

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

(Simulasi Praktikum Pembuatan Dashboard iot untuk menampilkan data dari database)



Muhammad Kadavi
Fakultas Vokasi, Universitas Brawijaya
Email: kadavi2945@student.ub.ac.id

Abstrak

Tujuan dari praktikum ini adalah untuk mengimplementasikan pengembangan sebuah dashboard berbasis Internet of Things (IoT) yang menampilkan data dari sensor yang disimpan dalam database. Fokus dari praktikum ini adalah penggunaan framework Laravel untuk mengembangkan aplikasi web yang menampilkan grafik dan mengekspor data ke Microsoft Excel. Dengan mengintegrasikan database IoT yang menyimpan transaksi data dari sensor, dashboard ini memberikan visualisasi dan analisis data yang mudah diakses oleh semua orang.

Latar Belakang

Dalam era teknologi yang terus berkembang, sistem Internet of Things (IoT) semakin banyak digunakan untuk memonitor berbagai perangkat secara real-time. Salah satu tantangan yang dihadapi adalah pengolahan dan pemantauan data sensor yang dihasilkan oleh perangkat IoT. Oleh karena itu, dashboard yang mampu menampilkan data sensor secara interaktif sangat diperlukan untuk memudahkan pengguna dalam mengambil keputusan berbasis data yang valid dan terkini. Praktikum ini bertujuan untuk merancang dan membangun sistem dashboard yang dapat menampilkan grafik dari data yang tersimpan dalam database menggunakan Laravel dan Chart.js.

Tujuan

Tujuan dari praktikum ini adalah untuk mengembangkan dan mengimplementasikan sebuah dashboard monitoring yang dapat menampilkan data yang diambil dari sensor IoT yang tersimpan dalam database. Dengan menggunakan framework Laravel dan berbagai teknologi terkait seperti Chart.js dan Maatwebsite Excel, praktikum ini bertujuan untuk:

1. Membangun aplikasi web berbasis Laravel yang mampu mengambil dan menampilkan data transaksi sensor dalam bentuk grafik yang interaktif.
2. Menyediakan fitur ekspor data dalam format Excel untuk memudahkan analisis data lebih lanjut.
3. Mengintegrasikan data dari sensor IoT yang tersimpan dalam database ke dalam dashboard yang mudah dipahami dan digunakan oleh pengguna.
4. Meningkatkan kemampuan pengelolaan dan pemantauan data sensor secara real-time dengan menampilkan informasi yang relevan dan akurat.

Software & Hardware

Software	Hardware
Laravel	Laptop
Maatwebsite	
Char.js	
PhP	
VsCode	
Platform io	

2.2 Implementation Steps (Langkah Implementasi)

- **Menyiapkan Laravel Project:**

Mulai dengan membuka folder laravel yang sudah dibuat pada Praktik 12 pada VSCode dan buka terminal. Jalankan perintah untuk menginstall paket yang diperlukan dan membuat controller baru:

bash

CopyEdit

```
composer require maatwebsite/excel
```

```
php artisan make:controller GraphController
```

- **Membuat GraphController:**

Setelah controller dibuat, buka dan tambahkan kode berikut pada GraphController untuk menangani pengambilan data dan rendering grafik:

php

CopyEdit

```
namespace App\Http\Controllers;
```

```
use App\Exports\TransaksiSensorExport;
```

```
use Maatwebsite\Excel\Facades\Excel;
```

```
use App\Models\TransaksiSensor;
```

```
class GraphController extends Controller
```

```
{
```

```
    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'));
```

```
    }
```

```
    public function exportToExcel()
```

```
    {
```

```
        return Excel::download(new TransaksiSensorExport, 'transaksi_sensor.xlsx');
```

```
    }
```

```
}
```

- **Membuat TransaksiSensorExport:**

Setelah membuat controller, jalankan perintah untuk membuat file ekspor dan tambahkan kode berikut pada TransaksiSensorExport:

php

CopyEdit

```
namespace App\Exports;
```

```
use App\Models\TransaksiSensor;
```

```
use Maatwebsite\Excel\Concerns\FromCollection;
```

```
class TransaksiSensorExport implements FromCollection
```

```
{
```

```
    public function collection()
```

```
    {
```

```
        return TransaksiSensor::all();
```

```
    }
```

```
}
```

- **Menambahkan Route pada web.php:**

Tambahkan route pada file web.php untuk menghubungkan halaman grafik dan fitur ekspor:

php

CopyEdit

```
use Illuminate\Support\Facades\Route;
```

```
use App\Http\Controllers\GraphController;
```

```
Route::get('/', [GraphController::class, 'index'])->name('graph');
```

```
Route::get('/graph/export', [GraphController::class, 'exportToExcel'])->name('graph.export');
```

□ **Membuat View graph.blade.php:**

Buat file graph.blade.php pada folder resources/views dan tambahkan kode berikut untuk menampilkan grafik dan data transaksi:

html

CopyEdit

```
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard Monitoring Sensor | Sistem IoT</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <div class="dashboard-container">
    <h1 class="header-title">Dashboard Monitoring Sensor</h1>
    <div class="card">
      <h2 class="card-title">Grafik Perbandingan Sensor</h2>
      <canvas id="sensorChart"></canvas>
      <button class="btn btn-outline" onclick="window.location.href='{ route('graph.export') }'">Export</button>
    </div>
  </div>

  <script>
    const labels = @json($labels);
    const dataNilai1 = @json($dataNilai1);
    const dataNilai2 = @json($dataNilai2);

    const ctx = document.getElementById('sensorChart').getContext('2d');
    new Chart(ctx, {
      type: 'line',
      data: {
        labels: labels,
        datasets: [
          {
            label: 'Sensor 1',
            data: dataNilai1,
            borderColor: '#4361ee',
            backgroundColor: 'rgba(67, 97, 238, 0.1)',
            borderWidth: 2,
            tension: 0.3,
            fill: true,
          },
          {
            label: 'Sensor 2',
            data: dataNilai2,
            borderColor: '#4cc9f0',
            backgroundColor: 'rgba(76, 201, 240, 0.1)',
            borderWidth: 2,
            tension: 0.3,
            fill: true,
          },
        ],
      },
      options: {
        responsive: true,
        interaction: { mode: 'index', intersect: false },
      },
    });
  </script>
</body>
</html>
```

- **Jalankan Aplikasi:**

Jalankan aplikasi menggunakan perintah php artisan serve dan akses dashboard di browser.

1. Results and Discussion (Hasil dan Pembahasan)

Setelah implementasi selesai, program ini berhasil menampilkan grafik dari data sensor yang disimpan dalam database. Data yang ditampilkan dalam bentuk grafik interaktif memungkinkan pengguna untuk melihat perbandingan nilai sensor secara visual. Fitur ekspor ke Excel juga memungkinkan pengguna untuk mendownload data transaksi dalam format yang mudah diolah lebih lanjut.

- **Grafik Perbandingan Sensor:** Grafik yang ditampilkan memberikan gambaran jelas mengenai nilai-nilai yang dihasilkan oleh dua sensor yang berbeda, memungkinkan analisis lebih mendalam terkait kinerja masing-masing sensor.
- **Export Data:** Pengguna dapat mengunduh data transaksi dalam format Excel, yang memudahkan dalam pengolahan data lebih lanjut, seperti analisis tren atau pembuatan laporan.

1.1 Experimental Results (Hasil Eksperimen)



