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

**Fakultas Vokasi, Universitas Brawijaya**

# **Praktik BAB 15 Membuat Tampilan Interface Web Dashboard IoT**

Azizah Nur Istiqomah

Fakultas Vokasi, Universitas Brawijaya

azizahnuristiqomah456@gmail.com

**Abstrak**

Internet of Things (IoT) adalah paradigma teknologi yang memungkinkan perangkat fisik untuk saling terhubung, bertukar data, dan dikendalikan secara jarak jauh melalui jaringan internet. Dalam praktik ini, dirancang dan diimplementasikan sebuah sistem IoT yang terintegrasi dengan Application Programming Interface (API) menggunakan framework Laravel. Laravel digunakan sebagai backend untuk menerima dan mengelola data yang dikirimkan dari perangkat IoT, serta menyediakan endpoint komunikasi yang dapat diakses oleh antarmuka pengguna. Untuk menghubungkan server lokal ke jaringan publik secara aman dan praktis, digunakan layanan tunneling *Ngrok*, sehingga perangkat IoT yang berada di luar jaringan lokal tetap dapat mengakses API secara real-time.

Sebagai bagian dari sistem pemantauan, dibangun sebuah dashboard monitoring berbasis web yang dikembangkan menggunakan HTML dan CSS murni sebagai antarmuka pengguna. Dashboard ini menampilkan data yang diterima dari perangkat IoT dalam bentuk tabel, teks dinamis, dan elemen visual sederhana seperti indikator warna untuk status perangkat. Penggunaan HTML dan CSS memungkinkan pembuatan antarmuka yang ringan dan responsif tanpa ketergantungan pada pustaka pihak ketiga. Pembaruan data pada dashboard dilakukan melalui pemanggilan API secara berkala menggunakan JavaScript dasar, yang mengambil data terbaru dan menampilkannya secara otomatis tanpa memuat ulang halaman.

Sistem ini juga dilengkapi dengan fitur penyimpanan data historis di database, validasi data yang masuk, serta visualisasi status perangkat berdasarkan ambang batas tertentu. Hasil dari implementasi ini menunjukkan bahwa kombinasi Laravel sebagai backend, Ngrok sebagai jembatan koneksi global, dan HTML-CSS sebagai frontend dapat menghasilkan sistem pemantauan IoT yang sederhana, fungsional, dan mudah dikembangkan. Proyek ini memberikan contoh konkret bagaimana komponen open-source dapat digunakan untuk membangun solusi IoT yang efektif, bahkan dengan teknologi frontend yang minimalis.

*Kata Kunci: API, HTML, CSS, Laravel, Ngrok.*

**1.** **PENDAHULUAN**

**1.1.** **Latar Belakang**

Perkembangan teknologi Internet of Things (IoT) telah membawa perubahan besar dalam cara manusia berinteraksi dengan lingkungan dan perangkat di sekitarnya. IoT memungkinkan berbagai perangkat fisik—seperti sensor, aktuator, dan mikrokontroler—untuk saling terhubung dan bertukar data melalui jaringan internet. Dalam sistem semacam ini, keberadaan backend yang mampu menerima, memproses, dan menyimpan data dari perangkat IoT secara efisien menjadi sangat krusial. Salah satu komponen utama dari sistem backend ini adalah *Application Programming Interface* (API), yang bertindak sebagai jembatan komunikasi antara perangkat keras (hardware) dan aplikasi perangkat lunak (software).

API memungkinkan perangkat IoT untuk mengirimkan data sensor, menerima perintah kontrol, dan melakukan pertukaran informasi secara real-time. Agar komunikasi data dapat berjalan secara aman, efisien, dan terstruktur, diperlukan arsitektur API yang handal. Untuk tujuan tersebut, Laravel—sebuah framework PHP modern—digunakan dalam praktik ini karena kemampuannya dalam membangun RESTful API secara cepat dan terorganisir. Laravel versi 11, yang digunakan dalam praktik ini, menawarkan berbagai fitur terbaru seperti sistem routing yang fleksibel, middleware keamanan, validasi input, dan integrasi basis data yang kuat, menjadikannya pilihan ideal untuk pengembangan sistem backend pada proyek berbasis IoT.

Salah satu tantangan utama dalam mengembangkan API secara lokal adalah keterbatasan akses dari jaringan luar ke server lokal pengembang. Untuk mengatasi hal ini, digunakan layanan *tunneling* seperti **Ngrok**, yang berfungsi untuk menjembatani koneksi antara internet dan server lokal. Dengan menggunakan Ngrok, pengembang dapat memperoleh URL publik yang dapat langsung diakses oleh perangkat IoT, tanpa harus melakukan *deployment* ke server produksi. Hal ini mempercepat proses pengujian dan debugging dalam tahap awal pengembangan sistem.

Namun, keberhasilan sistem IoT tidak hanya ditentukan oleh kemampuan backend dalam mengelola data, tetapi juga oleh bagaimana data tersebut disajikan dan dimonitor oleh pengguna. Oleh karena itu, dalam praktik ini juga dikembangkan **dashboard monitoring berbasis web** menggunakan **HTML dan CSS murni**, tanpa framework frontend modern. Dashboard ini dirancang untuk menampilkan data sensor secara real-time, dalam format yang mudah dipahami, seperti tabel, indikator status, dan teks dinamis. Dengan menggunakan HTML dan CSS sebagai dasar, antarmuka ini menjadi ringan, cepat dimuat, dan mudah dikustomisasi. Data dari API ditarik secara berkala menggunakan JavaScript dasar (*vanilla JS*) untuk memperbarui tampilan tanpa perlu *refresh* halaman secara manual.

Pengembangan dashboard ini bertujuan memberikan gambaran menyeluruh tentang bagaimana sistem IoT bekerja secara utuh, mulai dari pengambilan data oleh perangkat, pengiriman melalui API, hingga visualisasi data oleh pengguna akhir. Melalui praktik ini, mahasiswa tidak hanya memperoleh pemahaman teknis mengenai Laravel dan Ngrok, tetapi juga dibekali keterampilan dalam membuat antarmuka monitoring yang fungsional dan responsif. Pengalaman langsung dalam membangun dan mengintegrasikan sistem ini menjadi bekal penting untuk menghadapi tantangan dunia nyata di bidang IoT, yang menuntut ketelitian, keterpaduan sistem, serta kemampuan untuk menyampaikan informasi secara efektif kepada pengguna.

**1.2.** **Tujuan Eksperimen**

Tujuan dari praktik eksperimen ini adalah untuk:

1. Mengembangkan pemahaman mahasiswa dalam membangun dan mengimplementasikan API menggunakan Laravel 11 sebagai fondasi sistem backend yang mendukung komunikasi data IoT secara real-time.
2. Meningkatkan kemampuan mahasiswa dalam menghubungkan sistem backend dengan perangkat IoT menggunakan layanan tunneling seperti Ngrok, guna mengatasi keterbatasan akses jaringan lokal.
3. Melatih mahasiswa dalam melakukan konfigurasi environment Laravel, pembuatan routing dan controller, validasi data, serta pengolahan dan penyimpanan data yang diterima dari perangkat IoT.
4. Memberikan pengalaman langsung dalam pengujian komunikasi dua arah antara perangkat IoT dan server backend, termasuk pemantauan stabilitas koneksi dan keakuratan data yang diterima.
5. Menumbuhkan pemahaman tentang pentingnya integrasi antar sistem, keamanan dalam pertukaran data, dan efisiensi dalam pengelolaan komunikasi antara perangkat keras dan perangkat lunak dalam sistem IoT.
6. Membekali mahasiswa dengan keterampilan dalam merancang dan mengembangkan dashboard monitoring berbasis HTML dan CSS, sebagai antarmuka visual yang menampilkan data sensor secara real-time dengan tampilan yang informatif, ringan, dan mudah diakses.
7. Mendorong mahasiswa untuk memahami peran antarmuka pengguna dalam sistem IoT secara menyeluruh, termasuk bagaimana menyajikan data yang kompleks secara sederhana dan dapat diinterpretasikan oleh pengguna akhir.

**2.** **METODOLOGI**

### **2.1 Alat dan Bahan**

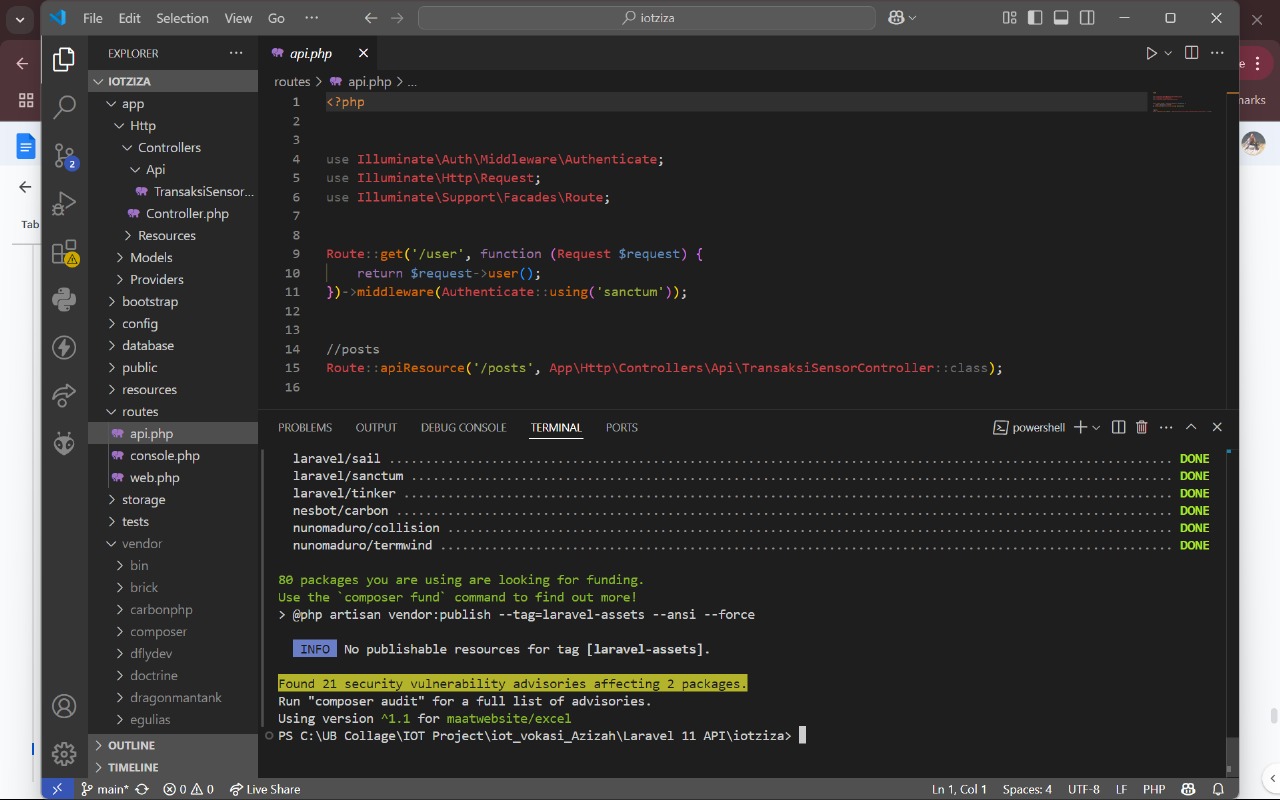
1. **MySQL** – Sebagai sistem manajemen basis data untuk menyimpan data sensor dari perangkat IoT.
2. **Laravel 11** – Framework PHP yang digunakan untuk membangun RESTful API sebagai backend.
3. **Postman** – Alat untuk menguji dan memvalidasi endpoint API secara manual.
4. **Ngrok** – Layanan tunneling untuk menghubungkan server lokal dengan jaringan internet.
5. **Visual Studio Code (VS Code)** – Editor kode untuk mengembangkan backend Laravel dan dashboard frontend.
6. **Browser (Chrome/Firefox)** – Untuk mengakses dan menguji tampilan dashboard monitoring.
7. **HTML, CSS, JavaScript (vanilla JS)** – Untuk membangun antarmuka dashboard monitoring berbasis web.
8. **C++ (melalui Arduino IDE atau Wokwi)** – Digunakan untuk pemrograman perangkat IoT (simulasi atau nyata).

### **2.2 Langkah Implementasi**

1. **Perancangan Sistem** Menyusun konsep dan logika kerja sistem IoT, mulai dari alur data sensor hingga visualisasi pada dashboard. Skema alur data dirancang mencakup interaksi antara perangkat IoT, API Laravel, dan dashboard web.
2. **Menentukan Komponen yang Digunakan**
   * Backend: Laravel 11
   * Database: MySQL
   * Testing API: Postman
   * Visualisasi: HTML, CSS, JavaScript
   * Tunneling: Ngrok
   * Perangkat IoT: Mikrokontroler (fisik/simulasi Wokwi)
3. **Pembuatan dan Konfigurasi Backend API Laravel**
   * Membuat database di phpMyAdmin.
   * Menyiapkan model TransaksiSensor.php
   * Membuat controller dan resource API menggunakan perintah artisan.
   * Menambahkan endpoint untuk menerima data sensor dan menyimpannya ke database.
   * Mengaktifkan fitur CORS jika diperlukan untuk akses dari frontend.
   * Menguji endpoint API menggunakan Postman untuk memastikan data dapat disimpan dengan benar.
4. **Pengembangan Program Perangkat IoT (C++)**
   * Menulis program untuk mikrokontroler (fisik atau simulasi Wokwi) agar dapat mengirimkan data sensor ke URL API Laravel.
   * Melakukan pengujian pengiriman data ke server dan memastikan penerimaan di database berjalan dengan baik.
5. **Menguji API dan Validasi Penyimpanan Data**
   * Melakukan pengujian insert data sensor ke database melalui API secara manual dan otomatis dari perangkat IoT.
   * Memastikan data baru berhasil masuk ke tabel database dan sesuai dengan format yang diharapkan.
6. **Membuka Akses API Menggunakan Ngrok**
   * Menjalankan Ngrok untuk mengekspos server Laravel ke internet.
   * Mengonfigurasi perangkat IoT atau Wokwi untuk mengakses endpoint melalui URL Ngrok.
   * Memastikan koneksi dan komunikasi berjalan lancar.
7. **Pengembangan Dashboard Monitoring (Frontend)**
   * Membuat halaman dashboard menggunakan HTML dan CSS untuk menampilkan data sensor secara real-time.
   * Menggunakan JavaScript dasar (AJAX/fetch API) untuk mengambil data terbaru dari API setiap interval waktu tertentu.
   * Menyusun tampilan dashboard dalam bentuk tabel dinamis, teks informasi, dan indikator status warna.
   * Menambahkan fitur pembaruan otomatis (*auto-refresh*) tanpa reload halaman.
8. **Integrasi dan Pengujian Sistem**
   * Melakukan pengujian penuh dari pengambilan data sensor oleh perangkat, pengiriman ke API Laravel melalui Ngrok, hingga visualisasi di dashboard monitoring.
   * Memastikan sistem berjalan stabil, data tersimpan dan ditampilkan secara akurat, serta dashboard dapat diakses dari browser secara lokal maupun publik.

**3.** **HASIL DAN PEMBAHASAN**

**3.1.** **Hasil Eksperimen**

****

****