

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

## Bab 13

*Praktik Akses API Melalui Simulasi WOKWI*

*Adi Prasetyo*

*233140707111011*

*Fakultas Vokasi, Universitas Brawijaya*

*student@tomi.engineer*

## Contents

<b>1</b>	<b>Introduction (Pendahuluan)</b>	<b>3</b>
1.1	Latar belakang praktikum IoT yang dilakukan . . . . .	3
1.2	Tujuan eksperimen . . . . .	3
<b>2</b>	<b>Methodology (Metodologi)</b>	<b>3</b>
2.1	Tools & Materials (Alat dan Bahan) . . . . .	3
2.2	Implementation Steps (Langkah Implementasi) . . . . .	3
<b>3</b>	<b>Results and Discussion (Hasil dan Pembahasan)</b>	<b>4</b>
3.1	Experimental Results (Hasil Eksperimen) . . . . .	4
3.2	Discussion (Pembahasan) . . . . .	6
<b>4</b>	<b>Appendix (Lampiran)</b>	<b>6</b>
4.1	Kode Program . . . . .	7
4.2	Konfigurasi Wokwi . . . . .	9

## Abstract

Proyek ini mengimplementasikan sistem pengiriman data sensor suhu dan kelembapan menggunakan DHT22 yang terhubung ke mikrokontroler ESP32. Data dikirimkan ke server melalui protokol HTTP menggunakan simulasi Wokwi. Sistem dirancang untuk mendemonstrasikan prinsip komunikasi API dalam aplikasi IoT, di mana ESP32 membaca data dari sensor DHT22 dan mengirimkannya ke endpoint API dalam format JSON. Proyek ini mengilustrasikan penerapan teknologi IoT dalam pengumpulan dan pengiriman data secara real-time.

*Keywords—Internet of Things, API, DHT22, ESP32, HTTP, Wokwi Simulation*

## 1. Introduction (Pendahuluan)

### 1.1 Latar belakang praktikum IoT yang dilakukan

Dalam era digital, pengumpulan dan pengiriman data secara real-time menjadi komponen penting dalam pengembangan aplikasi IoT. Sensor DHT22 digunakan untuk membaca suhu dan kelembapan, sedangkan ESP32 bertindak sebagai pengontrol utama yang mengirimkan data ke server melalui API. Simulasi Wokwi digunakan untuk memvalidasi sistem ini tanpa perangkat keras fisik.

### 1.2 Tujuan eksperimen

1. Memahami prinsip kerja sensor DHT22 dalam membaca suhu dan kelembapan.
2. Mempelajari teknik pengiriman data ke server melalui API menggunakan ESP32.
3. Mengimplementasikan simulasi sistem IoT menggunakan platform Wokwi.
4. Menganalisis respons server terhadap data yang dikirimkan.

## 2. Methodology (Metodologi)

### 2.1 Tools & Materials (Alat dan Bahan)

1. Mikrokontroler: ESP32 DevKit C V4
2. Sensor: DHT22 Temperature and Humidity Sensor
3. Software: VS Code, Wokwi Simulator, PlatformIO

### 2.2 Implementation Steps (Langkah Implementasi)

1. Sensor DHT22 terhubung ke ESP32 dengan konfigurasi berikut:
  - VCC sensor terhubung ke pin 3V3 ESP32
  - GND sensor terhubung ke pin GND ESP32
  - Data sensor terhubung ke pin GPIO 27 ESP32

2. Program dikembangkan untuk membaca data suhu dan kelembapan dari sensor DHT22.
3. Data dikirimkan ke server melalui protokol HTTP dalam format JSON menggunakan WiFi-ClientSecure.
4. Endpoint API yang digunakan adalah `https://40ac-175-45-191-10.ngrok-free.app/api/posts`.
5. Simulasi dilakukan menggunakan platform Wokwi untuk memverifikasi pengiriman data.

### 3. Results and Discussion (Hasil dan Pembahasan)

#### 3.1 Experimental Results (Hasil Eksperimen)

```
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
Menghubungkan ke WiFi.. Terhubung!
{"nama_sensor":"Sensor GD", "nilai1":40.00, "nilai2":32.00}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":8,"nama_sensor":"Sensor GD","nilai1":40,"nilai2":32}}
{"nama_sensor":"Sensor GD", "nilai1":40.00, "nilai2":50.00}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":9,"nama_sensor":"Sensor GD","nilai1":40,"nilai2":50}}
{"nama_sensor":"Sensor GD", "nilai1":40.00, "nilai2":50.00}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":10,"nama_sensor":"Sensor GD","nilai1":40,"nilai2":50}}
█
```

Figure 1: Hasil dari terminal menunjukkan data suhu dan kelembapan yang dikirimkan ke server.

```
1 {  
2   "data": [  
3     {  
4       "id": 3,  
5       "nama_sensor": "test",  
6       "nilai1": 10,  
7       "nilai2": 20  
8     },  
9     {  
10      "id": 2,  
11      "nama_sensor": "sensor c",  
12      "nilai1": 111,  
13      "nilai2": 211  
14    },  
15    {  
16      "id": 1,  
17      "nama_sensor": "sensor c",  
18      "nilai1": 111,  
19      "nilai2": 211  
20    }  
21  ],
```

Figure 2: Isi di database setelah menerima data dari ESP32 melalui endpoint API.

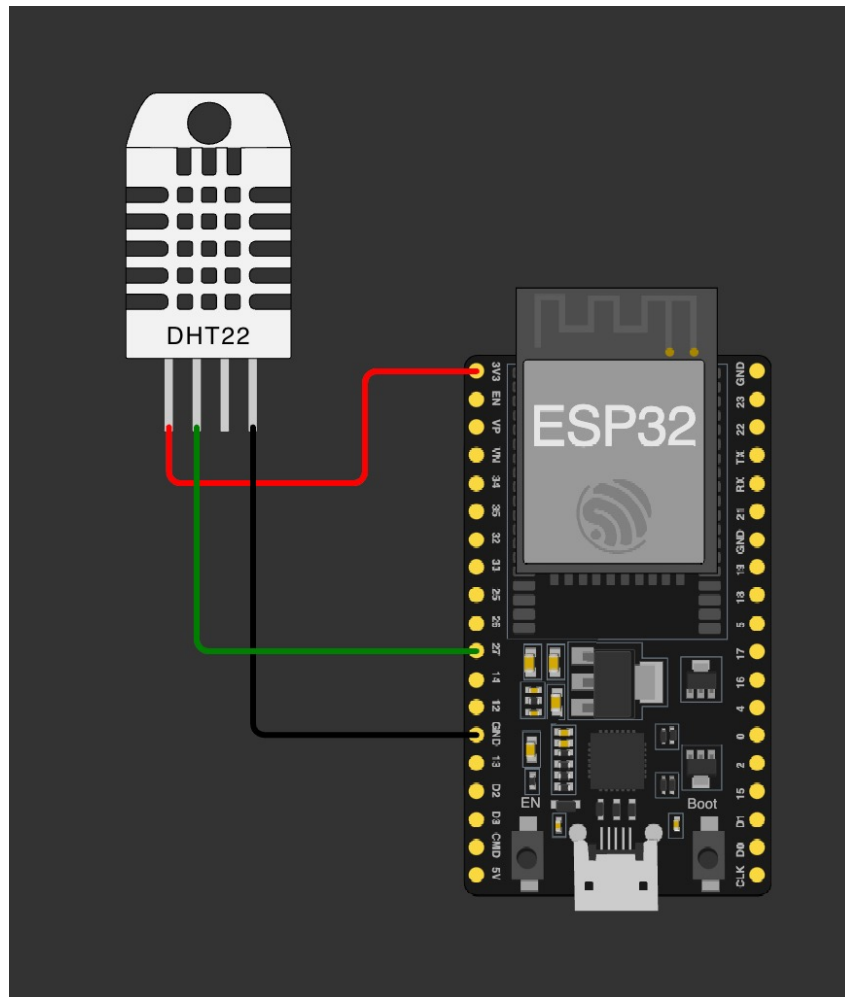


Figure 3: Diagram koneksi perangkat pada Wokwi Simulator.

Hasil simulasi menunjukkan beberapa temuan sebagai berikut:

- Sistem berhasil membaca data suhu dan kelembapan dari sensor DHT22.
- Data dikirimkan ke server dalam format JSON dengan interval 5 detik.
- Respons server menunjukkan bahwa data diterima dengan sukses.
- Simulasi Wokwi memvalidasi bahwa sistem dapat bekerja tanpa perangkat keras fisik.

### 3.2 Discussion (Pembahasan)

Sistem ini menunjukkan bagaimana ESP32 dapat digunakan untuk mengirimkan data sensor ke server melalui API. Dalam implementasi nyata, sistem ini dapat digunakan untuk aplikasi seperti pemantauan lingkungan, sistem rumah pintar, atau pengumpulan data cuaca. Tantangan seperti koneksi WiFi yang tidak stabil dan validasi data pada server perlu diperhatikan dalam implementasi praktis.

## 4. Appendix (Lampiran)

Kode program dan diagram konfigurasi Wokwi untuk sistem pengiriman data sensor DHT22:

## 4.1 Kode Program

```
1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <HttpClient.h>
4  #include <WiFiClientSecure.h>
5  #include "DHT.h"
6
7  #define DHTPIN 27
8  #define DHTTYPE DHT22
9
10 DHT dht(DHTPIN, DHTTYPE);
11
12 // Ganti dengan kredensial WiFi Anda
13 const char* ssid = "Wokwi-GUEST";
14 const char* password = "";
15
16 unsigned long previousMillis = 0;
17 const long interval = 5000; // Interval 5 detik (5000 ms)
18
19 void setup() {
20     Serial.begin(115200);
21
22     // Hubungkan ke WiFi
23     WiFi.begin(ssid, password);
24     Serial.print("Menghubungkan ke WiFi");
25     while (WiFi.status() != WL_CONNECTED) {
26         delay(500);
27         Serial.print(".");
28     }
29     Serial.println(" Terhubung!");
30
31     dht.begin();
32
33     // Tunggu sebentar agar koneksi stabil
34     delay(1000);
35 }
36
37 void loop() {
38     unsigned long currentMillis = millis();
39
40     // Lakukan POST setiap interval yang telah ditentukan
41     if (currentMillis - previousMillis >= interval) {
42         previousMillis = currentMillis;
43
44         float h = round(dht.readHumidity());
45         // Read temperature as Celsius (the default)
```

```

46     float t = round(dht.readTemperature());
47
48     // Check if any reads failed and exit early (to try again).
49     if (isnan(h) || isnan(t)) {
50         Serial.println(F("Failed to read from DHT sensor!"));
51         return;
52     }
53
54     // Compute heat index in Celsius (isFahreheit = false)
55     float hic = dht.computeHeatIndex(t, h, false);
56
57     // Inisialisasi WiFiClientSecure dan HTTPClient
58     WiFiClientSecure client;
59     client.setInsecure(); // Disable SSL certificate verification
60
61     HTTPClient http;
62     http.begin(client, "https://40ac-175-45-191-10.ngrok-free.app/api/
posts"); // Use HTTPS
63     http.addHeader("Content-Type", "application/json");
64
65     String payload = "{\"nama_sensor\":\"Sensor GD\", \"nilai1\":\" +
String(h) + ", \"nilai2\":\" + String(t) + "}";
66     Serial.println(payload); // Untuk melihat apakah payload sudah
terbentuk dengan benar
67
68     // Kirim POST request
69     int httpResponseCode = http.POST(payload);
70
71     // Tampilkan kode respons HTTP
72     Serial.print("Kode respons HTTP: ");
73     Serial.println(httpResponseCode);
74
75     // Tampilkan respons dari server jika request berhasil
76     if (httpResponseCode == 200 || httpResponseCode == 201) {
77         String response = http.getString();
78         Serial.println("Respons dari server:");
79         Serial.println(response);
80     } else {
81         Serial.println("Gagal mengirim data");
82     }
83
84     // Tutup koneksi HTTP
85     http.end();
86 }
87 }

```

Listing 1: main.cpp



## 4.2 Konfigurasi Wokwi

```
1  {
2    "version": 1,
3    "author": "Tomi",
4    "editor": "wokwi",
5    "parts": [
6      {
7        "type": "board-esp32-devkit-c-v4",
8        "id": "esp",
9        "top": 86.4,
10       "left": 24.04,
11       "attrs": {}
12     },
13     {
14       "type": "wokwi-dht22",
15       "id": "dht1",
16       "top": 19.5,
17       "left": -91.8,
18       "attrs": {}
19     }
20   ],
21   "connections": [
22     [
23       "esp:TX",
24       "$serialMonitor:RX",
25       "",
26       []
27     ],
28     [
29       "esp:RX",
30       "$serialMonitor:TX",
31       "",
32       []
33     ],
34     [
35       "dht1:VCC",
36       "esp:3V3",
37       "red",
38       [
39         "v19.2",
40         "h67.2",
41         "v-67.2"
42       ]
43     ],
44     [
45       "esp:GND.1",
```

```
46     "dht1:GND",
47     "black",
48     [
49         "h0"
50     ]
51 ],
52 [
53     "dht1:SDA",
54     "esp:27",
55     "green",
56     [
57         "v0"
58     ]
59 ]
60 ],
61 "dependencies": {}
62 }
```

Listing 2: diagram.json