LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Membuat Tampilan Interface  
 Web Dashboard IoT**

*Abdillah Jibran - 233140700111046*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: abdillahjibran12@gmail.com*

**Abstract** (Abstrak)

Eksperimen ini bertujuan untuk membuat tampilan dashboard berbasis web yang menampilkan data transaksi sensor dalam bentuk grafik serta menyediakan fitur untuk mengunduh data tersebut dalam format Excel. Menggunakan Laravel sebagai framework backend, Maatwebsite Excel untuk ekspor data, dan Chart.js untuk visualisasi grafik. Dashboard ini akan membantu dalam memantau data sensor secara real-time melalui antarmuka yang mudah dipahami.

*Kata kunci: Dashboard IoT, Maatwebsite Excel, Monitoring Suhu dan Kelembaban.*

**1. Pendahuluan**

**1.1 Latar belakang**

Dalam implementasi Internet of Things (IoT), visualisasi data sangat penting untuk mempermudah pemantauan dan analisis. Dashboard berbasis web yang menampilkan data dari berbagai sensor dalam bentuk grafik memberikan gambaran yang jelas dan cepat. Dengan Laravel sebagai framework backend, kita dapat dengan mudah menangani data dan menampilkan grafik interaktif yang menarik bagi pengguna.

**1.2 Tujuan eksperimen**

1. Membuat antarmuka dashboard berbasis web untuk menampilkan data sensor.
2. Menggunakan Chart.js untuk menampilkan grafik dari data sensor.
3. Menggunakan Maatwebsite Excel untuk mengekspor data transaksi sensor ke dalam format Excel.
4. Menerapkan prinsip dasar pemrograman Laravel untuk menangani dan menampilkan data IoT secara efisien.

**2. Methodology (Metodologi)**

**2.1 Tools & Materials (Alat dan Bahan)**

Alat dan bahan yang di gunakan dalam membuat praktik ini adalah sebagai berikut:

* Laptop dengan Visual Studio Code, laravel, mysql
* Maatwebsite Excel untuk ekspor data ke Excel.
* XAMPP  
  1. **Implementation Steps (Langkah Implementasi)**

**2.1. Setup Backend Laravel**

* Instalasi Maatwebsite Excel: “composer require maatwebsite/excel”
* Membuat Controller: “php artisan make:controller GraphController” lalu tambahkan Method pada GraphController:

<?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**()

    {

*$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'));

    }

*/\*\**

*\* Mengunduh data transaksi sensor dalam format Excel*

*\**

*\* @return \Symfony\Component\HttpFoundation\BinaryFileResponse*

*\*/*

    public function **exportToExcel**()

    {

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

    }

}

* Membuat Export untuk Data Sensor: “php artisan make:exportTransaksiSensorExport --model=TransaksiSensor” lalu tambahkan code berikut:

<?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**();

    }

}

* Menambahkan Route di web.php:

<?php

*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');

**2.2. Membuat Tampilan Grafik pada graph.blade.php**

* Tulis program sebagai berikut:

<!DOCTYPE *html*>

<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://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

        <link *href*="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" *rel*="stylesheet">

    <script *src*="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <link   *rel*="stylesheet" *href*="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">

    <style>

*:root* {

*--primary-color*: #4361ee;

*--primary-light*: #e0e7ff;

*--secondary-color*: #3f37c9;

*--accent-color*: #4cc9f0;

*--accent-light*: #e0fbfc;

*--success-color*: #4bb543;

*--warning-color*: #f8961e;

*--danger-color*: #f94144;

*--light-color*: #f8f9fa;

*--dark-color*: #212529;

*--gray-color*: #6c757d;

        }

        \* {

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

            font-family: 'Poppins', sans-serif;

            background: linear-gradient(135*deg*, #f5f7fa 0*%*, #e2e8f0 100*%*);

            min-height: 100*vh*;

            padding: 2*rem* 1*rem*;

            color: var(*--dark-color*);

            line-height: 1.6;

        }

*.dashboard-container* {

            max-width: 1200*px*;

            margin: 0 auto;

        }

*.header* {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 2*rem*;

            flex-wrap: wrap;

            gap: 1*rem*;

        }

*.header-title* {

            font-size: 1.8*rem*;

            font-weight: 600;

            color: var(*--primary-color*);

            display: flex;

            align-items: center;

            gap: 0.75*rem*;

        }

*.header-title* i {

            color: var(*--accent-color*);

        }

*.card* {

            background-color: white;

            border-radius: 12*px*;

            box-shadow: 0 4*px* 20*px* rgba(0, 0, 0, 0.08);

            padding: 1.75*rem*;

            margin-bottom: 2*rem*;

            transition: transform 0.3*s* ease, box-shadow 0.3*s* ease;

        }

*.card:hover* {

            transform: translateY(-5*px*);

            box-shadow: 0 8*px* 30*px* rgba(0, 0, 0, 0.12);

        }

*.card-header* {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 1.5*rem*;

            padding-bottom: 1*rem*;

            border-bottom: 1*px* solid rgba(0, 0, 0, 0.05);

        }

*.card-title* {

            font-size: 1.25*rem*;

            font-weight: 600;

            color: var(*--primary-color*);

            display: flex;

            align-items: center;

            gap: 0.75*rem*;

        }

*.card-title* i {

            font-size: 1.1*em*;

        }

*.card-actions* {

            display: flex;

            gap: 0.75*rem*;

        }

*.btn* {

            padding: 0.5*rem* 1*rem*;

            border-radius: 8*px*;

            border: none;

            font-weight: 500;

            font-size: 0.9*rem*;

            cursor: pointer;

            transition: all 0.3*s* ease;

            display: inline-flex;

            align-items: center;

            gap: 0.5*rem*;

        }

*.btn-primary* {

            background-color: var(*--primary-color*);

            color: white;

        }

*.btn-primary:hover* {

            background-color: var(*--secondary-color*);

        }

*.btn-outline* {

            background-color: transparent;

            border: 1*px* solid var(*--primary-color*);

            color: var(*--primary-color*);

        }

*.btn-outline:hover* {

            background-color: var(*--primary-color*);

            color: white;

        }

*.btn-success* {

            background-color: var(*--success-color*);

            color: white;

        }

*.btn-success:hover* {

            opacity: 0.9;

        }

*.chart-container* {

            position: relative;

            height: 400*px*;

            width: 100*%*;

            margin-bottom: 1.5*rem*;

        }

*.data-summary* {

            display: grid;

            grid-template-columns: repeat(auto-fit, minmax(250*px*, 1*fr*));

            gap: 1.25*rem*;

            margin-top: 1.5*rem*;

        }

*.summary-card* {

            background-color: white;

            border-radius: 10*px*;

            padding: 1.25*rem*;

            box-shadow: 0 2*px* 10*px* rgba(0, 0, 0, 0.05);

            transition: transform 0.2*s* ease;

        }

*.summary-card:hover* {

            transform: translateY(-3*px*);

        }

*.summary-header* {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 0.75*rem*;

        }

*.summary-title* {

            font-size: 0.9*rem*;

            font-weight: 500;

            color: var(*--gray-color*);

        }

*.summary-icon* {

            width: 36*px*;

            height: 36*px*;

            border-radius: 8*px*;

            display: flex;

            align-items: center;

            justify-content: center;

            font-size: 1*rem*;

        }

*.sensor-1* {

            background-color: var(*--primary-light*);

            color: var(*--primary-color*);

        }

*.sensor-2* {

            background-color: var(*--accent-light*);

            color: var(*--accent-color*);

        }

*.summary-value* {

            font-size: 1.5*rem*;

            font-weight: 600;

            margin-bottom: 0.25*rem*;

        }

*.summary-change* {

            font-size: 0.85*rem*;

            display: flex;

            align-items: center;

            gap: 0.25*rem*;

        }

*.positive* {

            color: var(*--success-color*);

        }

*.negative* {

            color: var(*--danger-color*);

        }

*.neutral* {

            color: var(*--gray-color*);

        }

*.time-selector* {

            display: flex;

            justify-content: flex-end;

            gap: 0.5*rem*;

            margin-bottom: 1*rem*;

        }

*.time-btn* {

            padding: 0.35*rem* 0.75*rem*;

            border-radius: 6*px*;

            background-color: var(*--light-color*);

            border: none;

            font-size: 0.85*rem*;

            cursor: pointer;

            transition: all 0.2*s* ease;

        }

*.time-btn.active* {

            background-color: var(*--primary-color*);

            color: white;

        }

*.time-btn:hover:not*(*.active*) {

            background-color: #e9ecef;

        }

*@media* (max-width: 768*px*) {

*.header* {

                flex-direction: column;

                align-items: flex-start;

            }

*.chart-container* {

                height: 300*px*;

            }

*.data-summary* {

                grid-template-columns: 1*fr*;

            }

*.card-actions* {

                width: 100*%*;

                justify-content: space-between;

            }

        }

*.fade-in* {

            animation: fadeIn 0.6*s* ease-in-out;

        }

*@keyframes* *fadeIn* {

            from { opacity: 0; transform: translateY(10*px*); }

            to { opacity: 1; transform: translateY(0); }

        }

    </style>

</head>

<body>

    <div *class*="dashboard-container">

        <div *class*="header animate\_\_animated animate\_\_fadeIn">

            <h1 *class*="header-title">

                <i *class*="fas fa-chart-network"></i>

                Dashboard Monitoring Sensor

            </h1>

            <div *class*="time-selector">

                <button *class*="time-btn active">24 Jam</button>

                <button *class*="time-btn">7 Hari</button>

                <button *class*="time-btn">30 Hari</button>

                <button *class*="time-btn">Custom</button>

            </div>

        </div>

        <div *class*="card animate\_\_animated animate\_\_fadeIn animate\_\_delay-1s">

            <div *class*="card-header">

                <h2 *class*="card-title">

                    <i *class*="fas fa-wave-square"></i>

                    Grafik Perbandingan Sensor

                </h2>

                <div *class*="card-actions">

                    <button *class*="btn btn-outline" *onclick*="*window*.*location*.*href***=**'{{ route('*graph*.*export*') }}'">

                        <i *class*="fas fa-download"></i> Export

                    </button>

                </div>

            </div>

            <div *class*="chart-container">

                <canvas *id*="sensorChart"></canvas>

            </div>

            <div *class*="data-summary">

                <div *class*="summary-card fade-in">

                    <div *class*="summary-header">

                        <span *class*="summary-title">Sensor 1 (Rata-rata)</span>

                        <div *class*="summary-icon sensor-1">

                            <i *class*="fas fa-thermometer-half"></i>

                        </div>

                    </div>

                    <div *class*="summary-value" *id*="avg-sensor1">0</div>

                    <div *class*="summary-change positive">

                        <i *class*="fas fa-arrow-up"></i> <span *id*="change-sensor1">0%</span> dari periode sebelumnya

                    </div>

                </div>

                <div *class*="summary-card fade-in">

                    <div *class*="summary-header">

                        <span *class*="summary-title">Sensor 2 (Rata-rata)</span>

                        <div *class*="summary-icon sensor-2">

                            <i *class*="fas fa-thermometer-quarter"></i>

                        </div>

                    </div>

                    <div *class*="summary-value" *id*="avg-sensor2">0</div>

                    <div *class*="summary-change negative">

                        <i *class*="fas fa-arrow-down"></i> <span *id*="change-sensor2">0%</span> dari periode sebelumnya

                    </div>

                </div>

                <div *class*="summary-card fade-in">

                    <div *class*="summary-header">

                        <span *class*="summary-title">Korelasi</span>

                        <div *class*="summary-icon">

                            <i *class*="fas fa-link"></i>

                        </div>

                    </div>

                    <div *class*="summary-value" *id*="correlation-value">0.00</div>

                    <div *class*="summary-change neutral">

                        <i *class*="fas fa-info-circle"></i> <span *id*="correlation-strength">Tidak berkorelasi</span>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script>

        const *labels* **=** @**json**(*$labels*);

        const *dataNilai1* **=** @**json**(*$dataNilai1*);

        const *dataNilai2* **=** @**json**(*$dataNilai2*);

        function **calculateStats**(*data*) {

            const *sum* **=** *data*.**reduce**((*a*, *b*) => *a* **+** *b*, 0);

            const *avg* **=** *sum* **/** *data*.length;

            const *max* **=** *Math*.**max**(**...***data*);

            const *min* **=** *Math*.**min**(**...***data*);

*return* { *sum*, *avg*, *max*, *min* };

        }

        function **calculateCorrelation**(*x*, *y*) {

            const *n* **=** *x*.length;

            let *sumX* **=** 0, *sumY* **=** 0, *sumXY* **=** 0, *sumX2* **=** 0, *sumY2* **=** 0;

*for* (let *i* **=** 0; *i* **<** *n*; *i***++**) {

*sumX* **+=** *x*[*i*];

*sumY* **+=** *y*[*i*];

*sumXY* **+=** *x*[*i*] **\*** *y*[*i*];

*sumX2* **+=** *x*[*i*] **\*** *x*[*i*];

*sumY2* **+=** *y*[*i*] **\*** *y*[*i*];

            }

            const *numerator* **=** *sumXY* **-** (*sumX* **\*** *sumY*) **/** *n*;

            const *denominator* **=** *Math*.**sqrt**((*sumX2* **-** (*sumX* **\*** *sumX*) **/** *n*) **\*** (*sumY2* **-** (*sumY* **\*** *sumY*) **/** *n*));

*return* *denominator* **===** 0 **?** 0 **:** *numerator* **/** *denominator*;

        }

        const *stats1* **=** **calculateStats**(*dataNilai1*);

        const *stats2* **=** **calculateStats**(*dataNilai2*);

        const *correlation* **=** **calculateCorrelation**(*dataNilai1*, *dataNilai2*);

*document*.**getElementById**('avg-sensor1').*textContent* **=** *stats1*.*avg*.**toFixed**(2);

*document*.**getElementById**('avg-sensor2').*textContent* **=** *stats2*.*avg*.**toFixed**(2);

*document*.**getElementById**('change-sensor1').*textContent* **=** (*Math*.**random**() **\*** 5).**toFixed**(1) **+** '%';

*document*.**getElementById**('change-sensor2').*textContent* **=** (*Math*.**random**() **\*** 3).**toFixed**(1) **+** '%';

*document*.**getElementById**('correlation-value').*textContent* **=** *correlation*.**toFixed**(2);

        const *correlationStrength* **=** *document*.**getElementById**('correlation-strength');

*if* (*Math*.**abs**(*correlation*) **>** 0.7) {

*correlationStrength*.*textContent* **=** 'Korelasi kuat';

*correlationStrength*.*className* **=** 'positive';

        } *else* *if* (*Math*.**abs**(*correlation*) **>** 0.3) {

*correlationStrength*.*textContent* **=** 'Korelasi sedang';

*correlationStrength*.*className* **=** 'neutral';

        } *else* {

*correlationStrength*.*textContent* **=** 'Korelasi lemah';

*correlationStrength*.*className* **=** 'negative';

        }

        const *ctx* **=** *document*.**getElementById**('sensorChart').**getContext**('2d');

        const *chart* **=** **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,

                        pointBackgroundColor: 'white',

                        pointBorderColor: '#4361ee',

                        pointBorderWidth: 2,

                        pointRadius: 4,

                        pointHoverRadius: 6,

                        yAxisID: 'y'

                    },

                    {

                        label: 'Sensor 2',

                        data: *dataNilai2*,

                        borderColor: '#4cc9f0',

                        backgroundColor: 'rgba(76, 201, 240, 0.1)',

                        borderWidth: 2,

                        tension: 0.3,

                        fill: true,

                        pointBackgroundColor: 'white',

                        pointBorderColor: '#4cc9f0',

                        pointBorderWidth: 2,

                        pointRadius: 4,

                        pointHoverRadius: 6,

                        yAxisID: 'y'

                    }

                ]

            },

            options: {

                responsive: true,

                maintainAspectRatio: false,

                interaction: {

                    mode: 'index',

                    intersect: false

                },

                plugins: {

                    legend: {

                        position: 'top',

                        labels: {

                            usePointStyle: true,

                            padding: 20,

                            font: {

                                size: 13,

                                weight: '500'

                            }

                        }

                    },

                    tooltip: {

                        backgroundColor: 'rgba(0, 0, 0, 0.85)',

                        titleFont: {

                            size: 14,

                            weight: '600'

                        },

                        bodyFont: {

                            size: 13

                        },

                        padding: 12,

                        cornerRadius: 8,

                        usePointStyle: true,

                        callbacks: {

**label**: function(*context*) {

                                let *label* **=** *context*.*dataset*.*label* **||** '';

*if* (*label*) {

*label* **+=** ': ';

                                }

*if* (*context*.*parsed*.*y* **!==** null) {

*label* **+=** *context*.*parsed*.*y*.**toFixed**(2);

                                }

*return* *label*;

                            }

                        }

                    },

                    annotation: {

                        annotations: {

                            line1: {

                                type: 'line',

                                yMin: *stats1*.*avg*,

                                yMax: *stats1*.*avg*,

                                borderColor: '#4361ee',

                                borderWidth: 1,

                                borderDash: [5, 5],

                                label: {

                                    content: 'Rata-rata S1: ' **+** *stats1*.*avg*.**toFixed**(2),

                                    enabled: true,

                                    position: 'right',

                                    backgroundColor: 'rgba(67, 97, 238, 0.7)'

                                }

                            },

                            line2: {

                                type: 'line',

                                yMin: *stats2*.*avg*,

                                yMax: *stats2*.*avg*,

                                borderColor: '#4cc9f0',

                                borderWidth: 1,

                                borderDash: [5, 5],

                                label: {

                                    content: 'Rata-rata S2: ' **+** *stats2*.*avg*.**toFixed**(2),

                                    enabled: true,

                                    position: 'right',

                                    backgroundColor: 'rgba(76, 201, 240, 0.7)'

                                }

                            }

                        }

                    }

                },

                scales: {

                    y: {

                        beginAtZero: false,

                        grid: {

                            color: 'rgba(0, 0, 0, 0.05)'

                        },

                        ticks: {

                            font: {

                                size: 12

                            }

                        }

                    },

                    x: {

                        grid: {

                            display: false

                        },

                        ticks: {

                            font: {

                                size: 12

                            }

                        }

                    }

                },

                animation: {

                    duration: 1000,

                    easing: 'easeOutQuart'

                }

            }

        });

*// Time selector functionality*

*document*.**querySelectorAll**('.time-btn').**forEach**(*btn* => {

*btn*.**addEventListener**('click', function() {

*document*.**querySelectorAll**('.time-btn').**forEach**(*b* => *b*.*classList*.**remove**('active'));

*this*.*classList*.**add**('active');

*chart*.*data*.*datasets*.**forEach**(*dataset* => {

*dataset*.*data* **=** *dataset*.*data*.**map**(() => *Math*.**random**() **\*** 100);

                });

*chart*.**update**();

            });

        });

*window*.**addEventListener**('resize', function() {

*chart*.**resize**();

        });

    </script>

</body>

</html>

**3. Hasil Praktik dan lampiran**

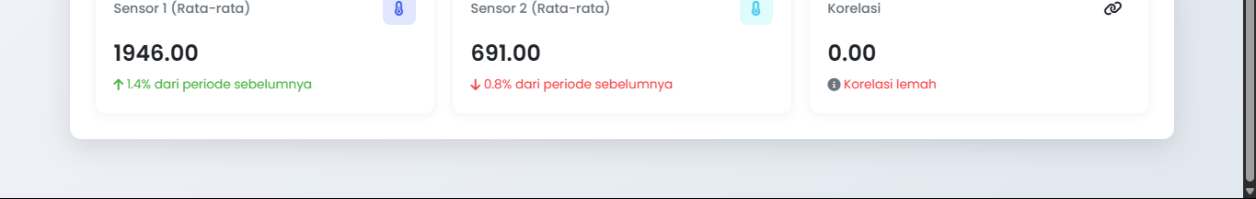
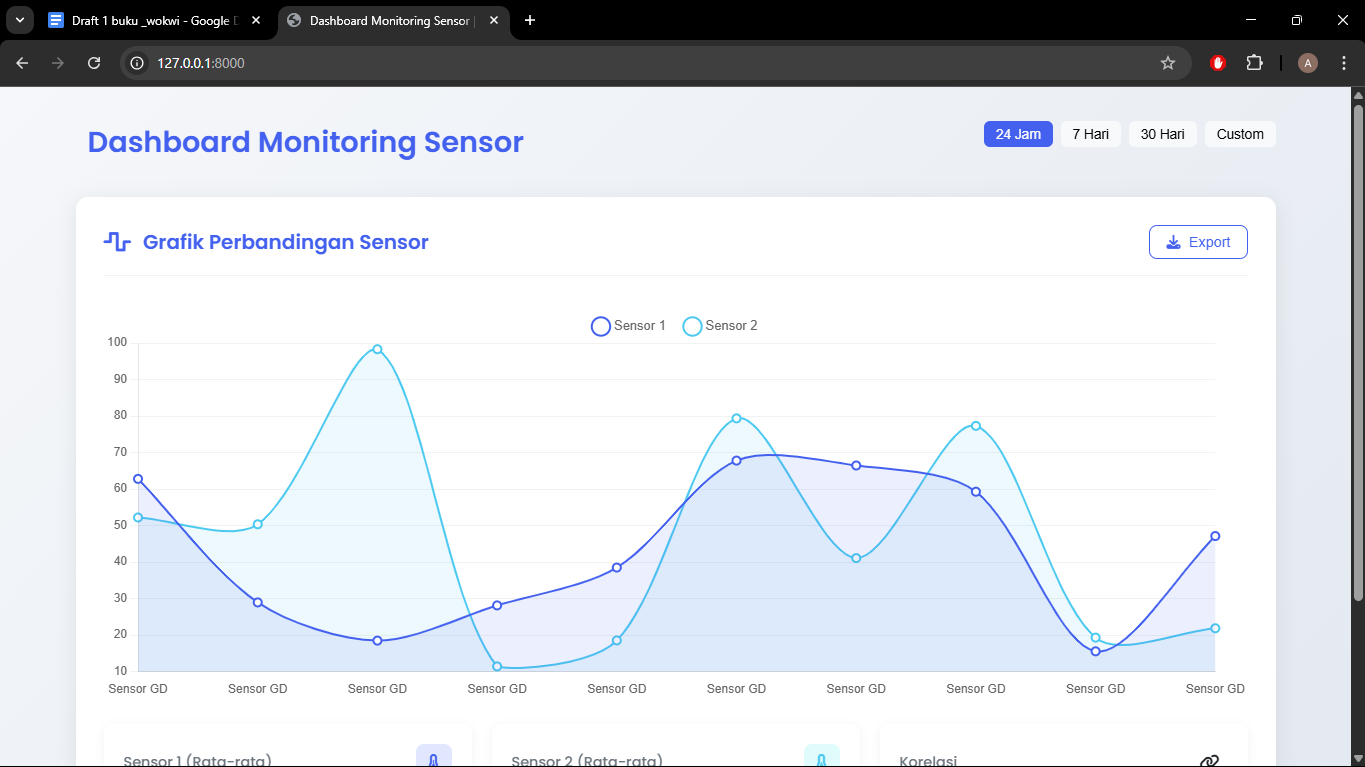
**3.1 Hasil Eksperimen**

#### Grafik berhasil ditampilkan menggunakan Chart.js, menampilkan dua jenis data (kelembaban dan suhu) yang diambil dari tabel TransaksiSensor.

#### Pengguna dapat mengunduh data transaksi sensor dalam format Excel menggunakan fitur yang telah disediakan.

#### Data yang ditampilkan bersumber dari database MySQL dan dapat diperbarui secara real-time.

**3.2 Lampiran**

****

