LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Konfigurasi dan Deteksi Perangkat ESP32 di macOS menggunakan PlatformIO dan Visual Studio Code**

*Salsabilla Permata Dewi – 233140701111016*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: salsabillapermata02@gamil.com*

**Abstract** (Abstrak)

Laporan ini membahas pengembangan Web Dashboard IoT menggunakan framework Laravel untuk menampilkan dan memantau data dari puluhan hingga ratusan sensor secara real-time. Tujuan utama pembuatan dashboard ini adalah menyediakan antarmuka yang terorganisir, responsif, dan mudah digunakan guna memonitor performa perangkat IoT serta mengelola data sensor secara efisien.

*Keywords: IoT, real-time, Web Dashboard.*

1. **Pendahuluan**
2. **Latar Belakang**

Perkembangan Internet of Things (IoT) menjadi pilar utama transformasi digital, dengan penerapan luas di sektor industri, pertanian, perkotaan, hingga rumah pintar. IoT memungkinkan pengumpulan data masif dari sensor yang terhubung, namun kompleksitas infrastruktur menghadirkan tantangan dalam manajemen dan visualisasi data. Tanpa antarmuka terintegrasi, data sering kali tersebar, tidak terstruktur, dan sulit dipantau secara real-time, menurunkan efisiensi respons terhadap anomali.

Solusi dashboard IoT yang ada saat ini masih memiliki keterbatasan, seperti tampilan yang tidak intuitif, kurangnya skalabilitas, dan pembaruan data yang tidak real-time. Hal ini menekankan pentingnya pengembangan antarmuka web yang andal, skalabel, dan mudah digunakan.

Penggunaan Laravel sebagai backend dalam pengembangan Web Dashboard IoT menjadi solusi strategis. Laravel mendukung pengelolaan database, autentikasi aman, serta integrasi dengan front-end modern. Dengan pendekatan modular dan integrasi protokol MQTT, sistem ini mendukung komunikasi real-time antara perangkat IoT dan server.

Proyek ini bertujuan menyajikan data sensor secara real-time sekaligus menyediakan alat manajemen terpusat yang efisien. Diharapkan dashboard ini dapat menjadi model implementasi IoT skala besar dalam mendukung industri 4.0 dan smart city, serta meningkatkan kecepatan analisis, mengurangi kesalahan, dan memperkuat pengambilan keputusan berbasis data.

1. **Tujuan Eksperimen**

Eksperimen ini bertujuan untuk menguji kinerja dan fungsionalitas Web Dashboard IoT berbasis Laravel dalam menangani data dari puluhan hingga ratusan sensor secara real-time

1. **Metodelogi**
2. **Alat dan Bahan**

**●** Laptop

● Visual Studio Code

● phpMyAdmin

● XAMPP

● Herd

● Laravel 11

● Koneksi internet

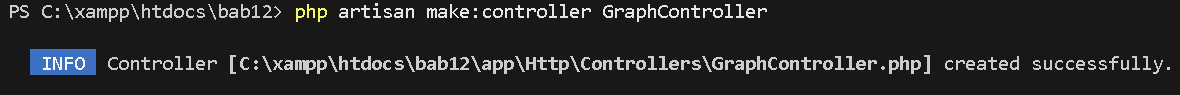
1. **Langkah Implementasi**
2. Buka Terminal di Vscode dan jalankan perintah berikut pada praktik Vscode bab 12 Laravel :

composer require maatwebsite/excel

1. Langkah berikutnya adalah membuat GraphController menggunakan perintah berikut:

php artisan make:controller GraphController

Tampilan jika berhasil



1. Setelah berhasil tambahkan code berikut pada GraphContoller:

<?php

namespace App\Http\Controllers;

use App\Exports\TransaksiSensorExport; use Maatwebsite\Excel\Facades\Excel;

use App\Models\TransaksiSensor;

class GraphController extends Controller

{

/\*\*

* Menampilkan grafik transaksi sensor.

\*

* @return \Illuminate\View\View

\*/

public function index()

{

// Mengambil data transaksi sensor

$transaksiSensors = TransaksiSensor::latest()->take(10)->get();

// Mengambil data label

$transaksiSensors->pluck('nama\_sensor');

$labels =

grafik

// Mengambil data nilai1 dan nilai2 untuk

$dataNilai1 =

$transaksiSensors->pluck('nilai1');

$dataNilai2 =

$transaksiSensors->pluck('nilai2');

return view('graph', compact('labels', 'dataNilai1', 'dataNilai2'));

}

/\*\*

\* Mengunduh data transaksi sensor dalam format

Excel

\* @return

\Symfony\Component\HttpFoundation\BinaryFileResponse

\*/

public function exportToExcel()

{

return Excel::download(new TransaksiSensorExport, 'transaksi\_sensor.xlsx');

}

}

1. Setelah itu jalankan perintah pada terminal seperti perintah berikut:

php artisan make:export TransaksiSensorExport

--model=TransaksiSensor

1. Setelah itu, edit file web.php
2. Setelah semua sudah, jalankan program tersebut dengan perintah berikut:

php artisan serve

1. **Hasil Eksperimen**

