## **PEMBUATAN TAMPILAN INTERFACE WEB DASHBOARD IoT**

*Candra Wahyu Perdana*

Vocational Faculty, Brawijaya University

Email: candraega28@gmail.com

**Abstract**

Internet of Things (IoT) merupakan teknologi yang memungkinkan perangkat untuk saling terhubung dan bertukar data melalui jaringan internet. Dalam sistem IoT, data yang dikirimkan oleh sensor perlu divisualisasikan agar mudah dipahami oleh pengguna. Pada praktikum ini, dilakukan pembuatan tampilan antarmuka (interface) web dashboard berbasis Laravel yang berfungsi untuk menampilkan data sensor dalam bentuk grafik menggunakan Chart.js. Selain itu, disediakan fitur untuk mengekspor data ke dalam format Excel menggunakan pustaka Laravel Excel. Hasil praktikum menunjukkan bahwa dashboard dapat menampilkan grafik interaktif yang merepresentasikan dua nilai sensor terbaru secara real-time serta mempermudah pengguna dalam mengunduh data untuk keperluan analisis lebih lanjut. Dengan adanya tampilan dashboard ini, pemantauan data IoT menjadi lebih efisien dan informatif.

*Keywords—ESP32, Dashboard,PlatformIO, IoT,* Laravel

**1. Introduction**

**1.1 Background**

Pada bab ini, dijelaskan proses pembuatan tampilan antarmuka (interface) Web Dashboard IoT yang menampilkan grafik data sensor serta fitur ekspor data ke dalam file Excel. Dashboard ini dibangun menggunakan framework Laravel dan memanfaatkan Chart.js untuk visualisasi data serta Maatwebsite Excel untuk ekspor data.

**1.2 Objectives**

* Membuat tampilan antarmuka web dashboard IoT menggunakan Laravel dan Chart.js.
* Menampilkan data sensor secara visual dalam bentuk grafik.
* Menambahkan fitur ekspor data sensor ke file Excel.

**2. Methodology**

**2.1 Tools & Materials**

* **Software:** PlatformIO (VSCode), Arduino Framework
* **Hardware (simulated):**
  + Laptop/PC (sebagai server lokal untuk menjalankan Laravel dan database)
  + Modul ESP32 (sebagai perangkat IoT yang mengirimkan data sensor)
  + Sensor suhu dan kelembaban (misalnya DHT11/DHT22)
  + Kabel USB (untuk menghubungkan ESP32 ke laptop/PC)

#### **2.2 Step by Step**

#### **Masuk ke Direktori Project Laravel**

#### Masuk ke folder Laravel yang telah dibuat pada praktikum sebelumnya (praktik 12).

1. **Menambahkan Paket Ekspor Excel**

Jalankan perintah berikut pada terminal untuk menginstal paket ekspor:

| composer require maatwebsite/excel |
| --- |

## **Membuat Controller**

Buat controller baru dengan nama GraphController menggunakan perintah:

| php artisan make:controller GraphController |
| --- |

1. **Menambahkan Routing**

Tambahkan routing baru pada file routes/web.php sebagai berikut:

| use App\Http\Controllers\GraphController;  Route::get('/', [GraphController::class, 'index'])->name('graph');  Route::get('/graph/export', [GraphController::class, 'exportToExcel'])->name('graph.export'); |
| --- |

1. **Menambahkan Kode pada Controller**

Tambahkan kode berikut pada file GraphController.php:

| <?php  namespace App\Http\Controllers;  use App\Exports\TransaksiSensorExport;  use Maatwebsite\Excel\Facades\Excel;  use App\Models\TransaksiSensor;  class GraphController extends Controller  {  public function index()  {  $transaksiSensors = TransaksiSensor::latest()->take(10)->get();  $labels = $transaksiSensors->pluck('nama\_sensor');  $dataNilai1 = $transaksiSensors->pluck('nilai1');  $dataNilai2 = $transaksiSensors->pluck('nilai2');  return view('graph', compact('labels', 'dataNilai1', 'dataNilai2'));  }  public function exportToExcel()  {  return Excel::download(new TransaksiSensorExport, 'transaksi\_sensor.xlsx');  }  } |
| --- |

1. **Membuat File Ekspor**

Buat file ekspor menggunakan perintah berikut:

| php artisan make:export TransaksiSensorExport --model=TransaksiSensor |
| --- |

Isi file TransaksiSensorExport.php

| <?php  namespace App\Exports;  use App\Models\TransaksiSensor;  use Maatwebsite\Excel\Concerns\FromCollection;  class TransaksiSensorExport implements FromCollection  {  public function collection()  {  return TransaksiSensor::all();  }  } |
| --- |

1. **Membuat Tampilan Antarmuka Dashboard**

Buat file baru dengan nama graph.blade.php pada folder resources/views. File ini digunakan untuk menampilkan grafik menggunakan Chart.js dan tombol ekspor data ke Excel.

Cuplikan isi file graph.blade.php

| <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>  <canvas id="sensorChart"></canvas>  <script>  const ctx = document.getElementById('sensorChart').getContext('2d');  const sensorChart = new Chart(ctx, {  type: 'line',  data: {  labels: @json($labels),  datasets: [  {  label: 'Nilai 1',  data: @json($dataNilai1),  backgroundColor: 'rgba(67, 97, 238, 0.2)',  borderColor: 'rgba(67, 97, 238, 1)',  borderWidth: 2,  tension: 0.3  },  {  label: 'Nilai 2',  data: @json($dataNilai2),  backgroundColor: 'rgba(76, 201, 240, 0.2)',  borderColor: 'rgba(76, 201, 240, 1)',  borderWidth: 2,  tension: 0.3  }  ]  },  options: {  responsive: true,  maintainAspectRatio: false  }  });  </script> |
| --- |

## **3. Results & Discussion**

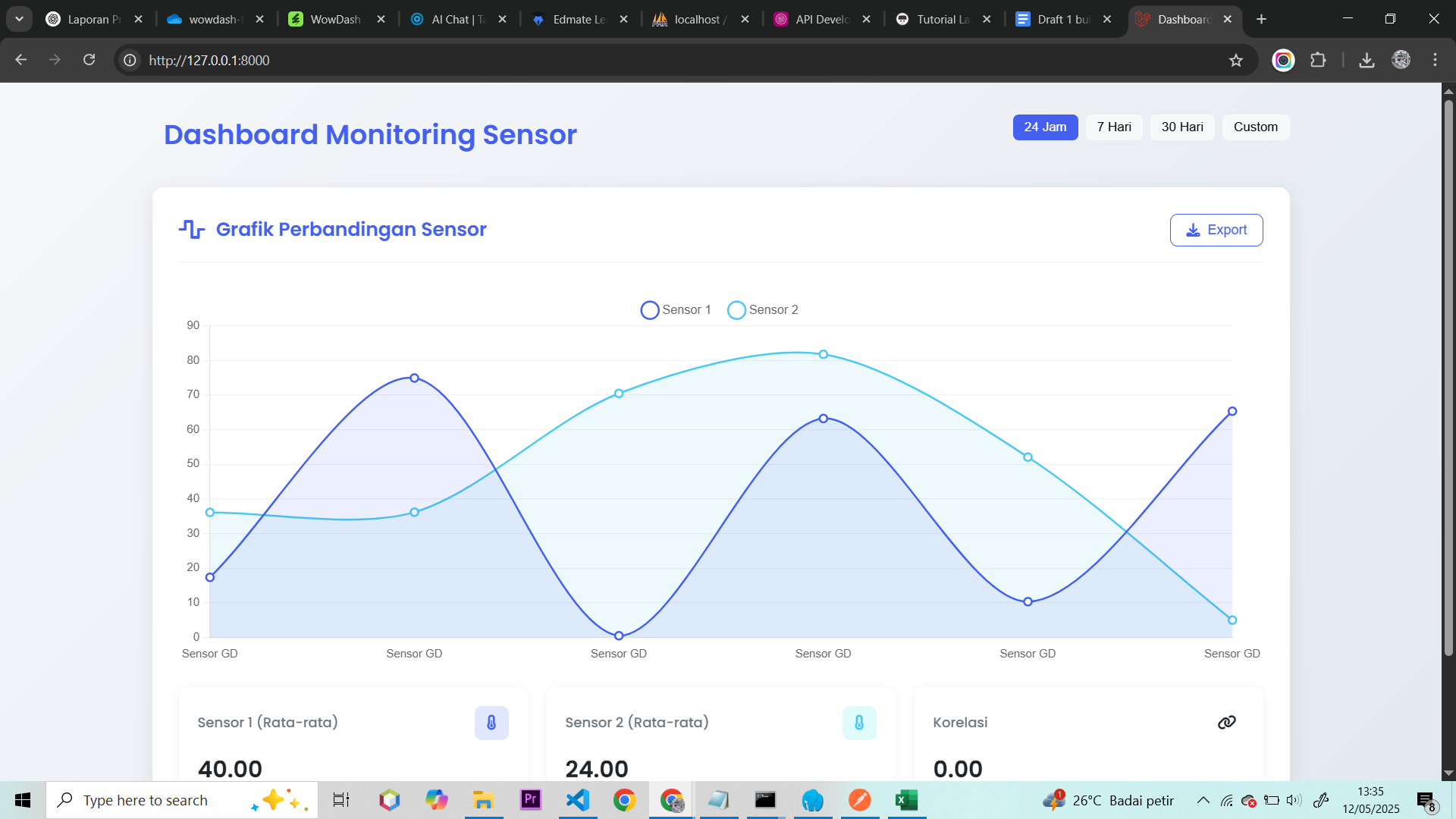
**3.1 Observations**

Setelah seluruh langkah dilaksanakan, maka halaman utama (/) akan menampilkan grafik data sensor dari database dalam bentuk grafik garis (line chart). Data yang ditampilkan meliputi dua nilai sensor terakhir dari 10 data terbaru. Selain itu, terdapat tombol **Export to Excel** yang dapat digunakan untuk mengunduh seluruh data sensor dalam format .xlsx.

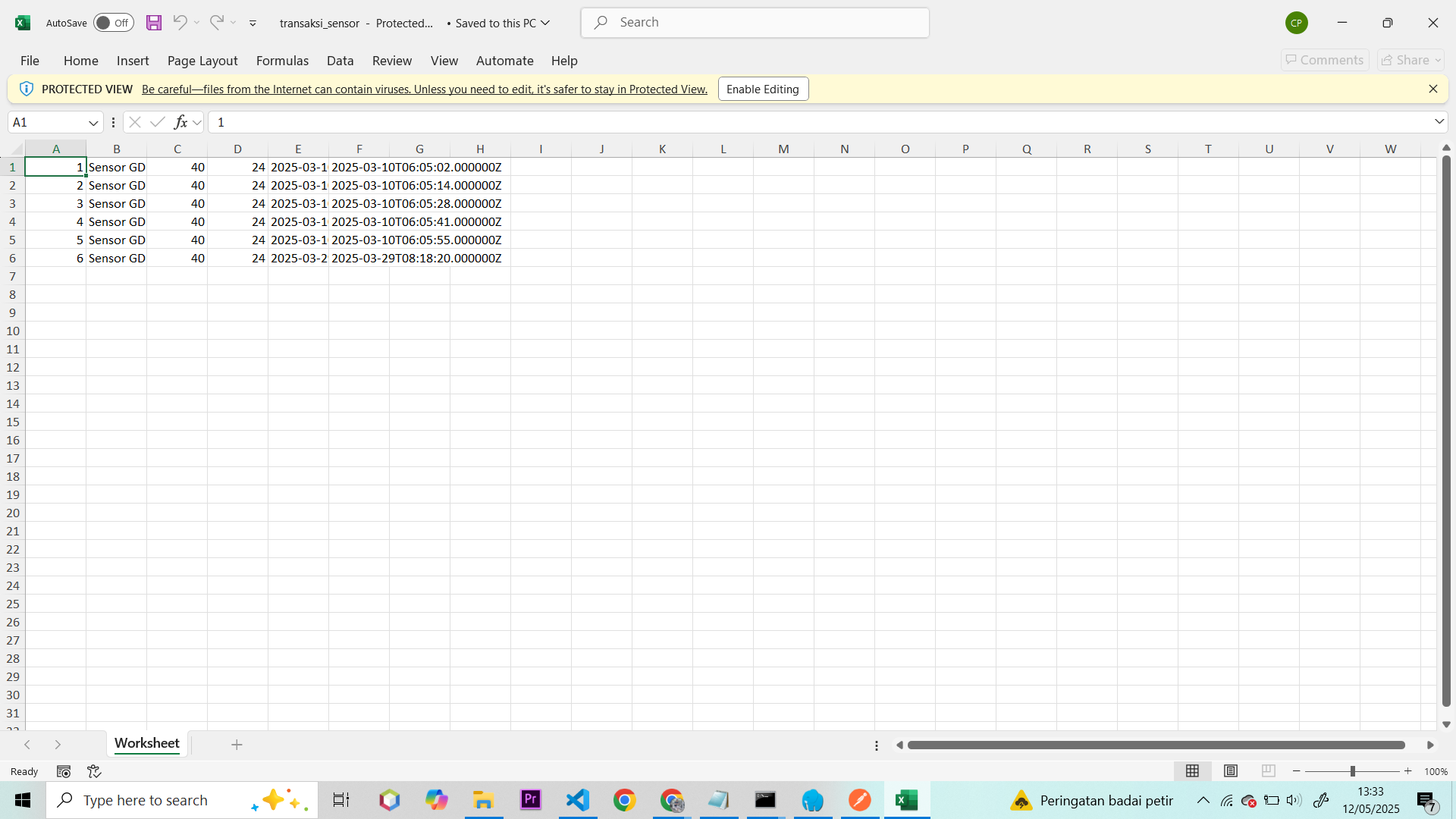
Hasil grafik dapat berubah secara dinamis sesuai dengan data terbaru yang masuk ke dalam database.

### **3.3 Photo Evidence**

**Figure 1.** Tampilan Dashboard



**Figure 2.** Hasil Ekstrak file Excel



## **4. Conclusion**

Pada praktik ini telah berhasil dibuat sebuah dashboard berbasis web untuk menampilkan data sensor IoT dalam bentuk grafik menggunakan Chart.js. Selain itu, ditambahkan juga fitur ekspor data sensor ke format Excel menggunakan Laravel dan package maatwebsite/excel. Tampilan dashboard ini memudahkan pengguna dalam memantau dan menganalisis data secara visual dan efisien.