# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Membuat Tampilan Interface**

**Web Dashboard IoT**

*Eva Latifah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: ev.tifa@gmail.com*

**Abstrak**

Perkembangan teknologi Internet of Things (IoT) memungkinkan pengumpulan dan pengelolaan data secara real-time dari berbagai perangkat yang terhubung. Dalam laporan ini dibahas proses pembuatan tampilan interface web dashboard untuk sistem IoT yang bertujuan memudahkan pengguna dalam memonitor dan mengendalikan perangkat IoT secara visual dan interaktif. Dashboard dikembangkan menggunakan teknologi web modern dengan framework Laravel sebagai backend dan berbagai library frontend untuk menampilkan data sensor dalam bentuk grafik, tabel, dan indikator status. Desain antarmuka fokus pada kemudahan penggunaan, responsivitas, dan penyajian data yang informatif sehingga dapat mendukung pengambilan keputusan pengguna secara efektif. Hasil implementasi menunjukkan bahwa dashboard mampu menampilkan data secara real-time dan menyediakan fitur kontrol yang intuitif, sehingga mempermudah manajemen perangkat IoT.

**Kata Kunci**—*Internet of Things, ESP32, Database, Sensor DHT22, Laravel API, Dashboard*

***Abstract***

*The development of Internet of Things (IoT) technology enables real-time data collection and management from various connected devices. This report discusses the process of creating a web dashboard interface for an IoT system aimed at facilitating users in monitoring and controlling IoT devices visually and interactively. The dashboard is developed using modern web technologies with the Laravel framework as the backend and various frontend libraries to display sensor data in the form of graphs, tables, and status indicators. The interface design focuses on ease of use, responsiveness, and informative data presentation to effectively support user decision-making. The implementation results show that the dashboard is capable of displaying real-time data and provides intuitive control features, thereby simplifying IoT device management.*

***Keywords****—Internet of Things, ESP32, Database, DHT22 Sensor, Laravel API, Dashboard*

**1. Introduction**

**1.1 Latar belakang**

Perkembangan teknologi Internet of Things (IoT) telah membawa perubahan signifikan dalam berbagai bidang, seperti industri, pertanian, kesehatan, dan rumah pintar. IoT memungkinkan berbagai perangkat untuk saling terhubung dan bertukar data secara otomatis, sehingga meningkatkan efisiensi dan kemudahan dalam pengelolaan sistem. Namun, pengelolaan data dan kontrol perangkat IoT secara langsung sering kali masih memerlukan antarmuka yang mudah dipahami dan digunakan oleh pengguna.

Web dashboard sebagai salah satu solusi interface visual dapat memudahkan pengguna dalam memonitor dan mengendalikan perangkat IoT secara real-time melalui tampilan yang interaktif dan informatif. Dengan menggunakan dashboard, pengguna dapat dengan cepat melihat kondisi perangkat, menganalisis data sensor, dan mengambil keputusan secara tepat tanpa harus mengakses perangkat secara manual.

Oleh karena itu, pembuatan tampilan interface web dashboard yang efektif dan responsif sangat penting untuk meningkatkan pengalaman pengguna dan mendukung fungsi sistem IoT secara optimal. Laporan ini membahas proses perancangan dan pengembangan web dashboard IoT menggunakan framework Laravel dan teknologi frontend modern untuk memenuhi kebutuhan tersebut.

**1.2 Tujuan eksperimen**

Praktikum ini bertujuan untuk:

1. Mempelajari konsep dasar dan penerapan Internet of Things (IoT) dalam sistem monitoring perangkat.
2. Mengembangkan tampilan interface web dashboard yang interaktif dan responsif untuk sistem IoT.
3. Mengimplementasikan framework Laravel sebagai backend dalam pengelolaan data IoT.
4. Menampilkan data sensor secara real-time dalam bentuk grafik, tabel, dan indikator status pada dashboard.
5. Meningkatkan kemampuan dalam perancangan antarmuka pengguna yang mudah digunakan dan informatif.

**2. Methodology**

**2.1 Tools & Materials**

1. Komputer/Laptop dengan sistem operasi Windows/Linux/MacOS
2. Laravel 12
3. MySQL
4. Web Browser

**2.2 Implementation Steps**

#### **Buka proyek laravel**

Buka folder proyek Laravel yang sudah dibuat sebelumnya menggunakan Visual Studio Code

#### **Pasang Package Excel**

composer require maatwebsite/excel

#### **Membuat GraphController.php**

php artisan make:controller GraphController

#### **Membuat TransaksiSensor.php**

php artisan make:export TransaksiSensorExport --model=TransaksiSensor

#### **Membuat file graph.blade.php pada folder resouces/views**

**3. Results and Discussion**

**3.1 Experimental Results**

Pembuatan tampilan interface web dashboard untuk sistem IoT bertujuan untuk memberikan kemudahan bagi pengguna dalam memonitor dan mengendalikan perangkat IoT secara real-time. Dalam praktik ini, dashboard dikembangkan menggunakan framework Laravel sebagai backend yang menyediakan API dan pengelolaan data, serta teknologi frontend untuk menampilkan data secara visual.

Pertama, dilakukan instalasi package maatwebsite/excel pada Laravel untuk mendukung fitur export data sensor dalam format Excel. Selanjutnya, dibuat controller GraphController yang mengatur logika pengambilan data dari basis data dan penyajian data ke tampilan.

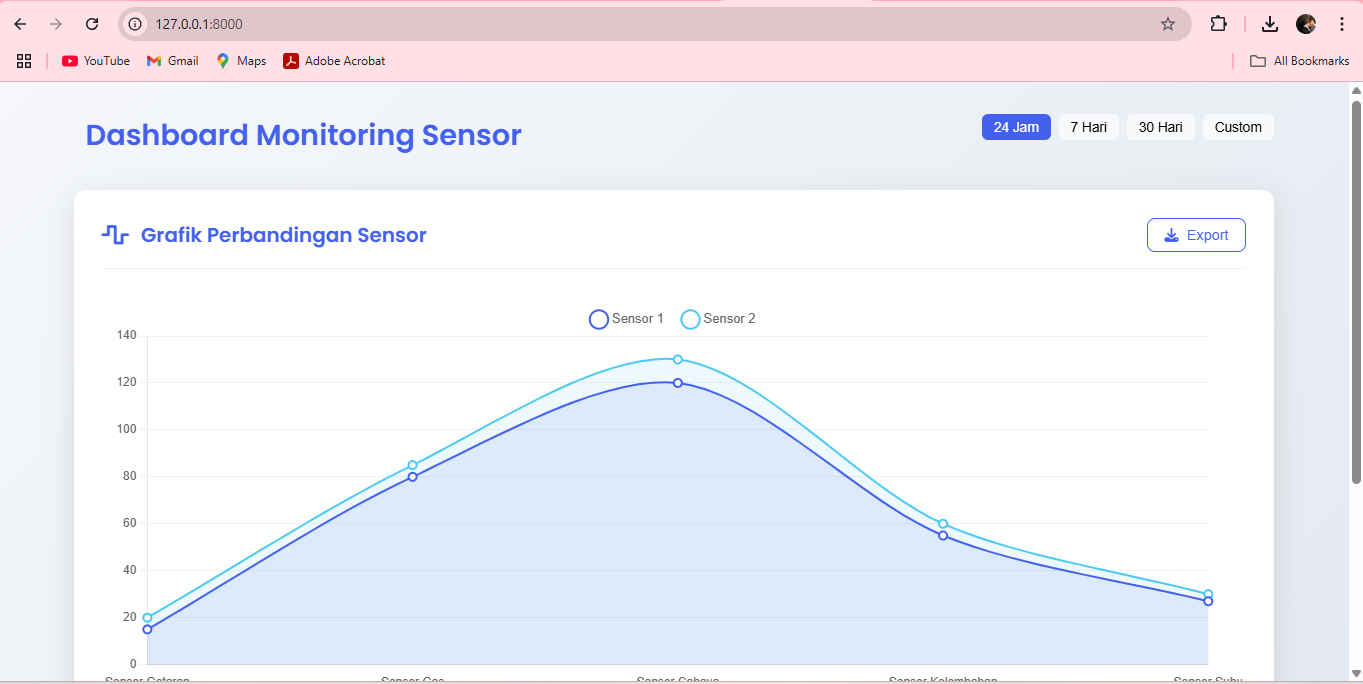
Routing pada Laravel diatur agar URL utama (/) mengarah ke metode index() di controller yang menampilkan dashboard, sementara URL /graph/export digunakan untuk mengekspor data sensor ke dalam file Excel. Hal ini memberikan kemudahan bagi pengguna untuk mengunduh dan menyimpan data secara offline.

Pada bagian tampilan, data sensor disajikan dalam bentuk grafik interaktif, tabel, dan indikator status perangkat. Grafik ini membantu pengguna untuk memahami tren dan kondisi perangkat IoT secara cepat dan intuitif. Desain interface dibuat responsif agar dapat diakses dengan baik pada berbagai perangkat seperti desktop maupun smartphone.

Hasil implementasi menunjukkan bahwa dashboard dapat menampilkan data secara real-time dan interaktif. Pengguna dapat dengan mudah memantau kondisi perangkat dan mengunduh data sensor jika diperlukan. Fitur ini sangat berguna dalam pengelolaan dan pengawasan perangkat IoT, sehingga dapat meningkatkan efektivitas pengambilan keputusan.

Secara keseluruhan, pembuatan dashboard ini berhasil memenuhi tujuan utama yaitu menyediakan antarmuka yang informatif, mudah digunakan, dan dapat diakses kapan saja. Pengembangan lebih lanjut dapat meliputi penambahan fitur notifikasi, kontrol perangkat secara langsung, dan integrasi dengan layanan cloud untuk skalabilitas yang lebih baik.

#### **Hasil Dashboard**



**4. Appendix**

1. GraphControler.php

<?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

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

// Mengambil data nilai1 dan nilai2 untuk grafik

$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. TransaksiSensorExport.php

<?php

namespace App\Exports;

use App\Models\TransaksiSensor;

use Maatwebsite\Excel\Concerns\FromCollection;

class TransaksiSensorExport implements FromCollection

{

/\*\*

\* @return \Illuminate\Support\Collection

\*/

public function collection()

{

return TransaksiSensor::all();

}

}

1. GraphBlade.php