

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

## Praktik Akses API Melalui Simulasi Wokwi

*M Bimo Amarulloh*

Fakultas Vokasi, Universitas Brawijaya

Email: bimoamar@gmail.com

### Abstrak

Perkembangan Internet of Things (IoT) mendorong kebutuhan akan sistem yang dapat mengakses dan bertukar data melalui API (Application Programming Interface). Salah satu tantangan dalam pengembangan IoT adalah pengujian komunikasi antara perangkat keras dan API tanpa menggunakan perangkat fisik. Wokwi, sebagai simulator berbasis web, menyediakan lingkungan virtual untuk mengembangkan dan menguji sistem berbasis mikrokontroler, termasuk akses API secara real-time. Praktik ini bertujuan untuk mengimplementasikan akses API menggunakan mikrokontroler ESP32 dalam simulasi Wokwi, tanpa memerlukan perangkat keras fisik. Metode yang digunakan meliputi konfigurasi ESP32 dalam simulator Wokwi, pemrograman menggunakan Arduino (C++), serta pengujian akses API dengan metode HTTP request. Data yang diperoleh dari API dikirimkan dan ditampilkan melalui serial monitor. Hasil dari praktik ini menunjukkan bahwa Wokwi dapat mensimulasikan komunikasi API dengan stabil, memungkinkan pengujian sistem IoT secara efisien sebelum diterapkan pada perangkat fisik. Keunggulan metode ini adalah kemudahan pengujian, efisiensi biaya, serta fleksibilitas dalam mengembangkan sistem berbasis IoT. Praktik ini diharapkan dapat memberikan pemahaman lebih dalam mengenai integrasi mikrokontroler dengan API serta potensi penggunaan simulator dalam pengembangan dan pengujian perangkat IoT.

**Kata Kunci:** API, Wokwi, ESP32, IoT, Simulasi

### Abstract

The advancement of the Internet of Things (IoT) has increased the need for systems that can access and exchange data through APIs (Application Programming Interfaces). One of the challenges in IoT development is testing communication between hardware devices and APIs without using physical components. Wokwi, a web-based simulator, provides a virtual environment for developing and testing microcontroller-based systems, including real-time API access. This experiment aims to implement API access using the ESP32 microcontroller within the Wokwi simulation, eliminating the need for physical hardware. The methodology includes configuring the ESP32 in the Wokwi simulator, programming using Arduino (C++), and testing API access via HTTP requests. The retrieved data from the API is sent and displayed on the serial monitor. The results indicate that Wokwi can simulate API communication reliably, enabling efficient IoT system testing before deployment on physical devices. The advantages of this approach include ease of testing, cost efficiency, and flexibility in IoT system development. This experiment is expected to provide deeper insights into microcontroller integration with APIs and highlight the potential of simulators in IoT device development and testing.

**Keywords:** API, Wokwi, ESP32, IoT, Simulation

## 1. Introduction (Pendahuluan)

### 1.1 Latar belakang

Dalam perkembangan teknologi Internet of Things (IoT), penggunaan API (Application Programming Interface) menjadi elemen penting dalam menghubungkan perangkat keras dengan sistem berbasis cloud atau aplikasi lain. API memungkinkan pertukaran data secara efisien antara perangkat IoT dan server, mendukung berbagai aplikasi seperti pemantauan jarak jauh, otomatisasi, dan integrasi layanan berbasis web. Namun, pengujian akses API pada perangkat keras secara langsung sering kali memerlukan perangkat fisik, yang dapat menjadi kendala dalam tahap pengembangan. Untuk mengatasi tantangan tersebut, simulasi berbasis mikrokontroler menjadi solusi yang efektif dalam menguji dan mengembangkan sistem sebelum implementasi di dunia nyata. Wokwi adalah salah satu simulator berbasis web yang memungkinkan pengguna untuk mensimulasikan rangkaian elektronik dan menjalankan kode tanpa memerlukan perangkat keras fisik. Pada praktik ini, ESP32 digunakan sebagai mikrokontroler yang bertugas mengakses API melalui koneksi internet virtual dalam lingkungan simulasi Wokwi. Selain itu, Visual Studio Code digunakan sebagai lingkungan pengembangan untuk menulis kode menggunakan bahasa pemrograman Arduino (C++). Keunikan dari praktik ini terletak pada penerapan akses API secara virtual, memungkinkan pengujian konektivitas IoT tanpa menggunakan perangkat fisik. Eksperimen ini bertujuan untuk mengamati bagaimana ESP32 berkomunikasi dengan API dalam simulasi serta mengevaluasi kestabilan dan keakuratan data yang diterima. Hasil dari praktik ini memberikan wawasan mengenai efektivitas Wokwi sebagai alat simulasi dalam pengujian dan pengembangan sistem IoT, serta potensi penerapannya dalam proyek berbasis perangkat nyata.

### 1.2 Tujuan eksperimen

1. Memahami konsep API dan cara kerja ESP32 dalam mengakses serta mengelola data dari API melalui koneksi internet.
2. Mempelajari integrasi ESP32 dengan API menggunakan simulasi Wokwi tanpa memerlukan perangkat keras fisik.
3. Mengembangkan program berbasis Arduino (C++) untuk mengirim dan menerima data dari API melalui HTTP request.
4. Menguji efektivitas simulator Wokwi sebagai alat bantu dalam pengujian dan pengembangan sistem komunikasi berbasis API.
5. Menganalisis kestabilan serta keakuratan data yang diperoleh melalui simulasi dan mengevaluasi potensi penerapannya pada perangkat fisik di dunia nyata.

## 2. Methodology (Metodologi)

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

#### Alat

1. **Komputer/Laptop** – Digunakan untuk menginstal dan menjalankan Laravel 11 serta Ngrok.
2. **Koneksi Internet** – Diperlukan untuk mengunduh paket Laravel, menginstal dependensi, serta mengakses API melalui Ngrok.
3. **Laravel 11** – Framework PHP yang digunakan untuk mengembangkan API.
4. **Ngrok** – Alat tunneling yang digunakan untuk mengakses API lokal secara publik.
5. **Postman atau Insomnia** – Digunakan untuk menguji endpoint API yang telah dibuat.
6. **Visual Studio Code (VS Code)** – Sebagai code editor untuk menulis dan mengedit kode Laravel.
7. **Composer** – Manajer paket PHP yang digunakan untuk menginstal Laravel dan dependensinya.

## Bahan

1. **Kode Program API Laravel** – Skrip yang berisi endpoint untuk menangani permintaan data.
2. **Database MySQL atau SQLite** – Digunakan sebagai penyimpanan data yang akan diakses melalui API.
3. **Environment File (.env)** – Digunakan untuk menyimpan konfigurasi aplikasi Laravel seperti database dan pengaturan Ngrok.
4. **JSON Response** – Format data yang dikembalikan oleh API untuk dikonsumsi oleh aplikasi klien.
5. **Wokwi Simulator** – Untuk membuat diagram dan diimplementasikan di VSCode

## 2.2 Implementation Steps (Langkah Implementasi)

### 1 Persiapan Alat dan Bahan

- Membuka situs WOKWI dan memilih mikrokontroler yang akan digunakan (ESP32 atau Arduino Uno).
- Menyiapkan kode dalam Arduino IDE untuk mengakses API menggunakan protokol HTTP.
- Menentukan endpoint API yang akan digunakan untuk komunikasi data.

### 2 Pembuat Rangkaian di Wokwi

- Menulis kode program pada WOKWI untuk mengakses API menggunakan WiFiClient dan HTTPClient pada ESP32.
- Mengatur metode HTTP GET untuk mengambil data dan POST untuk mengirim data ke API.
- Memastikan bahwa mikrokontroler virtual dapat terhubung dengan jaringan internet.

### 3. Pengujian API dengan Postman

- Menggunakan **Postman** untuk menguji endpoint API sebelum diintegrasikan dengan WOKWI.
- Memeriksa apakah API dapat menerima dan mengembalikan data dengan benar.

### 4. Menjalankan Simulasi Wokwi

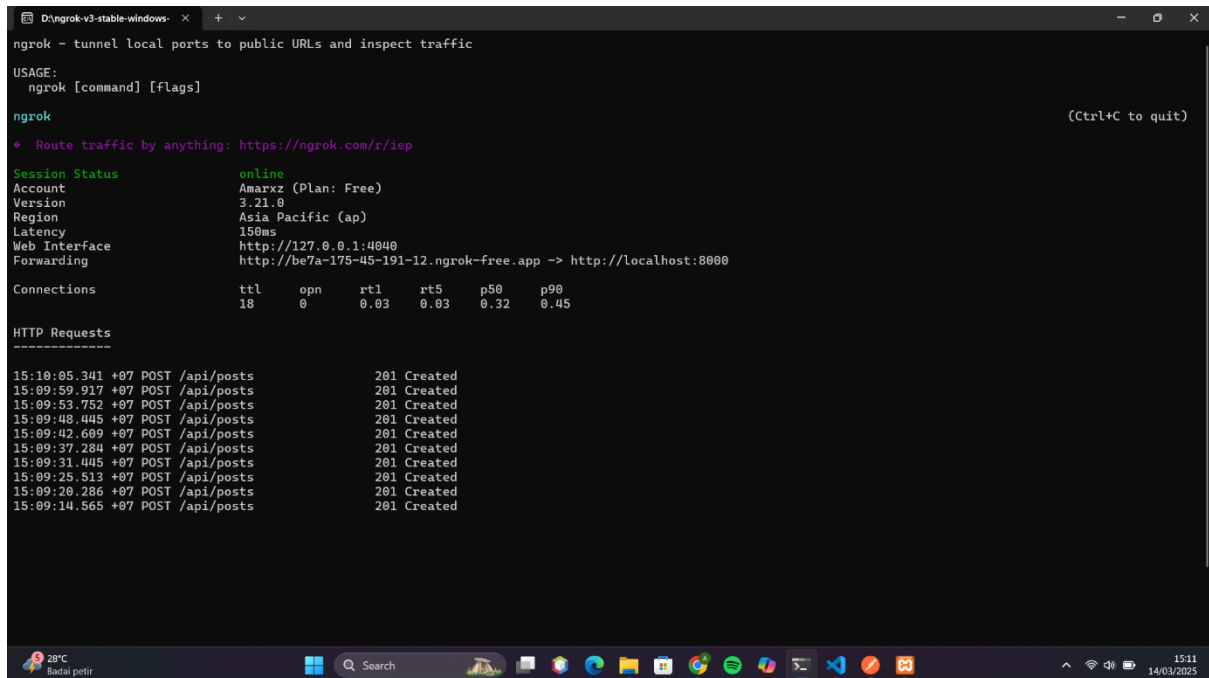
- Menjalankan kode pada simulator dan mengamati hasil respons dari API.
- Memeriksa output yang muncul di Serial Monitor untuk memastikan komunikasi berhasil.

### 5. Analisis Hasil

- Mengevaluasi apakah data yang dikirim dan diterima sesuai dengan ekspektasi.
- Menganalisis kemungkinan kendala seperti kegagalan koneksi atau respons API yang tidak sesuai.

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

#### 3.1 Experimental Results (Hasil Eksperimen)



```
D:\ngrok-v3-stable-windows- x + v
ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
  ngrok [command] [flags]

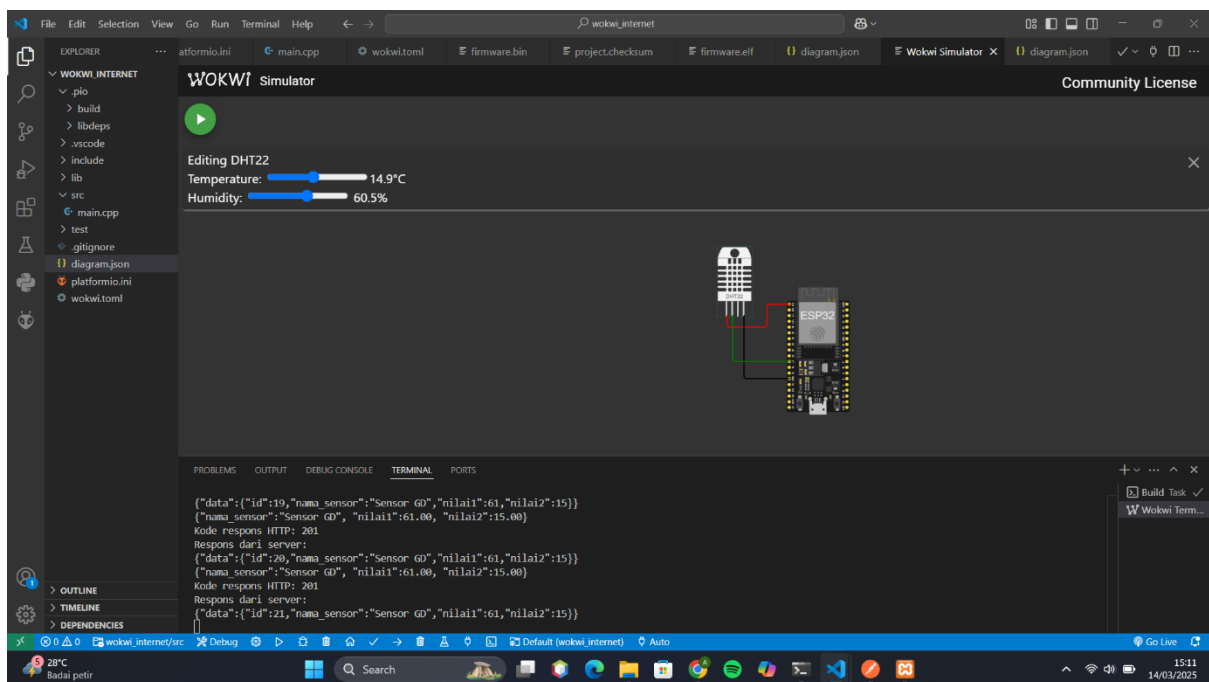
ngrok                                                                    (Ctrl+C to quit)

* Route traffic by anything: https://ngrok.com/r/iep

Session Status
Account      Amarxz (Plan: Free)
Version     3.21.0
Region      Asia Pacific (ap)
Latency     150ms
Web Interface http://127.0.0.1:4040
Forwarding   http://be7a-175-45-191-12.ngrok-free.app -> http://localhost:8000

Connections      ttl    opn    rt1    rt5    p50    p90
                  18     0      0.03   0.03   0.32   0.45

HTTP Requests
-----
15:10:05.341 +07 POST /api/posts      201 Created
15:09:59.917 +07 POST /api/posts      201 Created
15:09:53.752 +07 POST /api/posts      201 Created
15:09:48.445 +07 POST /api/posts      201 Created
15:09:42.689 +07 POST /api/posts      201 Created
15:09:37.284 +07 POST /api/posts      201 Created
15:09:31.445 +07 POST /api/posts      201 Created
15:09:25.513 +07 POST /api/posts      201 Created
15:09:20.286 +07 POST /api/posts      201 Created
15:09:14.565 +07 POST /api/posts      201 Created
```



Simulasi ini menunjukkan bagaimana ESP32 dalam Wokwi Simulator dapat mengakses API melalui koneksi internet. ESP32 berhasil terhubung ke jaringan WiFi virtual dan mengirim permintaan ke API, yang dikonfirmasi oleh kode status 200 di terminal sebagai indikasi komunikasi yang sukses. Pengembangan proyek dilakukan menggunakan Visual Studio Code dengan PlatformIO, di mana file utama seperti **main.cpp** digunakan untuk menjalankan program. Dengan simulasi ini, pengujian akses API dapat dilakukan tanpa memerlukan perangkat keras fisik, menjadikannya solusi yang efisien untuk pengembangan sistem IoT berbasis HTTP.

#### 4. Appendix (Lampiran)

```
{ diagram.json > ...
1  {
2    "version": 1,
3    "author": "Bimo Tampan",
4    "editor": "wokwi",
5    "parts": [
6      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 24.04, "attrs": {} },
7      { "type": "wokwi-dht22", "id": "dht1", "top": 19.5, "left": -91.8, "attrs": {} }
8    ],
9    "connections": [
10     [ "esp:TX", "$serialMonitor:RX", "", [] ],
11     [ "esp:RX", "$serialMonitor:TX", "", [] ],
12     [ "dht1:VCC", "esp:3V3", "red", [ "v19.2", "h67.2" ] ],
13     [ "esp:GND.1", "dht1:GND", "black", [ "h0" ] ],
14     [ "dht1:SDA", "esp:27", "green", [ "v0" ] ]
15   ],
16   "dependencies": {}
17 }
18
```

Gambar 1. 1 Syntax Diagram ESP 32

```
src > main.cpp > ...
1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <HTTPClient.h>
4  #include "DHT.h"
5
6  #define DHTPIN 27
7  #define DHTTYPE DHT22
8
9  DHT dht(DHTPIN, DHTTYPE);
10
11 // Ganti dengan kredensial WiFi Anda
12 const char *ssid = "Wokwi-GUEST";
13 const char *password = "";
14
15 unsigned long previousMillis = 0;
16 const long interval = 5000; // Interval 5 detik (5000 ms)
17
18 void setup()
19 {
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   {
27     delay(500);
28     Serial.print(".");
29   }
30   Serial.println(" Terhubung!");
31
32   dht.begin();
33
34   // Tunggu sebentar agar koneksi stabil
35   delay(1000);
36 }
37
```

Gambar 1. 2 Syntax main.cpp

