

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Praktik Akses API Melalui Simulasi WOKWI

*Lucky Ardiansyah 233140701111015
Fakultas Vokasi, Universitas Brawijaya
Email: luckyardiansyah685@gmail.com*

Abstract (Abstrak)

Praktikum ini bertujuan untuk mempelajari cara mengakses API menggunakan simulasi perangkat keras virtual melalui WOKWI. WOKWI adalah platform simulasi berbasis web yang memungkinkan pengembangan dan pengujian proyek IoT (Internet of Things) tanpa memerlukan perangkat fisik. Dalam praktikum ini, API yang dikembangkan menggunakan Laravel 11 dihubungkan dengan mikrokontroler virtual ESP32 pada WOKWI. Koneksi dilakukan dengan menggunakan protokol HTTP untuk mengirimkan dan menerima data dari API secara real-time. Untuk memungkinkan akses publik terhadap API, digunakan Ngrok sebagai terowongan jaringan. Hasil praktikum menunjukkan bahwa perangkat simulasi di WOKWI mampu melakukan permintaan HTTP (GET, POST) ke API Laravel, serta menerima respons dalam format JSON. Praktikum ini memberikan wawasan tentang integrasi sistem IoT dengan layanan berbasis cloud, serta pentingnya simulasi perangkat untuk menguji konektivitas dan fungsionalitas sebelum implementasi perangkat keras sesungguhnya.

Kata Kunci: WOKWI, Laravel 11, Ngrok, API, ESP32, IoT

1. Introduction (Pendahuluan)

1.1 Latar Belakang

Dalam pengembangan sistem IoT (Internet of Things), koneksi antara perangkat keras dan layanan berbasis cloud menjadi elemen penting. Salah satu cara untuk menghubungkan perangkat IoT dengan layanan backend adalah melalui Application Programming Interface (API). API memungkinkan perangkat untuk mengirimkan dan menerima data secara real-time menggunakan protokol HTTP. Namun, pengujian konektivitas ini sering kali terkendala oleh keterbatasan perangkat keras fisik. Oleh karena itu, WOKWI hadir sebagai solusi simulasi perangkat keras berbasis web yang memungkinkan pengembang untuk menguji mikrokontroler, seperti ESP32, secara virtual. Dengan WOKWI, pengembang dapat melakukan simulasi pengiriman dan penerimaan data ke API tanpa memerlukan perangkat fisik, mempercepat proses pengujian dan pengembangan. Dalam praktikum ini, dilakukan simulasi akses API menggunakan WOKWI, di mana perangkat ESP32 virtual dikonfigurasi untuk mengirimkan permintaan HTTP (GET dan POST) ke API yang dikembangkan menggunakan Laravel 11. Untuk menghubungkan API ke jaringan publik, digunakan Ngrok sebagai terowongan jaringan, sehingga perangkat virtual di WOKWI dapat mengakses API secara online. Melalui praktik ini, peserta diharapkan memahami bagaimana proses komunikasi antara perangkat IoT dan API berlangsung, serta pentingnya simulasi untuk memastikan fungsionalitas sistem sebelum implementasi nyata.

1.2 Tujuan Eksperimen

- Memahami konsep dasar API dan perannya dalam integrasi perangkat IoT dengan layanan backend.
- Mempelajari penggunaan WOKWI untuk mensimulasikan perangkat keras, khususnya mikrokontroler ESP32.
- Mengimplementasikan koneksi antara perangkat ESP32 virtual dan API berbasis Laravel 11 menggunakan metode HTTP request (GET dan POST).
- Menggunakan Ngrok untuk mengonline-kan API agar dapat diakses oleh perangkat virtual di WOKWI.
- Menguji fungsionalitas API dengan memproses data yang dikirimkan dari perangkat ESP32 melalui simulasi WOKWI.
- Meningkatkan keterampilan troubleshooting terkait konektivitas perangkat IoT dan pengujian API.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat & Bahan)

- **Alat:**
 - Laptop/Komputer
 - Browser (Google Chrome/Mozilla Firefox)
 - Terminal/Command Prompt
 - Postman
- **Bahan:**
 - WOKWI (ESP32)
 - Laravel 11
 - Ngrok
 - Database (phpMyAdmin + MySQL)
 - File konfigurasi Laravel (.env)
 - Kode program WOKWI (ESP32)
 - Koneksi internet stabil

2.2 Implementation Steps (Langkah Implementasi)

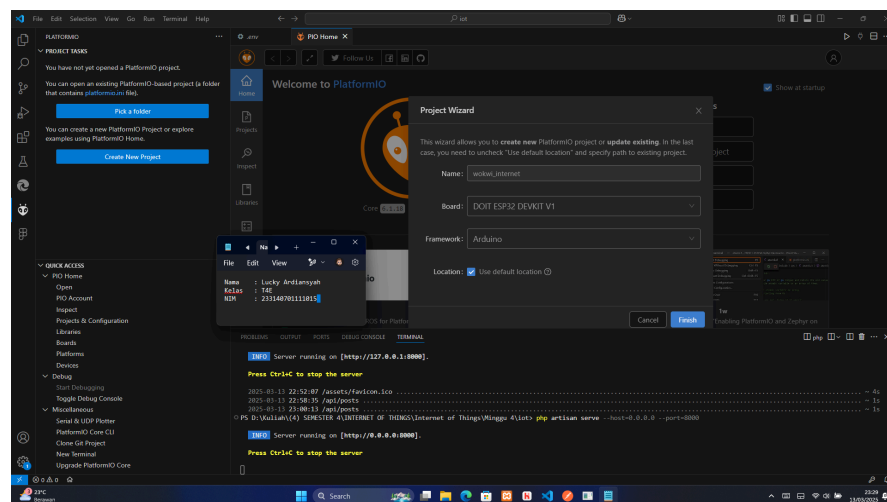
1. Menjalankan API dengan perintah Laravel

```
PS D:\Kuliah\4) SEMESTER 4\INTERNET OF THINGS\Internet of Things\Minggu 4\i> php artisan serve --host=0.0.0.0 --port=8000
```

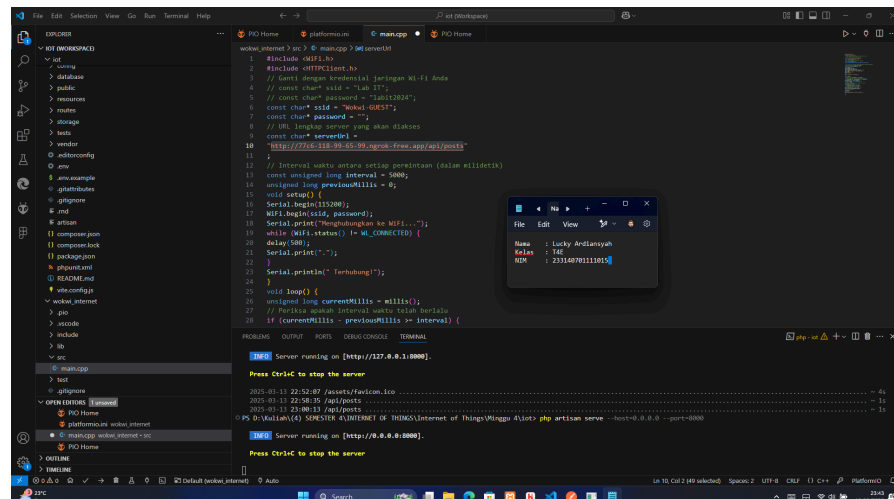
INFO Server running on [http://0.0.0.0:8000].

Press Ctrl+C to stop the server

2. Membuat file baru Platform.io



3. Script main.cpp



The screenshot displays the Visual Studio Code editor with a C++ project named 'platformio'. The main.cpp file contains the following code:

```

1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <WebServer.h>
4  #include <EEPROM.h>
5  #include <Wire.h>
6  #include <OneWire.h>
7  #include <DallasTemperature.h>
8  #include <Adafruit_NeoPixel.h>
9  #include <Adafruit_GFX.h>
10 #include <Adafruit_SSD1306.h>
11 #include <Adafruit_ILI9341.h>
12 #include <Adafruit_ST7735.h>
13 #include <Adafruit_GFX.h>
14 #include <Adafruit_ILI9341.h>
15 #include <Adafruit_ST7735.h>
16 #include <Adafruit_GFX.h>
17 #include <Adafruit_ILI9341.h>
18 #include <Adafruit_ST7735.h>
19 #include <Adafruit_GFX.h>
20 #include <Adafruit_ILI9341.h>
21 #include <Adafruit_ST7735.h>
22 #include <Adafruit_GFX.h>
23 #include <Adafruit_ILI9341.h>
24 #include <Adafruit_ST7735.h>
25 #include <Adafruit_GFX.h>
26 #include <Adafruit_ILI9341.h>
27 #include <Adafruit_ST7735.h>
28 #include <Adafruit_GFX.h>
29 #include <Adafruit_ILI9341.h>
30 #include <Adafruit_ST7735.h>
31 #include <Adafruit_GFX.h>
32 #include <Adafruit_ILI9341.h>
33 #include <Adafruit_ST7735.h>
34 #include <Adafruit_GFX.h>
35 #include <Adafruit_ILI9341.h>
36 #include <Adafruit_ST7735.h>
37 #include <Adafruit_GFX.h>
38 #include <Adafruit_ILI9341.h>
39 #include <Adafruit_ST7735.h>
40 #include <Adafruit_GFX.h>
41 #include <Adafruit_ILI9341.h>
42 #include <Adafruit_ST7735.h>
43 #include <Adafruit_GFX.h>
44 #include <Adafruit_ILI9341.h>
45 #include <Adafruit_ST7735.h>
46 #include <Adafruit_GFX.h>
47 #include <Adafruit_ILI9341.h>
48 #include <Adafruit_ST7735.h>
49 #include <Adafruit_GFX.h>
50 #include <Adafruit_ILI9341.h>
51 #include <Adafruit_ST7735.h>
52 #include <Adafruit_GFX.h>
53 #include <Adafruit_ILI9341.h>
54 #include <Adafruit_ST7735.h>
55 #include <Adafruit_GFX.h>
56 #include <Adafruit_ILI9341.h>
57 #include <Adafruit_ST7735.h>
58 #include <Adafruit_GFX.h>
59 #include <Adafruit_ILI9341.h>
60 #include <Adafruit_ST7735.h>
61 #include <Adafruit_GFX.h>
62 #include <Adafruit_ILI9341.h>
63 #include <Adafruit_ST7735.h>
64 #include <Adafruit_GFX.h>
65 #include <Adafruit_ILI9341.h>
66 #include <Adafruit_ST7735.h>
67 #include <Adafruit_GFX.h>
68 #include <Adafruit_ILI9341.h>
69 #include <Adafruit_ST7735.h>
70 #include <Adafruit_GFX.h>
71 #include <Adafruit_ILI9341.h>
72 #include <Adafruit_ST7735.h>
73 #include <Adafruit_GFX.h>
74 #include <Adafruit_ILI9341.h>
75 #include <Adafruit_ST7735.h>
76 #include <Adafruit_GFX.h>
77 #include <Adafruit_ILI9341.h>
78 #include <Adafruit_ST7735.h>
79 #include <Adafruit_GFX.h>
80 #include <Adafruit_ILI9341.h>
81 #include <Adafruit_ST7735.h>
82 #include <Adafruit_GFX.h>
83 #include <Adafruit_ILI9341.h>
84 #include <Adafruit_ST7735.h>
85 #include <Adafruit_GFX.h>
86 #include <Adafruit_ILI9341.h>
87 #include <Adafruit_ST7735.h>
88 #include <Adafruit_GFX.h>
89 #include <Adafruit_ILI9341.h>
90 #include <Adafruit_ST7735.h>
91 #include <Adafruit_GFX.h>
92 #include <Adafruit_ILI9341.h>
93 #include <Adafruit_ST7735.h>
94 #include <Adafruit_GFX.h>
95 #include <Adafruit_ILI9341.h>
96 #include <Adafruit_ST7735.h>
97 #include <Adafruit_GFX.h>
98 #include <Adafruit_ILI9341.h>
99 #include <Adafruit_ST7735.h>
100 #include <Adafruit_GFX.h>

```

The program is a simple HTTP server that listens on port 8080 and responds to GET requests with a JSON payload. The Output window shows the program's execution, including the IP address and port of the server.

3. Result & Discussion (Hasil dan Pembahasan)

3.1 Experimental Result (Hasil Eksperimen)

