LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

*Mirza Marwa Rosyidah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: mirzamarwa76@gmail.com*

**Abstract**

Internet of Things (IoT) memungkinkan perangkat untuk berkomunikasi dan bertukar data melalui jaringan internet. Praktikum ini bertujuan untuk mensimulasikan integrasi ESP32 dengan API Laravel menggunakan Wokwi Simulator. Sensor DHT22 digunakan untuk mengukur suhu dan kelembaban, yang kemudian dikirim ke server Laravel melalui HTTP POST dan disimpan dalam database MySQL. Hasil simulasi menunjukkan bahwa ESP32 berhasil mengirim data ke server dan menerima respons sukses dengan kode status HTTP 200. Praktikum ini membuktikan bahwa ESP32 dapat diintegrasikan dengan API Laravel untuk pengelolaan dan penyimpanan data sensor secara real-time, yang berguna untuk berbagai aplikasi IoT.

Kata Kunci : *Internet of Things, ESP32,* *Sensor DHT22, Wokwi, Vscode,* *API Laravel*

**1. Introduction**

* 1. **Latar belakang**

Perkembangan teknologi IoT telah memungkinkan berbagai perangkat untuk saling berkomunikasi dan bertukar data secara otomatis melalui jaringan internet. Salah satu tantangan utama dalam implementasi IoT adalah bagaimana data sensor dikumpulkan, dikirim, dan dikelola dengan efisien. ESP32 merupakan mikrokontroler dengan konektivitas WiFi yang memungkinkan pengiriman data ke server berbasis web. Dengan adanya API Laravel dan database MySQL, data yang dikirim dari perangkat dapat disimpan dan diakses untuk berbagai keperluan analisis dan pengambilan keputusan.

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan untuk,

1. Mempelajari cara mengintegrasikan ESP32 dengan API berbasis Laravel.
2. Mensimulasikan pengiriman data sensor suhu dan kelembaban menggunakan Wokwi Simulator.
3. Menguji apakah data yang dikirim dapat diterima dan disimpan dalam database MySQL dengan sukses.
4. Mengevaluasi respons API Laravel terhadap permintaan data dari ESP32

**2. Methodology (Metodologi)**

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

Alat dan bahan yang digunakan dalam praktikum ini:

1. ESP32 (Simulasi Wokwi)
2. Sensor DHT22 (Simulasi Wokwi)
3. Framework Laravel untuk API
4. MySQL sebagai database
5. Ngrok untuk akses API secara publik
6. PlatformIO pengembangan ESP32

**2.2 Implementation Steps (Langkah Implementasi)**

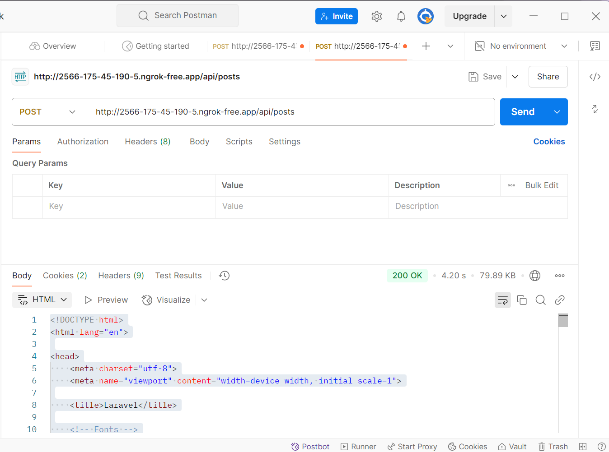
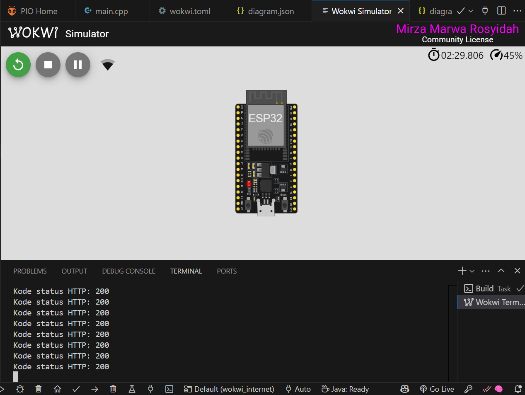
1. Create new project wokwi internet
2. Ubah codingan main.cpp sesuai modul
3. Ubah url ngrok dengan url milih sendiri
4. Create file wokwi.toml dan diagram.json dan ubah codingan sesuai dimodul
5. Build project tersebut
6. Jalankan digram.json hingga pada terminal muncul “kode status HTTP 200”
7. Jika suda berhasil buat projek pada web wokwi
8. Buat simulator DHT 22 dan ESP32 pada wokwi
9. Ubah codingan digram.json pada vscode sesuai dengan codingan diagram.json pada web simulator wokwi
10. Edit File platformio.ini sesuai dengan modul
11. Ubah main.cpp pada vscode dan ubah juga url ngrok dengan url ngrok milik pribadi
12. Start simulator wokwi
13. Dan lihat hasilnya

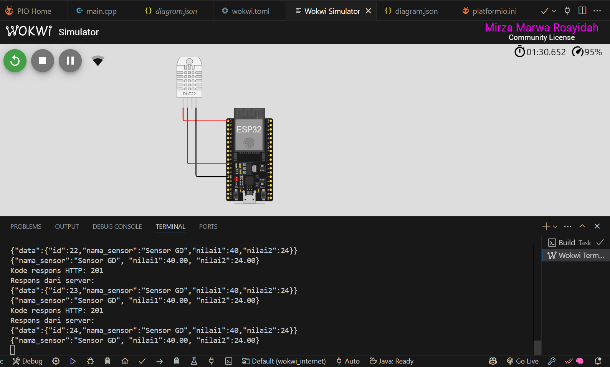
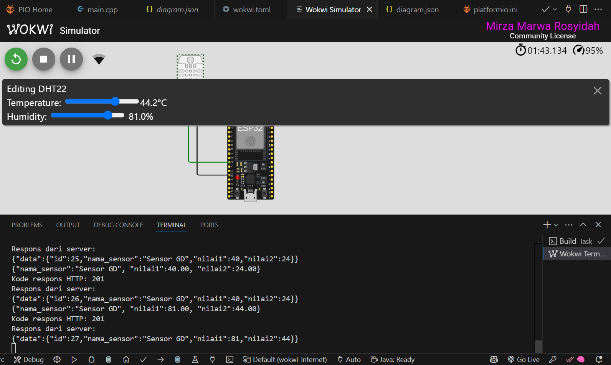
**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

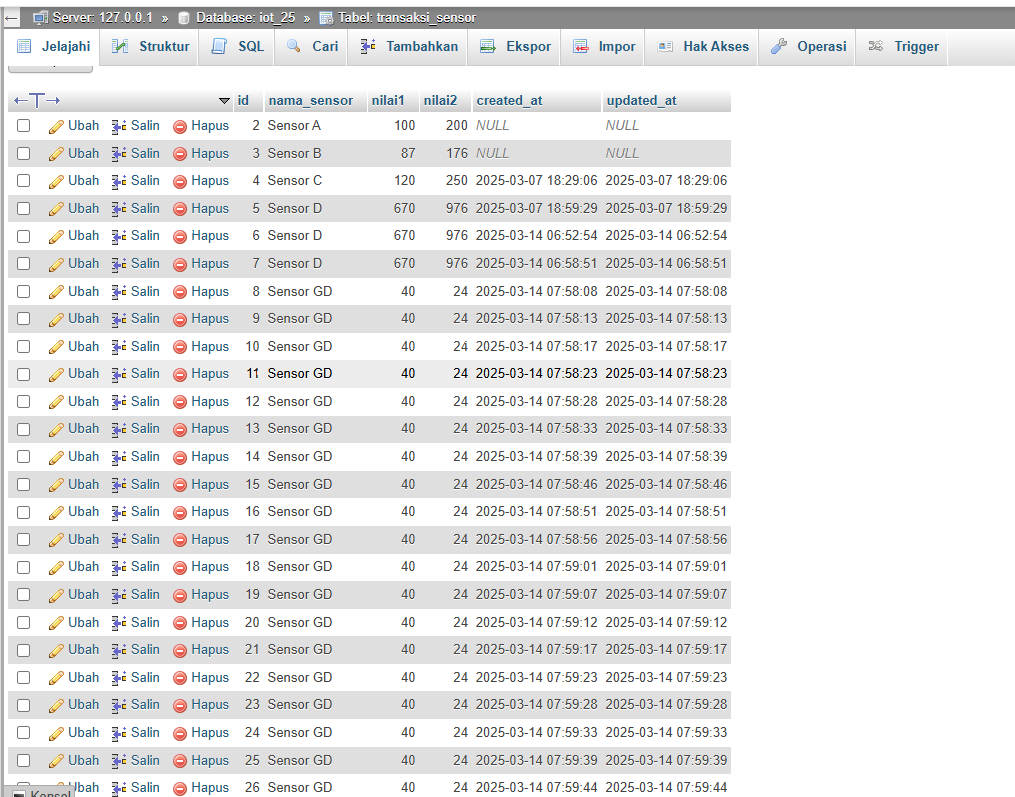
Hasil simulasi menunjukkan bahwa simulasi sensor HC-SR04 pada vscode berhasil membaca jarak 399.94 cm secara konsisten,

**Screenshoot hasil simulasi:**  
 Kode Status HTTP:200



Tampailan simulator dan serial monitor  
  

Data muncul dan tersimpan pada database



**Pembahasan:**

Pada praktik tersebut didapatkan ESP32 berhasil terhubung ke jaringan WiFi Wokwi-GUEST sehingga data suhu dan kelembaban berhasil dibaca dan dikirim ke API Laravel. API Laravel merespons dengan kode status HTTP 200, menandakan bahwa permintaan berhasil diproses. Data yang dikirim dan disimpan oleh ESP32 dalam database MySQL, menunjukkan keberhasilan sistem data sensor secara real-time.

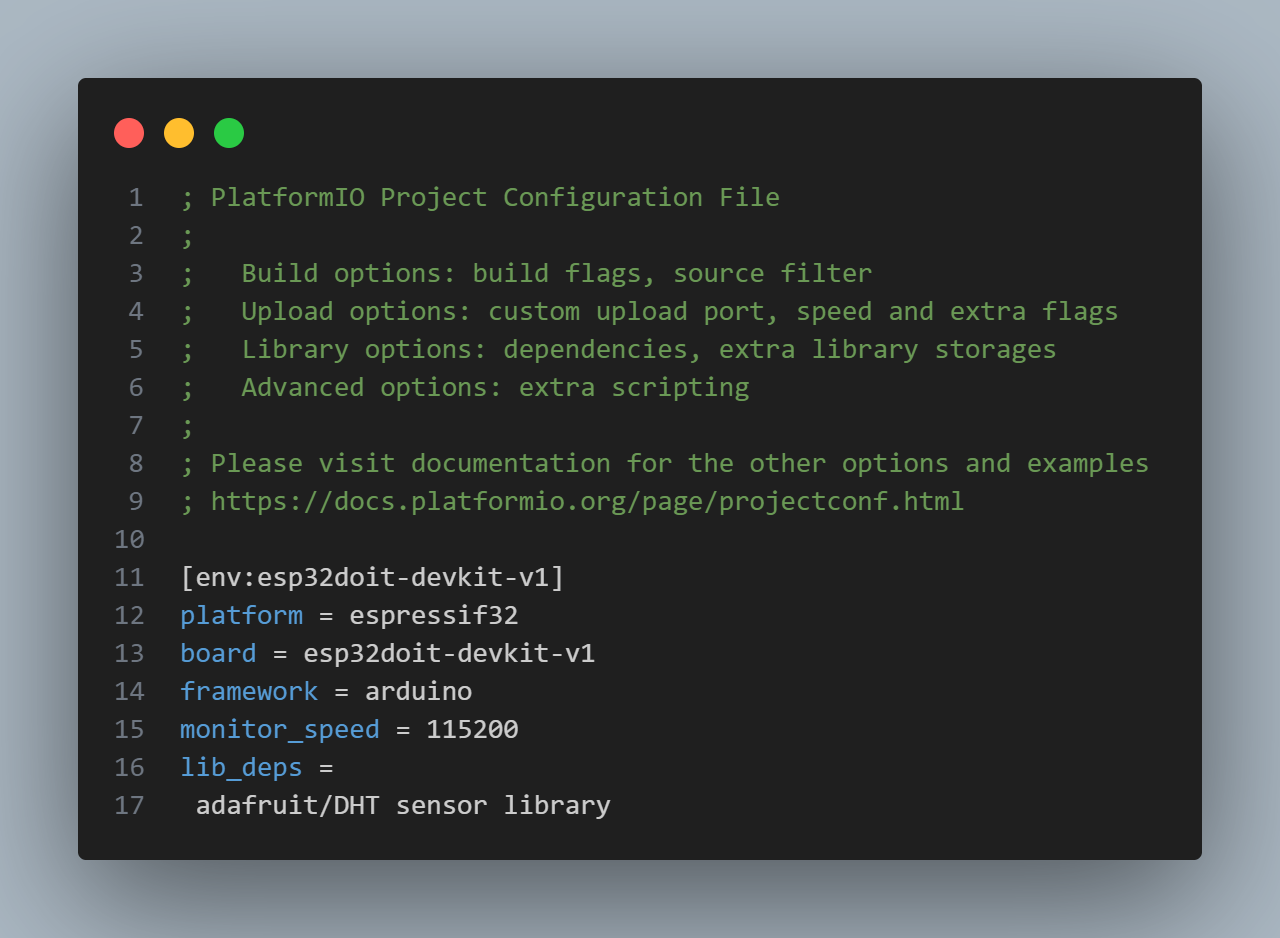
**4. Appendix**

Kode pemrograman main.cpp

Kose pemrograman diagram.json  
 

Kode pemrograman file platform.ini



Kode pemrograman file wokwi.toml

