**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN MENGENAI PROBLEM MASALAH**

**KONSISTENSI DATA DALAM SKENARIO APLIKASI REAL-TIME**

*Laporan ini disusun untuk memenuhi tugas 1 mata kuliah Pengembangan Web (Teori)*



Disusun oleh kelompok B4:

Asri Husnul Rosadi 221524035

Faris Abulkhoir 221524040

Mahardika Pratama 221524044

Muhamad Fahri Yuwan 221524047

Najib Alimudin Fajri 221524053

Septyana Agustina 221524058

Sarah 221524059

Dosen Pengampu:

Joe Lian Min, M.Eng.

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# **DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc177765518)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc177765519)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc177765520)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc177765521)

[**D.** **PELAKSANAAN EKSPERIMEN** 2](#_Toc177765522)

[**E.** **ANALISIS HASIL EKSPERIMEN** 2](#_Toc177765523)

[**F.** **KESIMPULAN** 5](#_Toc177765524)

Link chatGPT : https://chatgpt.com/share/66ed9306-c34c-8001-af26-0e4e93a2d3d6

# **IDENTIFIKASI PROBLEM**

Dalam aplikasi real-time seperti dashboard monitoring, konsistensi data menjadi krusial untuk memberikan informasi yang akurat dan tepat waktu. Terdapat berbagai metode untuk mengirim data secara real-time, seperti WebSockets dan Server-Sent Events (SSE). Namun, bagaimana kedua metode ini dibandingkan dalam hal konsistensi dan latency saat terjadi update cepat?

# **DESKRIPSI PROBLEM**

Masalah yang dihadapi adalah memahami perbedaan dalam konsistensi data yang dikirim melalui WebSockets dan SSE. Terutama dalam skenario di mana data diperbarui secara cepat, penting untuk mengetahui seberapa cepat dan konsisten masing-masing protokol dapat mengirimkan data. Latency menjadi faktor utama dalam mengukur seberapa cepat pesan dapat diterima oleh klien setelah dikirim oleh server.

# **METODOLOGI EKSPERIMEN**

Eksperimen dilakukan dengan membandingkan dua protokol pengiriman data: WebSockets dan SSE. Kode server dan klien diimplementasikan untuk mengirim dan menerima waktu secara berkala (setiap 5000 ms). Latency diukur dengan mencatat waktu yang dikirim dan waktu yang diterima. Data dikumpulkan untuk dianalisis dan dibandingkan.

# **PELAKSANAAN EKSPERIMEN**

|  |
| --- |
| 1. Setup Server dan Klien: Dua server diatur, satu untuk WebSocket dan satu untuk SSE, dengan endpoint yang mengirimkan timestamp secara berkala. Klien kemudian dibuat untuk menerima data dari masing-masing server. 2. Pengukuran Latency: Setiap kali pesan diterima, waktu penerimaan dicatat. Latency dihitung dengan selisih antara waktu yang dikirim dan waktu yang diterima. Data kemudian disimpan untuk analisis lebih lanjut. 3. Pengujian Berbagai Frekuensi Update: Eksperimen dilakukan dengan berbagai frekuensi update (500 ms, 1000 ms, 5000 ms) untuk mengamati dampaknya terhadap konsistensi dan latency. |

# **ANALISIS HASIL EKSPERIMEN**

|  |  |  |  |
| --- | --- | --- | --- |
| SSE   |  | | --- | | 500 ms | | 1000 ms | | 5000 ms | |

|  |  |  |  |
| --- | --- | --- | --- |
| WEBSOCKET   |  | | --- | | 500 ms | | 1000 ms | | 5000 ms | |

Hasil dari eksperimen menunjukkan perbedaan signifikan dalam latency antara WebSockets dan SSE. WebSockets menunjukkan latency yang lebih konsisten dalam situasi beban tinggi, sementara SSE menunjukkan beberapa keterlambatan dalam pengiriman data yang sangat cepat.

# **KESIMPULAN**

Dari eksperimen ini, dapat disimpulkan bahwa WebSockets lebih unggul dalam hal konsistensi data dan latency ketika digunakan dalam aplikasi real-time yang memerlukan pembaruan cepat. SSE dapat menjadi alternatif yang baik untuk aplikasi dengan kebutuhan yang lebih rendah, tetapi kurang stabil dalam kondisi beban tinggi. Hasil ini memberikan wawasan penting bagi pengembang dalam memilih protokol yang tepat berdasarkan kebutuhan aplikasi mereka.