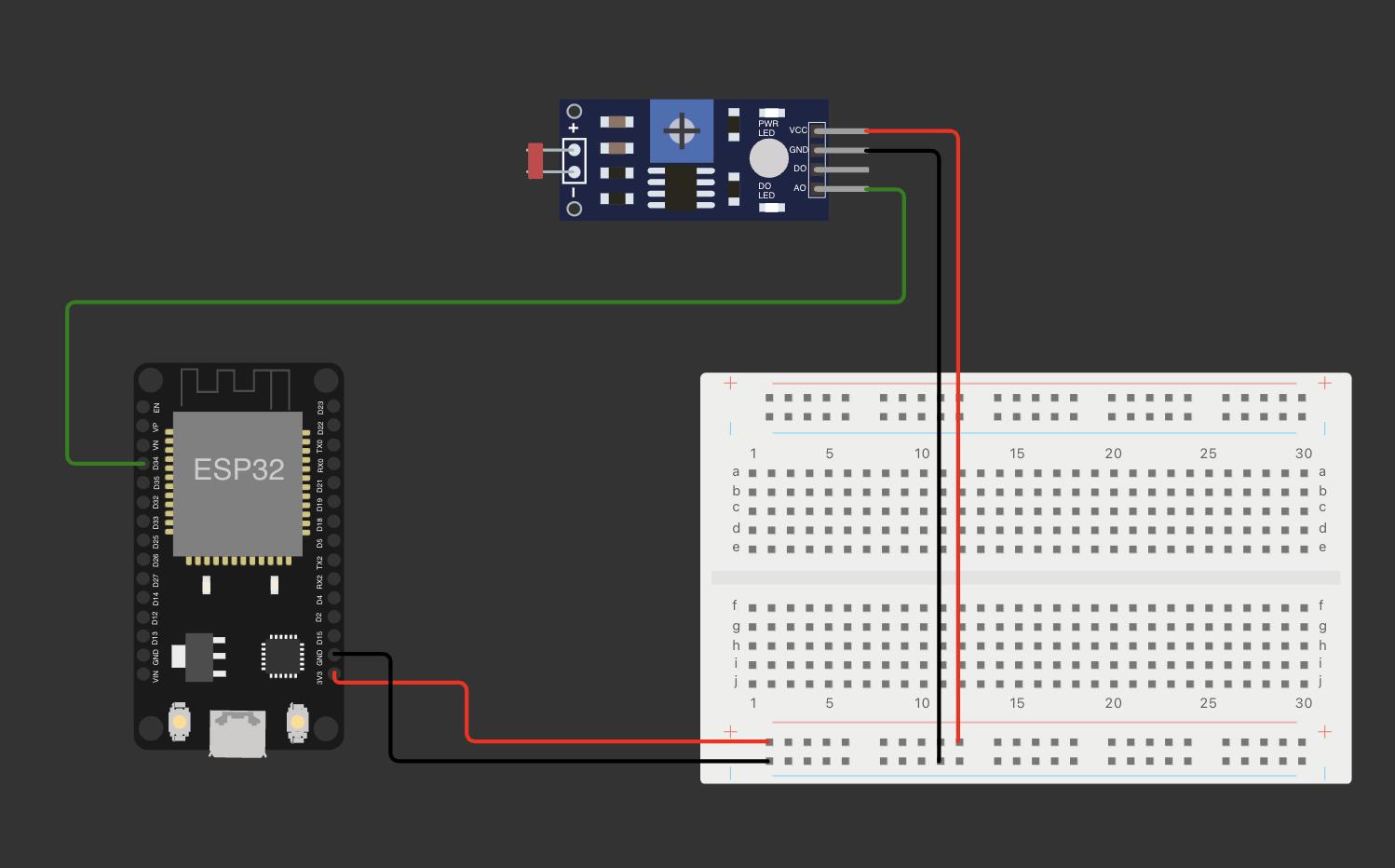
1. <https://wokwi.com/>
2. <https://emqx.com/>
3. <https://mqttx.app/>

**Soal!**

1. Buatlah proyek dengan skema berikut :



1. Buatlah proyek dengan skema berikut :



1. Buatlah proyek dengan skema berikut :

A circuit board with wires

Description automatically generated

Deskripsi Proyek:

Proyek ini bertujuan untuk membuat tiga buah node yang saling terhubung ke satu MQTT server. Node pertama akan digunakan untuk memonitor suhu dan kelembapan menggunakan sensor DHT22 pada ESP32, node kedua akan memonitor kecerahan/cahaya, dan node ketiga akan bertindak sebagai aktuator untuk menyiram tanaman menggunakan servo pada batas suhu diatas 25 dan kelembapan dibawah 60 dan menghidupkan lampu jika kecerahan/cahaya melewati batas 800.

Spesifikasi Proyek:

* 1. Node Pertama: Monitoring Suhu dan Kelembapan
* Menggunakan sensor DHT22 untuk memonitor suhu dan kelembapan lingkungan.
* Data suhu dan kelembapan akan dikirimkan ke MQTT server dengan topik **monitoring/sensor/kelembapan** dan **monitoring/sensor/suhu**.
  1. Node Kedua: Monitoring Cahaya/Kecerahan
* Memantau tingkat cahaya atau kecerahan lingkungan.
* Data cahaya akan dikirimkan ke MQTT server dengan topik **monitoring/sensor/cahaya.**
  1. Node Ketiga: Kontrol Aktuator
* Menerima data suhu dan kelembapan dari MQTT server.
* Menyiram tanaman menggunakan servo jika suhu atau kelembapan berada di bawah batas tertentu.
* Menghidupkan lampu jika tingkat kecerahan/cahaya melewati batas tertentu.

Langkah-langkah Proyek:

1. Persiapan Perangkat:

* Persiapkan tiga buah perangkat ESP32 dengan sensor DHT22 untuk node pertama, sensor cahaya untuk node kedua, dan servo serta lampu untuk node ketiga.
* Pastikan setiap perangkat terhubung ke jaringan WiFi dan memiliki akses ke MQTT server yang sama.

1. Pengembangan Kode Program:

* Tulis kode program untuk masing-masing node menggunakan Arduino IDE atau platform yang sesuai.
* Implementasikan logika untuk membaca data dari sensor DHT22 dan sensor cahaya.
* Gunakan MQTT library untuk mengirimkan dan menerima pesan dari MQTT server.
* Terapkan logika kontrol untuk aktuator (servo dan lampu) berdasarkan kondisi suhu, kelembapan, dan tingkat cahaya yang terukur.

1. Konfigurasi MQTT Server:

* Pastikan MQTT server telah dikonfigurasi dengan benar untuk menerima dan mengirimkan pesan dari dan ke tiga node.
* Buat topik-topik yang sesuai untuk menerima data suhu, kelembapan, dan cahaya, serta mengirimkan instruksi ke node ketiga.

1. Pengujian dan Debugging:

* Uji coba setiap node secara terpisah untuk memastikan sensor berfungsi dengan baik dan dapat terhubung ke MQTT server.
* Periksa apakah node ketiga dapat merespons dengan benar terhadap perubahan kondisi suhu, kelembapan, dan tingkat cahaya.

**Jawab!**

1. Rangkaian Skematik 1 (\*gambar/screenshoot)

|  |
| --- |
|  |

1. Rangkaian Skematik 2 (\*gambar/screenshoot)

|  |
| --- |
|  |

1. Rangkaian Skematik 3 (\*gambar/screenshoot)

|  |
| --- |
|  |

1. Program 1 (.ino,.c)

|  |
| --- |
| <https://wokwi.com/projects/394508490815732737>  #include <WiFi.h>  #include <WiFiClientSecure.h>  #include <PubSubClient.h>  #include <Arduino\_JSON.h>  #include "DHT.h"  #define DHTPIN 18     // Pin yang terhubung dengan output sensor DHT22  #define DHTTYPE DHT22 // Jenis sensor DHT (DHT22)  const char \*ssid = "Wokwi-GUEST";  const char \*password = "";  const char \*mqttServer = "dae791f7.ala.asia-southeast1.emqxsl.com";  const char \*topic = "iot-6/suhu";  const char \*mqtt\_username = "imaima";  const char \*mqtt\_password = "imaima";  int port = 8883;  static const char \*root\_ca PROGMEM = R"EOF(  -----BEGIN CERTIFICATE-----  MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh  MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3  d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD  QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT  MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j  b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG  9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB  CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97  nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt  43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P  T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4  gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO  BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR  TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw  DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr  hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg  06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF  PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls  YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk  CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=  -----END CERTIFICATE-----  )EOF";  DHT dht(DHTPIN, DHTTYPE);  WiFiClientSecure espClient;  PubSubClient client(espClient);  unsigned long lastTime = 0;  unsigned long intervalTime = 1000 \* 60 \* 5; // Publish data setiap 5 menit  void mqtt\_init();  void publishData();  void setup() {  **Serial**.begin(115200);    delay(1000);    WiFi.begin(ssid, password);  **Serial**.print("Connecting");    while (WiFi.status() != WL\_CONNECTED) {      delay(100);  **Serial**.print('.');    }  **Serial**.println("Connected");    dht.begin();    espClient.setCACert(root\_ca);    client.setServer(mqttServer, port);    while (!client.connected()) {      mqtt\_init();    }  }  void loop() {    client.loop();    unsigned long currentMillis = millis();    if (currentMillis - lastTime >= intervalTime) {      publishData();      lastTime = currentMillis;    }  }  void mqtt\_init() {    espClient.setCACert(root\_ca);    client.setServer(mqttServer, port);    String client\_id = "sub-esp32-iot-6-";    client\_id += WiFi.macAddress();    if (client.connect(client\_id.c\_str(), mqtt\_username, mqtt\_password)) {  **Serial**.println("Connected to MQTT broker");      client.subscribe(topic);    } else {  **Serial**.print("Failed to connect to MQTT broker, rc=");  **Serial**.println(client.state());      delay(2000);    }  }  void publishData() {    float temperature = dht.readTemperature();    float humidity = dht.readHumidity();    if (isnan(temperature) || isnan(humidity)) {  **Serial**.println("Failed to read data from DHT22!");      return;    }    JSONVar data;    data["temperature"] = temperature;    data["humidity"] = humidity;    String jsonString = JSON.stringify(data);    if (!client.publish(topic, jsonString.c\_str())) {  **Serial**.println("Failed to publish data to MQTT topic!");    } else {  **Serial**.println("Data published to MQTT topic:");  **Serial**.println(jsonString);    }  } |

1. Program 2 (.ino,.c)

|  |
| --- |
| <https://wokwi.com/projects/394509150748609537>  #include <WiFi.h>  #include <WiFiClientSecure.h>  #include <PubSubClient.h>  #include <Arduino\_JSON.h>  const char \*ssid = "Wokwi-GUEST";  const char \*password = "";  const char \*mqttServer = "dae791f7.ala.asia-southeast1.emqxsl.com";  const char \*topic = "iot-6/cahaya";  const char \*mqtt\_username = "imaima";  const char \*mqtt\_password = "imaima";  int port = 8883;  static const char \*root\_ca PROGMEM = R"EOF(  -----BEGIN CERTIFICATE-----  MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh  MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3  d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD  QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT  MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j  b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG  9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB  CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97  nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt  43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P  T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4  gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO  BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR  TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw  DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr  hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg  06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF  PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls  YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk  CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=  -----END CERTIFICATE-----  )EOF";  const int ldrPin = 34; // Pin yang terhubung dengan sensor LDR  WiFiClientSecure espClient;  PubSubClient client(espClient);  unsigned long lastTime = 0;  unsigned long intervalTime = 1000 \* 60 \* 5; // Publish data setiap 5 menit  void mqtt\_init();  void publishData();  void setup() {  **Serial**.begin(115200);    delay(1000);    WiFi.begin(ssid, password);  **Serial**.print("Connecting");    while (WiFi.status() != WL\_CONNECTED) {      delay(100);  **Serial**.print('.');    }  **Serial**.println("Connected");    pinMode(ldrPin, INPUT);    espClient.setCACert(root\_ca);    client.setServer(mqttServer, port);    while (!client.connected()) {      mqtt\_init();    }  }  void loop() {    client.loop();    unsigned long currentMillis = millis();    if (currentMillis - lastTime >= intervalTime) {      publishData();      lastTime = currentMillis;    }  }  void mqtt\_init() {    espClient.setCACert(root\_ca);    client.setServer(mqttServer, port);    String client\_id = "sub-esp32-iot-6-";    client\_id += WiFi.macAddress();    if (client.connect(client\_id.c\_str(), mqtt\_username, mqtt\_password)) {  **Serial**.println("Connected to MQTT broker");      client.subscribe(topic);    } else {  **Serial**.print("Failed to connect to MQTT broker, rc=");  **Serial**.println(client.state());      delay(2000);    }  }  void publishData() {    int lightLevel = analogRead(ldrPin);    JSONVar data;    data["light\_level"] = lightLevel;    String jsonString = JSON.stringify(data);    if (!client.publish(topic, jsonString.c\_str())) {  **Serial**.println("Failed to publish data to MQTT topic!");    } else {  **Serial**.println("Data published to MQTT topic:");  **Serial**.println(jsonString);    }  } |

1. Program 3 (.ino,.c)

|  |
| --- |
| <https://wokwi.com/projects/394509557568873473>  #include <WiFi.h>  #include <WiFiClientSecure.h>  #include <PubSubClient.h>  #include <Arduino\_JSON.h>  #include <ESP32Servo.h> // Tambahkan header library ESP32Servo  const char \*ssid = "Wokwi-GUEST";  const char \*password = "";  const char \*mqttServer = "dae791f7.ala.asia-southeast1.emqxsl.com";  const char \*topic = "iot-6/servo";  const char \*mqtt\_username = "imaima";  const char \*mqtt\_password = "imaima";  int port = 8883;  static const char \*root\_ca PROGMEM = R"EOF(  -----BEGIN CERTIFICATE-----  MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh  MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3  d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD  QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT  MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j  b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG  9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB  CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97  nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt  43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P  T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4  gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO  BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR  TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw  DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr  hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg  06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF  PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls  YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk  CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=  -----END CERTIFICATE-----  )EOF";  WiFiClientSecure espClient;  PubSubClient client(espClient);  unsigned long lastTime = 0;  unsigned long intervalTime = 1000 \* 60 \* 5; // Publish data setiap 5 menit  void mqtt\_init();  void publishData();  // Inisialisasi servo pada pin 18  Servo myServo;  void setup() {  **Serial**.begin(115200);    delay(1000);    WiFi.begin(ssid, password);  **Serial**.print("Connecting");    while (WiFi.status() != WL\_CONNECTED) {      delay(100);  **Serial**.print('.');    }  **Serial**.println("Connected");    espClient.setCACert(root\_ca);    client.setServer(mqttServer, port);    // Attach servo pada pin 18    myServo.attach(18);    while (!client.connected()) {      mqtt\_init();    }  }  void loop() {    client.loop();    unsigned long currentMillis = millis();    if (currentMillis - lastTime >= intervalTime) {      publishData();      lastTime = currentMillis;    }  }  void mqtt\_init() {    espClient.setCACert(root\_ca);    client.setServer(mqttServer, port);    String client\_id = "sub-esp32-iot-6-";    client\_id += WiFi.macAddress();    if (client.connect(client\_id.c\_str(), mqtt\_username, mqtt\_password)) {  **Serial**.println("Connected to MQTT broker");      client.subscribe(topic);    } else {  **Serial**.print("Failed to connect to MQTT broker, rc=");  **Serial**.println(client.state());      delay(2000);    }  }  void publishData() {    // Menggerakkan servo secara acak antara 0 dan 180 derajat    int pos = random(0, 180);    myServo.write(pos);    delay(500); // Memberikan waktu untuk servo bergerak    // Mendapatkan nilai suhu dan kelembaban    float temperature = random(20, 40); // Ganti dengan nilai suhu yang sebenarnya    float humidity = random(40, 70);    // Ganti dengan nilai kelembaban yang sebenarnya    JSONVar data;    data["temperature"] = temperature;    data["humidity"] = humidity;    String jsonString = JSON.stringify(data);    if (!client.publish(topic, jsonString.c\_str())) {  **Serial**.println("Failed to publish data to MQTT topic!");    } else {  **Serial**.println("Data published to MQTT topic:");  **Serial**.println(jsonString);    }  } |

1. Hasil Pengerjakan (screenshoot dijalankan)

|  |
| --- |
| Output 1 |
|  |

|  |
| --- |
| Output 2 |
|  |

|  |
| --- |
| Output 3 |
|  |