

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

FAKULTAS VOKASI, UNIVERSITAS BRAWIJAYA

INTERFACE WEB DASHBOARD IOT

Restu Nur Kharisma

Fakultas Vokasi, Universitas Brawijaya

Email: restukharisma@student.ub.ac.id

ABSTRACT

The development of Internet of Things (IoT) technology has enabled the collection and real-time monitoring of data from various sensor devices. One of the widely used IoT applications is environmental temperature monitoring using temperature sensors. This research aims to design and implement a web-based dashboard for real-time temperature data monitoring from IoT sensors. The system is developed using the Laravel framework for the backend and integrates with temperature sensors connected via the HTTP protocol. The dashboard allows users to monitor temperature data, access temperature trend graphs, and receive notifications when the temperature exceeds certain thresholds. Furthermore, the system is designed with a responsive user interface, making it accessible on both desktop and mobile devices. The result of this research is an efficient online temperature monitoring system that can be used for various purposes such as environmental surveillance, facility management, and other applications that require real-time temperature monitoring.

Keywords — Internet of Things (IoT), temperature sensor, web dashboard, Laravel framework, real-time monitoring

1. Pendahuluan

1.1. Latar Belakang

Perkembangan teknologi Internet of Things (IoT) telah membawa perubahan signifikan dalam berbagai bidang, termasuk pemantauan kondisi lingkungan secara real-time. Salah satu aplikasi IoT yang banyak digunakan adalah pemantauan suhu, yang sangat penting dalam berbagai sektor, seperti industri manufaktur, pertanian, pengelolaan gedung, dan lainnya. Dengan menggunakan sensor suhu yang terhubung dengan internet, data suhu dapat dikumpulkan dan dipantau secara langsung, memberikan kemudahan dalam pengawasan dan pengambilan keputusan yang cepat.

Namun, meskipun teknologi sensor suhu sudah cukup berkembang, tantangan yang dihadapi adalah bagaimana mengelola dan memvisualisasikan data suhu tersebut dengan cara yang efisien dan mudah dipahami. Untuk itu, dibutuhkan sebuah sistem yang dapat mengintegrasikan sensor suhu IoT dengan antarmuka pengguna yang intuitif dan responsif, sehingga memungkinkan pengguna untuk memantau data suhu secara real-time dari jarak jauh.

Salah satu solusi yang dapat digunakan adalah dengan membangun sebuah dashboard berbasis web yang memanfaatkan framework Laravel. Laravel adalah framework PHP yang sangat populer dan banyak digunakan untuk pengembangan aplikasi web, karena kemudahan dalam pengelolaan database, keamanan, dan kemampuan untuk menangani berbagai permintaan HTTP. Dengan memanfaatkan Laravel, data dari sensor suhu dapat ditampilkan dalam bentuk grafik, notifikasi, serta informasi penting lainnya yang dapat diakses secara online.

Melihat pentingnya pemantauan suhu dalam berbagai sektor, penelitian ini bertujuan untuk merancang dan mengembangkan sebuah dashboard web berbasis Laravel yang dapat memantau dan menampilkan data suhu IoT secara real-time. Dengan sistem ini, diharapkan dapat memberikan solusi yang efisien dalam pengawasan suhu secara online dan memberikan kemudahan bagi pengguna dalam mengakses informasi suhu secara praktis dan akurat.

1.2. Tujuan Eksperimen

- 1) Mendesain dan mengimplementasikan sistem dashboard berbasis web yang mampu menampilkan data suhu secara real-time yang diterima dari sensor suhu IoT.
- 2) Mengintegrasikan sensor suhu dengan sistem backend berbasis Laravel untuk memperoleh data suhu yang akurat dan dapat diproses untuk ditampilkan kepada pengguna dalam format yang mudah dipahami.
- 3) Membangun antarmuka pengguna yang responsif dan interaktif, sehingga pengguna dapat memantau data suhu melalui perangkat desktop maupun mobile dengan tampilan yang user-friendly.
- 4) Menyediakan fitur notifikasi untuk memberikan informasi kepada pengguna apabila suhu mencapai batas tertentu, sehingga tindakan preventif dapat diambil dengan cepat.
- 5) Menguji kinerja sistem dalam memproses dan menampilkan data suhu secara efisien, serta menilai kehandalan dan responsivitas sistem dalam berbagai kondisi penggunaan.

2. Metodologi

2.1. Alat dan Bahan

- 1) Alat: sensor suhu DHT22, ESP32, web server apache, laptop
- 2) Bahan: framework Laravel, MySQL Database, HTML, JavaScript, Wi-Fi, Visual Studio Code

2.2. Langkah Implementasi

- 1) Buka folder Laravel yang sudah dibuat sebelumnya pada VSC, dan buka terminal dengan menjalankan perintah **"composer require maatwebsite/excel"**
- 2) Lalu buat controller untuk graph dengan menggunakan command **"php artisan make:controller GraphController"**.
- 3) Ubah kode pada GraphController menjadi seperti ini.

```
3. <?php
4.
5. namespace App\Http\Controllers;
6.
7. use App\Exports\TransaksiSensorExport;
8. use Maatwebsite\Excel\Facades\Excel;
```

```

9. use App\Models\TransaksiSensor;
10.
11. class GraphController extends Controller
12. {
13.     /**
14.      * Menampilkan grafik transaksi sensor.
15.      *
16.      * @return \Illuminate\View\View
17.      */
18.     public function index()
19.     {
20.         // Mengambil data transaksi sensor
21.         $transaksiSensors = TransaksiSensor::latest()->take(10)->get();
22.
23.         // Mengambil data label
24.         $labels = $transaksiSensors->pluck('nama_sensor');
25.
26.         // Mengambil data nilai1 dan nilai2 untuk grafik
27.         $dataNilai1 = $transaksiSensors->pluck('nilai1');
28.         $dataNilai2 = $transaksiSensors->pluck('nilai2');
29.
30.         return view('graph', compact('labels', 'dataNilai1', 'dataNilai2'));
31.     }
32.
33.     /**
34.      * Mengunduh data transaksi sensor dalam format Excel
35.      *
36.      * @return \Symfony\Component\HttpFoundation\BinaryFileResponse
37.      */
38.     public function exportToExcel()
39.     {
40.         return Excel::download(new TransaksiSensorExport, 'transaksi_sensor.xlsx');
41.     }
42. }
43.

```

- 4) Setelah itu buat model untuk export dengan menjalankan command pada terminal dengan command **“php artisan make:export TransaksiSensorExport --model=TransaksiSensor”**
- 5) Lalu ubah kode pada TransaksiSensorExport dengan kode 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();
    }
}

```

- 6) Lalu buat file dengan nama **graph.blade.php** pada folder **resources/views** dan tambahkan kode berikut.

```

3. <!DOCTYPE html>
4. <html lang="id">
5.
6. <head>
7.     <meta charset="UTF-8">
8.     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9.     <title>Dashboard Monitoring Sensor | Sistem IoT</title>
10.
11.     <link rel="stylesheet"
12.         href="https://cdnjs.cloudflare.com/ajax/libs/font-
13.         awesome/6.4.0/css/all.min.css">
14.
15.     <link
16.         href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;
17.         700&display=swap" rel="stylesheet">
18.
19.     <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
20.
21.     <link rel="stylesheet"
22.         href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.c
23.         ss">
24.
25.     <style>
26.         :root {
27.             --primary-color: #4361ee;
28.             --primary-light: #e0e7ff;
29.             --secondary-color: #3f37c9;
30.             --accent-color: #4cc9f0;

```

```
25.         --accent-light: #e0fbfc;
26.         --success-color: #4bb543;
27.         --warning-color: #f8961e;
28.         --danger-color: #f94144;
29.         --light-color: #f8f9fa;
30.         --dark-color: #212529;
31.         --gray-color: #6c757d;
32.     }
33.
34.     * {
35.         margin: 0;
36.         padding: 0;
37.         box-sizing: border-box;
38.     }
39.
40.     body {
41.         font-family: 'Poppins', sans-serif;
42.         background: linear-gradient(135deg, #f5f7fa 0%, #e2e8f0
100%);
43.         min-height: 100vh;
44.         padding: 2rem 1rem;
45.         color: var(--dark-color);
46.         line-height: 1.6;
47.     }
48.
49.     .dashboard-container {
50.         max-width: 1200px;
51.         margin: 0 auto;
52.     }
53.
54.     .header {
55.         display: flex;
56.         justify-content: space-between;
57.         align-items: center;
58.         margin-bottom: 2rem;
59.         flex-wrap: wrap;
60.         gap: 1rem;
61.     }
62.
63.     .header-title {
64.         font-size: 1.8rem;
65.         font-weight: 600;
66.         color: var(--primary-color);
67.         display: flex;
68.         align-items: center;
69.         gap: 0.75rem;
70.     }
71.
72.     .header-title i {
73.         color: var(--accent-color);
74.     }
```

```
75.
76.     .card {
77.         background-color: white;
78.         border-radius: 12px;
79.         box-shadow: 0 4px 20px rgba(0, 0, 0, 0.08);
80.         padding: 1.75rem;
81.         margin-bottom: 2rem;
82.         transition: transform 0.3s ease, box-shadow 0.3s ease;
83.     }
84.
85.     .card:hover {
86.         transform: translateY(-5px);
87.         box-shadow: 0 8px 30px rgba(0, 0, 0, 0.12);
88.     }
89.
90.     .card-header {
91.         display: flex;
92.         justify-content: space-between;
93.         align-items: center;
94.         margin-bottom: 1.5rem;
95.         padding-bottom: 1rem;
96.         border-bottom: 1px solid rgba(0, 0, 0, 0.05);
97.     }
98.
99.     .card-title {
100.         font-size: 1.25rem;
101.         font-weight: 600;
102.         color: var(--primary-color);
103.         display: flex;
104.         align-items: center;
105.         gap: 0.75rem;
106.     }
107.
108.     .card-title i {
109.         font-size: 1.1em;
110.     }
111.
112.     .card-actions {
113.         display: flex;
114.         gap: 0.75rem;
115.     }
116.
117.     .btn {
118.         padding: 0.5rem 1rem;
119.         border-radius: 8px;
120.         border: none;
121.         font-weight: 500;
122.         font-size: 0.9rem;
123.         cursor: pointer;
124.         transition: all 0.3s ease;
125.         display: inline-flex;
```

```
126.         align-items: center;
127.         gap: 0.5rem;
128.     }
129.
130.     .btn-primary {
131.         background-color: var(--primary-color);
132.         color: white;
133.     }
134.
135.     .btn-primary:hover {
136.         background-color: var(--secondary-color);
137.     }
138.
139.     .btn-outline {
140.         background-color: transparent;
141.         border: 1px solid var(--primary-color);
142.         color: var(--primary-color);
143.     }
144.
145.     .btn-outline:hover {
146.         background-color: var(--primary-color);
147.         color: white;
148.     }
149.
150.     .btn-success {
151.         background-color: var(--success-color);
152.         color: white;
153.     }
154.
155.     .btn-success:hover {
156.         opacity: 0.9;
157.     }
158.
159.     .chart-container {
160.         position: relative;
161.         height: 400px;
162.         width: 100%;
163.         margin-bottom: 1.5rem;
164.     }
165.
166.     .data-summary {
167.         display: grid;
168.         grid-template-columns: repeat(auto-fit, minmax(250px,
169. 1fr));
170.         gap: 1.25rem;
171.         margin-top: 1.5rem;
172.     }
173.
174.     .summary-card {
175.         background-color: white;
176.         border-radius: 10px;
```

```
176.         padding: 1.25rem;
177.         box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);
178.         transition: transform 0.2s ease;
179.     }
180.
181.     .summary-card:hover {
182.         transform: translateY(-3px);
183.     }
184.
185.     .summary-header {
186.         display: flex;
187.         justify-content: space-between;
188.         align-items: center;
189.         margin-bottom: 0.75rem;
190.     }
191.
192.     .summary-title {
193.         font-size: 0.9rem;
194.         font-weight: 500;
195.         color: var(--gray-color);
196.     }
197.
198.     .summary-icon {
199.         width: 36px;
200.         height: 36px;
201.         border-radius: 8px;
202.         display: flex;
203.         align-items: center;
204.         justify-content: center;
205.         font-size: 1rem;
206.     }
207.
208.     .sensor-1 {
209.         background-color: var(--primary-light);
210.         color: var(--primary-color);
211.     }
212.
213.     .sensor-2 {
214.         background-color: var(--accent-light);
215.         color: var(--accent-color);
216.     }
217.
218.     .summary-value {
219.         font-size: 1.5rem;
220.         font-weight: 600;
221.         margin-bottom: 0.25rem;
222.     }
223.
224.     .summary-change {
225.         font-size: 0.85rem;
226.         display: flex;
```



```
227.         align-items: center;
228.         gap: 0.25rem;
229.     }
230.
231.     .positive {
232.         color: var(--success-color);
233.     }
234.
235.     .negative {
236.         color: var(--danger-color);
237.     }
238.
239.     .neutral {
240.         color: var(--gray-color);
241.     }
242.
243.     .time-selector {
244.         display: flex;
245.         justify-content: flex-end;
246.         gap: 0.5rem;
247.         margin-bottom: 1rem;
248.     }
249.
250.     .time-btn {
251.         padding: 0.35rem 0.75rem;
252.         border-radius: 6px;
253.         background-color: var(--light-color);
254.         border: none;
255.         font-size: 0.85rem;
256.         cursor: pointer;
257.         transition: all 0.2s ease;
258.     }
259.
260.     .time-btn.active {
261.         background-color: var(--primary-color);
262.         color: white;
263.     }
264.
265.     .time-btn:hover:not(.active) {
266.         background-color: #e9ecef;
267.     }
268.
269.     @media (max-width: 768px) {
270.         .header {
271.             flex-direction: column;
272.             align-items: flex-start;
273.         }
274.
275.         .chart-container {
276.             height: 300px;
277.         }

```

```

278.
279.         .data-summary {
280.             grid-template-columns: 1fr;
281.         }
282.
283.         .card-actions {
284.             width: 100%;
285.             justify-content: space-between;
286.         }
287.     }
288.
289.     .fade-in {
290.         animation: fadeIn 0.6s ease-in-out;
291.     }
292.
293.     @keyframes fadeIn {
294.         from {
295.             opacity: 0;
296.             transform: translateY(10px);
297.         }
298.
299.         to {
300.             opacity: 1;
301.             transform: translateY(0);
302.         }
303.     }
304. </style>
305. </head>
306.
307. <body>
308.     <div class="dashboard-container">
309.         <div class="header animate__animated animate__fadeIn">
310.             <h1 class="header-title">
311.                 <i class="fas fa-chart-network"></i>
312.                 Dashboard Monitoring Sensor
313.             </h1>
314.             <div class="time-selector">
315.                 <button class="time-btn active">24 Jam</button>
316.                 <button class="time-btn">7 Hari</button>
317.                 <button class="time-btn">30 Hari</button>
318.                 <button class="time-btn">Custom</button>
319.             </div>
320.         </div>
321.
322.         <div class="card animate__animated animate__fadeIn
323.             animate__delay-1s">
324.             <div class="card-header">
325.                 <h2 class="card-title">
326.                     <i class="fas fa-wave-square"></i>
327.                     Grafik Perbandingan Sensor

```

```

328.         <div class="card-actions">
329.             <a href="{ route('graph.export') }" class="btn
      btn-outline">
330.                 <i class="fas fa-download"></i> Export
331.             </a>
332.
333.         </div>
334.     </div>
335.
336.     <div class="chart-container">
337.         <canvas id="sensorChart"></canvas>
338.     </div>
339.
340.     <div class="data-summary">
341.         <div class="summary-card fade-in">
342.             <div class="summary-header">
343.                 <span class="summary-title">Sensor 1 (Rata-
      rata)</span>
344.                 <div class="summary-icon sensor-1">
345.                     <i class="fas fa-thermometer-half"></i>
346.                 </div>
347.             </div>
348.             <div class="summary-value" id="avg-
      sensor1">0</div>
349.             <div class="summary-change positive">
350.                 <i class="fas fa-arrow-up"></i> <span
      id="change-sensor1">0%</span> dari periode sebelumnya
351.             </div>
352.         </div>
353.
354.         <div class="summary-card fade-in">
355.             <div class="summary-header">
356.                 <span class="summary-title">Sensor 2 (Rata-
      rata)</span>
357.                 <div class="summary-icon sensor-2">
358.                     <i class="fas fa-thermometer-quarter"></i>
359.                 </div>
360.             </div>
361.             <div class="summary-value" id="avg-
      sensor2">0</div>
362.             <div class="summary-change negative">
363.                 <i class="fas fa-arrow-down"></i> <span
      id="change-sensor2">0%</span> dari periode sebelumnya
364.             </div>
365.         </div>
366.
367.         <div class="summary-card fade-in">
368.             <div class="summary-header">
369.                 <span class="summary-title">Korelasi</span>
370.             <div class="summary-icon">
371.                 <i class="fas fa-link"></i>

```

```

372.         </div>
373.     </div>
374.     <div class="summary-value" id="correlation-
value">0.00</div>
375.         <div class="summary-change neutral">
376.             <i class="fas fa-info-circle"></i> <span
id="correlation-strength">Tidak berkorelasi</span>
377.         </div>
378.     </div>
379. </div>
380. </div>
381. </div>
382.
383. <div id="chartData"
384.     data-labels="{ { json_encode($labels) } }"
385.     data-nilai1="{ { json_encode($dataNilai1) } }"
386.     data-nilai2="{ { json_encode($dataNilai2) } }">
387. </div>
388.
389. <script>
390.     const chartData = document.getElementById('chartData');
391.
392.     const labels = JSON.parse(chartData.dataset.labels);
393.     const dataNilai1 = JSON.parse(chartData.dataset.nilai1);
394.     const dataNilai2 = JSON.parse(chartData.dataset.nilai2);
395.
396.     function calculateStats(data) {
397.         const sum = data.reduce((a, b) => a + b, 0);
398.         const avg = sum / data.length;
399.         const max = Math.max(...data);
400.         const min = Math.min(...data);
401.         return {
402.             sum,
403.             avg,
404.             max,
405.             min
406.         };
407.     }
408.
409.     function calculateCorrelation(x, y) {
410.         const n = x.length;
411.         let sumX = 0,
412.             sumY = 0,
413.             sumXY = 0,
414.             sumX2 = 0,
415.             sumY2 = 0;
416.
417.         for (let i = 0; i < n; i++) {
418.             sumX += x[i];
419.             sumY += y[i];
420.             sumXY += x[i] * y[i];

```

```

421.             sumX2 += x[i] * x[i];
422.             sumY2 += y[i] * y[i];
423.         }
424.
425.         const numerator = sumXY - (sumX * sumY) / n;
426.         const denominator = Math.sqrt((sumX2 - (sumX * sumX) / n)
    * (sumY2 - (sumY * sumY) / n));
427.
428.         return denominator === 0 ? 0 : numerator / denominator;
429.     }
430.
431.     const stats1 = calculateStats(dataNilai1);
432.     const stats2 = calculateStats(dataNilai2);
433.     const correlation = calculateCorrelation(dataNilai1,
    dataNilai2);
434.
435.     document.getElementById('avg-sensor1').textContent =
    stats1.avg.toFixed(2);
436.     document.getElementById('avg-sensor2').textContent =
    stats2.avg.toFixed(2);
437.
438.     document.getElementById('change-sensor1').textContent =
    (Math.random() * 5).toFixed(1) + '%';
439.     document.getElementById('change-sensor2').textContent =
    (Math.random() * 3).toFixed(1) + '%';
440.
441.     document.getElementById('correlation-value').textContent =
    correlation.toFixed(2);
442.
443.     const correlationStrength =
    document.getElementById('correlation-strength');
444.     if (Math.abs(correlation) > 0.7) {
445.         correlationStrength.textContent = 'Korelasi kuat';
446.         correlationStrength.className = 'positive';
447.     } else if (Math.abs(correlation) > 0.3) {
448.         correlationStrength.textContent = 'Korelasi sedang';
449.         correlationStrength.className = 'neutral';
450.     } else {
451.         correlationStrength.textContent = 'Korelasi lemah';
452.         correlationStrength.className = 'negative';
453.     }
454.
455.     const ctx =
    document.getElementById('sensorChart').getContext('2d');
456.     const chart = new Chart(ctx, {
457.         type: 'line',
458.         data: {
459.             labels: labels,
460.             datasets: [{
461.                 label: 'Sensor 1',
462.                 data: dataNilai1,

```

```

463.         borderColor: '#4361ee',
464.         backgroundColor: 'rgba(67, 97, 238, 0.1)',
465.         borderWidth: 2,
466.         tension: 0.3,
467.         fill: true,
468.         pointBackgroundColor: 'white',
469.         pointBorderColor: '#4361ee',
470.         pointBorderWidth: 2,
471.         pointRadius: 4,
472.         pointHoverRadius: 6,
473.         yAxisID: 'y'
474.     },
475.     {
476.         label: 'Sensor 2',
477.         data: dataNilai2,
478.         borderColor: '#4cc9f0',
479.         backgroundColor: 'rgba(76, 201, 240, 0.1)',
480.         borderWidth: 2,
481.         tension: 0.3,
482.         fill: true,
483.         pointBackgroundColor: 'white',
484.         pointBorderColor: '#4cc9f0',
485.         pointBorderWidth: 2,
486.         pointRadius: 4,
487.         pointHoverRadius: 6,
488.         yAxisID: 'y'
489.     }
490. ]
491. },
492. options: {
493.     responsive: true,
494.     maintainAspectRatio: false,
495.     interaction: {
496.         mode: 'index',
497.         intersect: false
498.     },
499.     plugins: {
500.         legend: {
501.             position: 'top',
502.             labels: {
503.                 usePointStyle: true,
504.                 padding: 20,
505.                 font: {
506.                     size: 13,
507.                     weight: '500'
508.                 }
509.             }
510.         },
511.         tooltip: {
512.             backgroundColor: 'rgba(0, 0, 0, 0.85)',
513.             titleFont: {

```

```

514.                size: 14,
515.                weight: '600'
516.            },
517.            bodyFont: {
518.                size: 13
519.            },
520.            padding: 12,
521.            cornerRadius: 8,
522.            usePointStyle: true,
523.            callbacks: {
524.                label: function(context) {
525.                    let label = context.dataset.label ||
';
526.                    if (label) {
527.                        label += ': ';
528.                    }
529.                    if (context.parsed.y !== null) {
530.                        label +=
context.parsed.y.toFixed(2);
531.                    }
532.                    return label;
533.                }
534.            }
535.        },
536.        annotation: {
537.            annotations: {
538.                line1: {
539.                    type: 'line',
540.                    yMin: stats1.avg,
541.                    yMax: stats1.avg,
542.                    borderColor: '#4361ee',
543.                    borderWidth: 1,
544.                    borderDash: [5, 5],
545.                    label: {
546.                        content: 'Rata-rata S1: ' +
stats1.avg.toFixed(2),
547.                        enabled: true,
548.                        position: 'right',
549.                        backgroundColor: 'rgba(67, 97,
238, 0.7)'
550.                    }
551.                },
552.                line2: {
553.                    type: 'line',
554.                    yMin: stats2.avg,
555.                    yMax: stats2.avg,
556.                    borderColor: '#4cc9f0',
557.                    borderWidth: 1,
558.                    borderDash: [5, 5],
559.                    label: {

```

```

560.             content: 'Rata-rata S2: ' +
    stats2.avg.toFixed(2),
561.             enabled: true,
562.             position: 'right',
563.             backgroundColor: 'rgba(76, 201,
    240, 0.7)'
564.         }
565.     }
566. }
567. },
568. scales: {
569.     y: {
570.         beginAtZero: false,
571.         grid: {
572.             color: 'rgba(0, 0, 0, 0.05)'
573.         },
574.         ticks: {
575.             font: {
576.                 size: 12
577.             }
578.         }
579.     },
580.     x: {
581.         grid: {
582.             display: false
583.         },
584.         ticks: {
585.             font: {
586.                 size: 12
587.             }
588.         }
589.     }
590. },
591. animation: {
592.     duration: 1000,
593.     easing: 'easeOutQuart'
594. }
595. }
596. });
597.
598.
599. // Time selector functionality
600. document.querySelectorAll('.time-btn').forEach(btn => {
601.     btn.addEventListener('click', function() {
602.         document.querySelectorAll('.time-btn').forEach(b =>
        b.classList.remove('active'));
603.         this.classList.add('active');
604.
605.         chart.data.datasets.forEach(dataset => {
606.             dataset.data = dataset.data.map(() =>
        Math.random() * 100);

```



```
607.         });
608.         chart.update();
609.     });
610. });
611.
612.     window.addEventListener('resize', function() {
613.         chart.resize();
614.     });
615. </script>
616. </body>
617.
618. </html>
619.
```

- 6) Setelah semua sudah, jalankan program dengan perintah **“php artisan serve”** pada terminal

620. Hasil dan Pembahasan

3.1. Hasil Eksperimen

