**LAPORAN PRAKTIUM INTERNET OF THINGS (IoT)**

Fakultas Vokasi , Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

*Iffah Alayya Azzahro*

*Fakultas Vokasi, Universitas Brawijaya*

*iffaazzahro@gmail.com*

**Abstrak**

Dalam ekosistem Internet of Things (IoT), API (Application Programming Interface) memainkan peran penting dalam menghubungkan perangkat dengan server untuk pertukaran data secara real-time. Praktikum ini bertujuan untuk mengimplementasikan dan menguji akses API menggunakan simulasi perangkat IoT di WOKWI yang terhubung dengan server Laravel melalui Ngrok. API dikembangkan menggunakan Laravel 11 dan diakses melalui port 8000, yang juga digunakan oleh Ngrok untuk membuat API dapat diakses secara publik. Perangkat IoT dalam simulasi WOKWI mengirimkan data ke API Laravel, yang kemudian disimpan dalam database MySQL melalui phpMyAdmin. Hasil praktikum menunjukkan bahwa perangkat IoT simulasi dapat berkomunikasi dengan API, mengirimkan data sensor, dan menyimpannya dalam database dengan sukses. Praktikum ini memberikan wawasan tentang pengembangan API untuk IoT, penggunaan Laravel sebagai backend, serta integrasi dengan WOKWI dan Ngrok untuk memungkinkan akses data secara online.

*Keyword : Internet of Things (IoT), API, Laravel, Ngrok*

1. **Introduction (Pendahuluan)**
   1. **Latar Belakang**

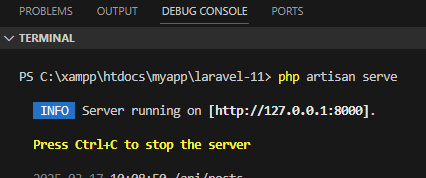
Internet of Things (IoT) memungkinkan perangkat untuk berkomunikasi dengan sistem backend melalui API, memungkinkan otomatisasi dan pertukaran data yang efisien. Dalam pengembangan IoT, diperlukan infrastruktur backend yang dapat menerima, memproses, dan menyimpan data yang dikirimkan oleh perangkat. Laravel sebagai framework PHP menawarkan solusi backend yang fleksibel dan aman untuk menangani komunikasi data perangkat IoT.

Dalam praktikum ini, perangkat IoT disimulasikan menggunakan WOKWI, yang memungkinkan pengujian tanpa perangkat keras fisik. API dikembangkan menggunakan Laravel 11, yang berjalan di port 8000, dan diakses menggunakan Ngrok untuk memungkinkan koneksi dari luar jaringan lokal. Data yang dikirim oleh perangkat simulasi akan dikirim ke API Laravel dan disimpan dalam database MySQL yang dikelola melalui phpMyAdmin.

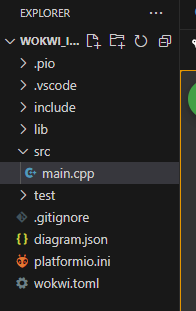
Praktikum ini bertujuan untuk memberikan pemahaman kepada mahasiswa tentang bagaimana perangkat IoT dapat mengirimkan data ke server backend melalui API, bagaimana Laravel digunakan untuk membangun API yang dapat menangani permintaan dari perangkat, serta bagaimana Ngrok digunakan untuk membuka akses API ke internet. Dengan menggunakan simulasi WOKWI, mahasiswa juga dapat menguji fungsionalitas perangkat IoT secara virtual sebelum diterapkan pada perangkat fisik.

* 1. **Tujuan Eksperimen**

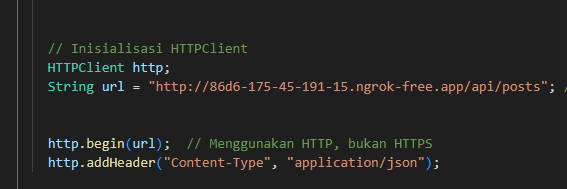
1. Memahami konsep API dalam sistem IoT, khususnya dalam pertukaran data antara perangkat dan server.
2. Mengembangkan API menggunakan Laravel 11 untuk menerima data dari perangkat IoT simulasi.
3. Mensimulasikan perangkat IoT menggunakan WOKWI dan menghubungkannya dengan API Laravel.
4. Menggunakan Ngrok sebagai tunneling service untuk membuat API dapat diakses secara online melalui port 8000.
5. Mengirim dan menyimpan data ke database MySQL menggunakan phpMyAdmin melalui API Laravel.
6. Menguji komunikasi antara perangkat IoT simulasi dengan server dan memastikan data dapat tersimpan di database dengan benar.
7. **Methodology (Metodologi)**
   1. **Tools & Materials (Alat dan Bahan)**
8. Laptop
9. Internet
10. Visual Studio Code
11. Website ( Wokwi.com)
12. Ngrok
13. MySQL
    1. **Langkah Implementasi**
14. Jalankan project Laravel 11 yang sudah dibuat sebelumnya dengan perintah **php artisan serve**



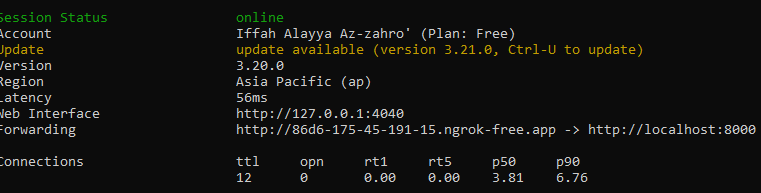
1. Membuat file baru wokwi simulator di platform.io



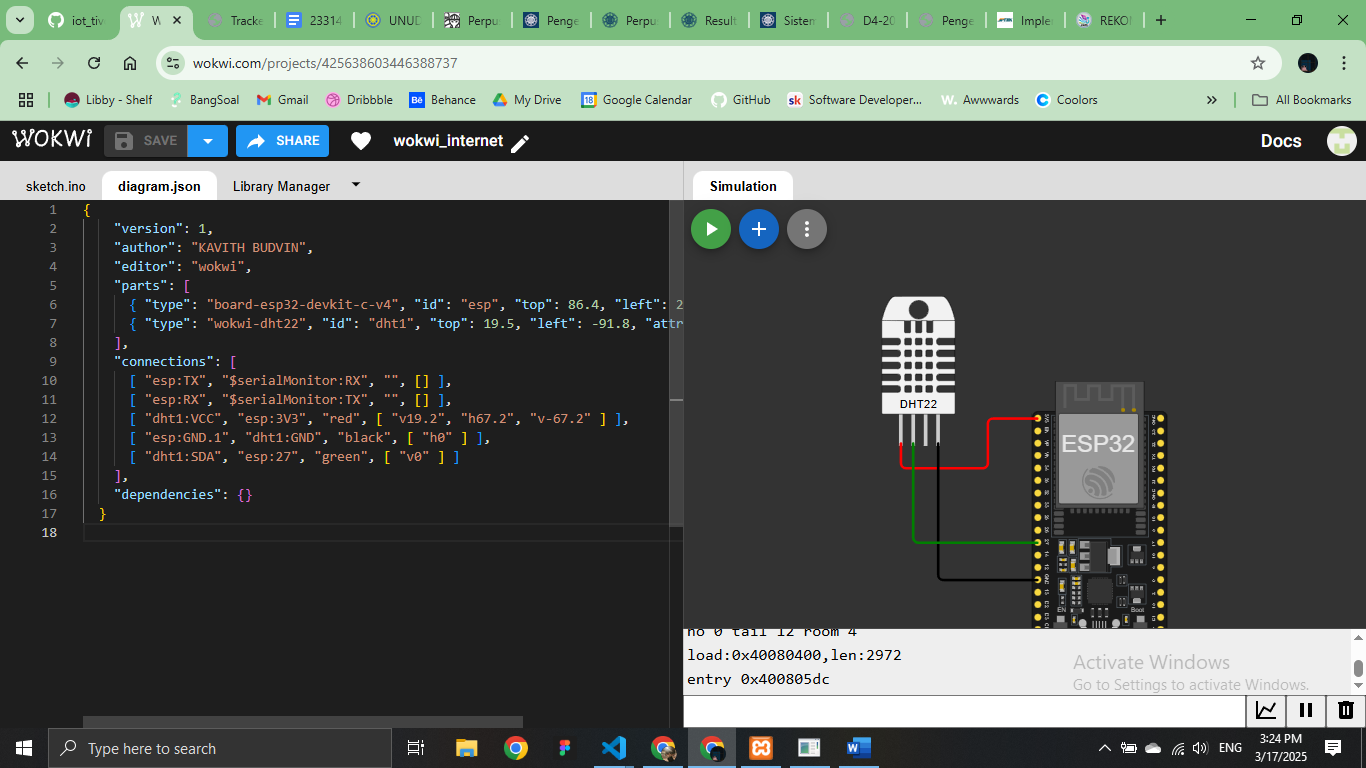
1. Kemudian ubah isi file **main.cpp** lalu buat file **wokwi.toml dan diagram.json** dengan kode yang ada di modul. Untuk URL server disesuaikan dengan URL yang ada di ngrok



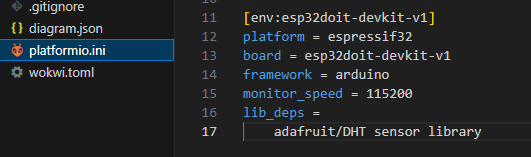
1. Pastikan ngrok berjalan pada port 8000 karena pada project Laravel menggunakan port 8000. Jalankan ngrok dengan perintah **ngrok http --scheme=http 8000**



1. Selanjutnya, melakukan modifikasi simulasi dengan menambahkan rangkaian sensor DHT22 pada ESP32. Kemudian salin kode **diagram.json** ke file diagram.json yang ada di vscode.

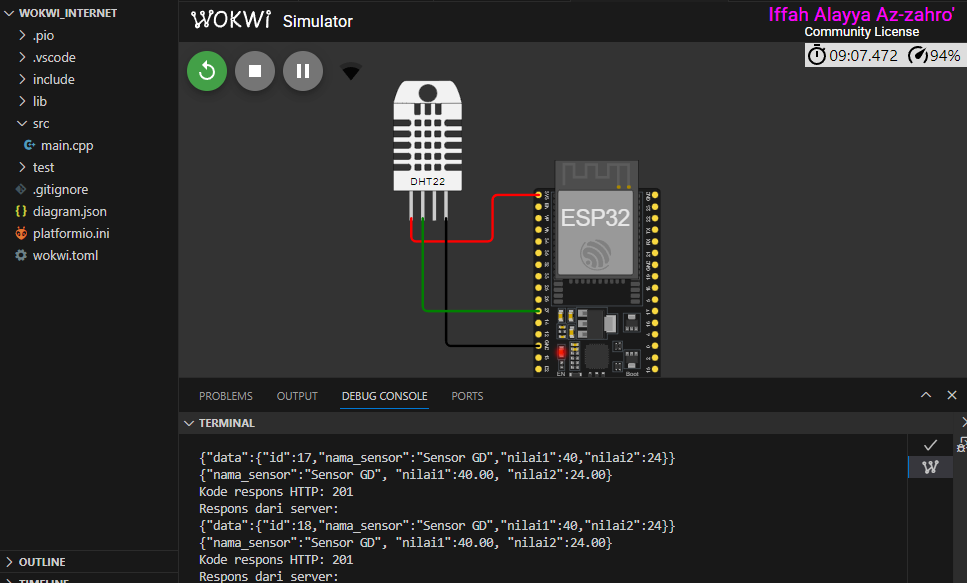


1. Kemudian ubah file **platform.io** dengan menambahkan 2 setting yaitu **monitor speed dan lib\_deps**

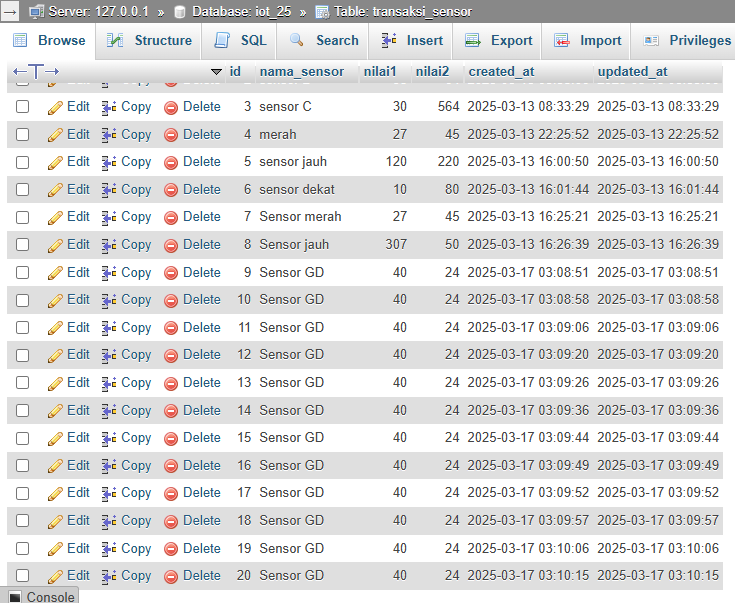


1. Setelah itu, modifikasi file **main.cpp** dengan code yang ada di modul dan juga sesuaikan URL yang ada di ngrok
2. Lakukan simulasi dengan build file **main.cpp,** setelah itu jalankan simulasi dengan perintah

**> Wokwi Start Simulator**

****

1. Lakukan pengecekan database pada **phpMyAdmin,** pastikan sudah masuk dan tersimpan pada database yg sudah dibuat sebelumnya

****

1. **Results and Discussion (Hasil dan Pembahasan)**
2. **Implementasi API dengan Laravel 11**

API dikembangkan menggunakan framework Laravel 11, yang berjalan di port 8000. Pengujian awal dilakukan untuk memastikan endpoint dapat menerima data dari perangkat IoT yang disimulasikan menggunakan WOKWI. Endpoint API dikonfigurasi untuk menerima data berbentuk JSON yang dikirim oleh perangkat simulasi, dan Laravel mengolah serta menyimpannya ke dalam database MySQL.

Pengujian dilakukan dengan mengirimkan request POST dan GET ke endpoint yang telah dibuat. Hasil pengujian menunjukkan bahwa API mampu menerima data dengan format yang sesuai dan memberikan respons dalam bentuk status kode HTTP 200 (OK) dan HTTP 201 (created).

1. **Simulasi Perangkat IoT Menggunakan WOKWI**

WOKWI digunakan untuk mensimulasikan perangkat IoT yang mengirimkan data sensor ke API Laravel. Kode program untuk simulasi perangkat ditulis menggunakan bahasa pemrograman C/C++ dengan library HTTP client untuk mengirim data melalui protokol HTTP.

Dalam pengujian, perangkat simulasi dikonfigurasi untuk mengirim data setiap 10 detik ke server. Hasil simulasi menunjukkan bahwa data dapat dikirim dan diterima secara real-time, serta disimpan dalam database tanpa mengalami kehilangan data. Hal ini menunjukkan bahwa komunikasi antara perangkat simulasi dengan server backend berjalan dengan baik.

1. **Penggunaan Ngrok untuk Akses API Secara Online**

Ngrok digunakan sebagai layanan tunneling untuk membuka akses API Laravel ke internet melalui port 8000. Konfigurasi dilakukan dengan menjalankan perintah: ngrok http 8000

Setelah dijalankan, Ngrok memberikan URL publik yang dapat digunakan untuk mengakses API dari luar jaringan lokal. Pengujian dilakukan dengan mengakses API melalui URL yang diberikan oleh Ngrok, dan hasilnya menunjukkan bahwa API dapat diakses dari perangkat simulasi yang berjalan di luar jaringan lokal.

1. **Penyimpanan Data ke Database MySQL melalui phpMyAdmin**

Database MySQL dikelola menggunakan phpMyAdmin untuk memverifikasi penyimpanan data yang dikirim oleh perangkat IoT simulasi. Setelah beberapa kali pengujian, data berhasil disimpan dengan format yang sesuai. Berikut adalah contoh data yang tersimpan dalam database:

Hasil pengujian menunjukkan bahwa setiap data yang dikirim dari perangkat simulasi berhasil masuk ke database tanpa duplikasi atau kehilangan data.

1. **Kesimpulan**

Berdasarkan hasil pengujian, sistem API yang dikembangkan menggunakan Laravel 11 dapat berfungsi dengan baik dalam menerima data dari perangkat IoT simulasi yang berjalan di WOKWI. Penggunaan Ngrok memungkinkan API dapat diakses dari luar jaringan lokal, dan data yang dikirim berhasil disimpan dalam database MySQL melalui phpMyAdmin.

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

45. // Check if any reads failed and exit early (to try again).
46. if (isnan(h) || isnan(t)) {
47. Serial.println(F("Failed to read from DHT sensor!"));
48. return;
49. }
51. // Compute heat index in Celsius (isFahreheit = false)
52. float hic = dht.computeHeatIndex(t, h, false);
54. // Inisialisasi HTTPClient
55. HTTPClient http;
56. String url = "http://86d6-175-45-191-15.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar
57. http.begin(url);  // Menggunakan HTTP, bukan HTTPS
58. http.addHeader("Content-Type", "application/json");
60. String payload = "{\"nama\_sensor\":\"Sensor GD\", \"nilai1\":" + String(h) + ", \"nilai2\":" + String(t) + "}";
61. Serial.println(payload);  // Untuk melihat apakah payload sudah terbentuk dengan benar
62. // Kirim POST request
63. int httpResponseCode = http.POST(payload);
65. // Tampilkan kode respons HTTP
66. Serial.print("Kode respons HTTP: ");
67. Serial.println(httpResponseCode);
68. // Tampilkan respons dari server jika request berhasil
69. if (httpResponseCode == 200 || httpResponseCode == 201) {
70. String response = http.getString();
71. Serial.println("Respons dari server:");
72. Serial.println(response);
73. } else {
74. Serial.println("Gagal mengirim data");
75. }
76. // Tutup koneksi HTTP
77. http.end();
78. }
79. }