

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah memberikan dampak signifikan pada cara kita mengakses dan mengelola informasi. Di dalam dunia web, kita sering menemui halaman-halaman yang bersifat statis, di mana fakta dan angka hanya diperbarui sesekali, menyebabkan informasi menjadi usang dan kadaluwarsa. Hal ini menjadi kendala, terutama dalam sektor-sektor di mana komunikasi *real-time* memiliki peran penting (Gupta dan M. P. Vani, 2018). Seiring dengan berjalannya waktu teknologi komunikasi telah mengalami banyak perubahan dari masa ke masa, salah satunya adalah komunikasi *real-time*. Fitur perpesanan menjadi salah satu bentuk komunikasi *real-time* yang banyak digunakan untuk menjalin komunikasi antar individu maupun kelompok (Thohir dkk., 2018).

Aplikasi perpesanan merupakan aplikasi yang bersifat *real-time*, untuk membuat aplikasi *real-time* terdapat beberapa teknik yang bisa digunakan. Dua teknik yang sering digunakan adalah long polling dan WebSocket. Long polling merupakan teknik dimana client melakukan request ke server namun server tidak akan langsung mengembalikan *response* (Pradana, 2017). WebSocket menyediakan koneksi dua arah antara klien dan server, memungkinkan pertukaran data secara *real-time*. Protokol WebSocket menerapkan komunikasi full-duplex sehingga memungkinkan lebih banyak interaksi antara klien dan situs web, memfasilitasi konten live, aplikasi *real-time* dan komunikasi dua arah (Zhangling dan Mao, 2012). Keberadaan WebSockets menjadi pilihan yang sangat relevan dibandingkan dengan metode konvensional dalam komunikasi *real-time*. Hal ini menjadi krusial terutama ketika kita mempertimbangkan kelemahan halaman web statis yang tidak dapat melakukan perubahan data secara *real-time* tanpa memerlukan refresh berulang (Gupta dan M. P. Vani, 2018).

Namun, pengguna semakin mengharapkan lebih banyak dari aplikasi perpesanan, seperti waktu respons yang lebih cepat, dan konsumsi energi yang lebih rendah menjadi tuntutan utama. Seiring dengan meningkatnya waktu respons

aplikasi, kepuasan pengguna cenderung menurun. Hal ini dapat menyebabkan penurunan jumlah pelanggan karena mereka beralih ke solusi lain (Hansson, 2020).

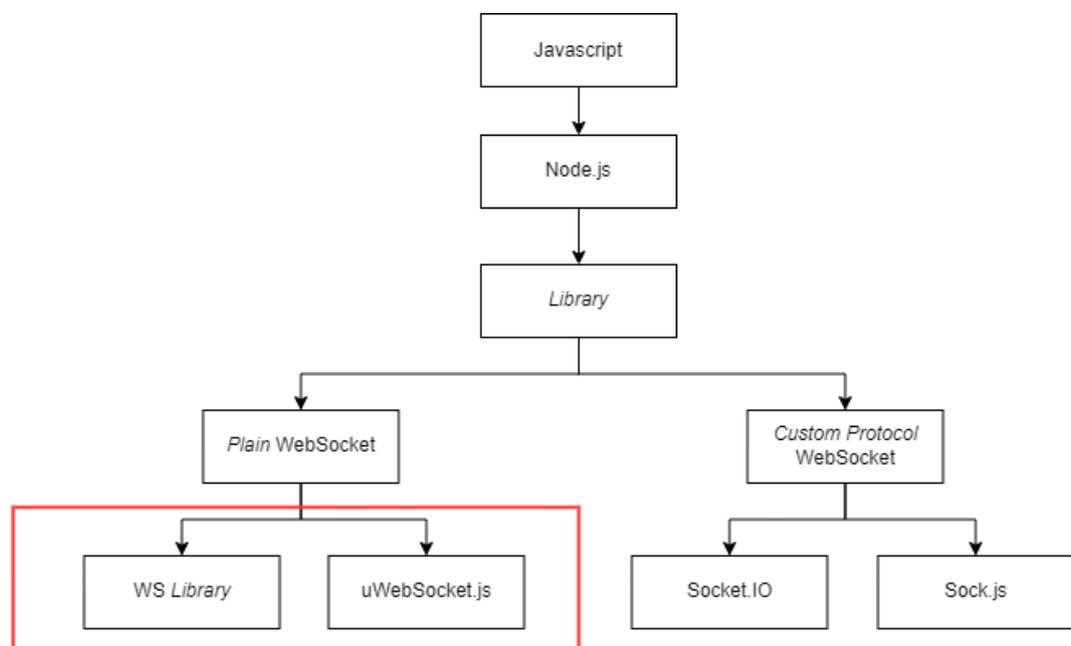
Dalam studi yang melibatkan 71,637 koneksi WebSocket dari 55,805 situs web, ditemukan bahwa WebSockets digunakan untuk berbagai kasus penggunaan di situs web populer. Beberapa kasus penggunaan yang paling umum adalah perpesanan, analitik, dan pembaruan langsung untuk skor olahraga dan harga saham. Perpesanan online membentuk sebagian besar koneksi WebSocket di situs-situs teratas. Sebanyak 94,9% koneksi tersebut melibatkan skrip yang berkomunikasi melalui API WebSocket dari pihak ketiga. Sebagian besar situs web mengandalkan layanan pihak ketiga untuk menyediakan fungsionalitas WebSocket mereka (Paul Murley, dkk, 2021).

Sebuah studi mengenai perbandingan antara *plain* WebSocket, *library* Socket.io, dan *library* Sock.js ditemukan bahwa terdapat perbedaan signifikan dalam hal kecepatan pengiriman data dan konsumsi daya antara *plain* WebSocket, *library* Socket.io, dan *library* Sock.js. Kedua hal tersebut dipengaruhi oleh *library* yang digunakan. Dari studi tersebut dapat disimpulkan bahwa *plain* WebSocket memiliki kecepatan dan konsumsi daya yang lebih baik dari penggunaan *library* Socket.io dan Sock.js. pemilihan *library* berpengaruh dalam kecepatan pengiriman data dan konsumsi daya (Hansson, 2020). Walaupun Socket.io saat memungkinkan menggunakan WebSocket sebagai transportasi, Socket.io menambahkan metadata tambahan pada setiap paket sama halnya pada Sock.js. (<https://socket.io>) (<https://github.com/sockjs/sockjs-client>). Hal ini membuktikan bahwa transportasi *plain* WebSocket lebih baik dari *library* yang sudah menambahkan metadata pada transportasi WebSocket. (Hansson, 2020).

WS dan uWebSocket.js merupakan 2 *library* yang menggunakan transportasi *plain* WebSocket tanpa menambahkan metadata pada setiap paketnya. Kedua *library* ini merupakan 2 *library* terpopuler dengan transportasi *plain* WebSocket atau aplikasi protokol yang sama dan di optimasi untuk Node.js dengan 21.5k *stars* pada WS dan 17.2k *stars* pada uWebSocket.js. (github.com/WS) (github.com/uNetworking/uWebSockets)

uWebSocket.js adalah implementasi WebSockets yang mudah dan aman digunakan serta dioptimalkan dengan baik sesuai standar. uWebSocket.js dilengkapi dengan dukungan pub/sub, routing URL, TLS 1.3, SNI, IPv6, permessage-deflate dan telah teruji sebagai salah satu implementasi paling populer, yang menjangkau jutaan pengguna setiap hari (github.com/uNetworking/uWebSockets). uWebSocket.js merupakan *library* C++ yang digunakan di sisi server sedangkan disisi client Objek API WebSocket menggunakan bahasa Javascript (Dykes dkk., 2018).

WS merupakan sebuah implementasi client dan server WebSocket yang mudah digunakan, sangat cepat, dan telah diuji secara menyeluruh untuk Node.js. WS bekerja pada *low level* yang berarti saat WebSocket server dibuat dengan menggunakan WS, WebSocket akan memulai listening untuk koneksi yang akan terhubung dari client dan setelah koneksi terhubung server akan menerima data dari klien. Data diterima sebagai *raw messages* yang berarti bisa menjadi string atau byte buffer (Sharma dan Agarwal, 2023)



Gambar 1.1 Hierarki *Library*

Pada Gambar 1.1 menggambarkan bahwa WS dan uWebsocket.js merupakan *library* yang berjalan di atas protokol yang sama yaitu *plain* WebSocket. Kedua library ini menggunakan bahasa pemrograman Javascript, yang dijalankan pada

runtime Node.js. Penelitian ini akan membandingkan kinerja WebSocket dengan menggunakan *library* WS dan uWebSocket.js. Pengujian performansi dilakukan untuk membandingkan kinerja dari komunikasi *real-time* dengan menggunakan *library* WS dan uWebSocket.js. Dengan membandingkan kinerja kedua *library* ini dalam konteks aplikasi perpesanan *real-time*, kita dapat mendapatkan pemahaman yang lebih baik tentang bagaimana teknologi ini dapat digunakan untuk memaksimalkan efisiensi dan efektivitas komunikasi *real-time*.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian yang diangkat, masalah dari penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana kinerja WebSocket dalam komunikasi aplikasi perpesanan ketika menggunakan *library* uWebSocket.js dan WS?
2. Bagaimana tingkat penggunaan sumber daya dari *library* WS dan uWebSocket.js saat menghadapi peningkatan jumlah pengguna?

1.3 Tujuan

Agar penelitian sesuai dengan tujuan, maka perlu adanya rumusan tujuan yang jelas. Sejalan dengan masalah penelitian yang dikemukakan di atas, tujuan penelitian ini adalah sebagai berikut:

1. Menganalisis dan membandingkan kinerja WebSocket dalam komunikasi aplikasi perpesanan dengan menggunakan *library* uWebSocket.js dan WS.
2. Menganalisis dan membandingkan tingkat penggunaan sumber daya dari *library* WS dan uWebSocket.js saat menghadapi peningkatan jumlah pengguna.

1.4 Manfaat

Beberapa manfaat yang diharapkan tercapai dalam penelitian ini adalah sebagai berikut:

1. Memberikan wawasan mendalam mengenai kinerja kedua *library* WebSocket dalam komunikasi aplikasi perpesanan.

2. Memberikan wawasan mendalam mengenai perbandingan tingkat penggunaan sumber daya dari *library* WS dan *uWebSocket.js* saat menghadapi peningkatan jumlah pengguna.

1.5 Batasan Penelitian

Adapun batasan yang ada pada penelitian ini adalah sebagai berikut:

1. Penelitian ini terfokus pada server WebSocket dengan menggunakan dua *library*, yaitu *uWebSocket* dan *WS*.
2. Implementasi *library* hanya mencakup fitur-fitur dasar dari setiap *library*.
3. Pengujian dilakukan dalam lingkungan *virtual machine* yang sama.

1.6 Struktur Penulisan

Sebagaimana yang telah dijelaskan pada Peraturan Rektor Universitas Pendidikan Indonesia Nomor 3260/UN40/HK/2018 Tentang Pedoman Penulisan Karya Ilmiah UPI Tahun Akademik 2018 disebutkan bahwa pada struktur penulisan memuat sistematika penulisan skripsi dengan memberikan gambaran pada setiap bab, urutan penulisannya dan keterkaitan antara bab satu dan lainnya. Berikut struktur penulisan pada penelitian ini.

1. BAB I: PENDAHULUAN

Pada bagian ini, mencakup latar belakang masalah yang menjadi dasar penelitian, rumusan masalah yang berisi pernyataan tentang masalah yang akan diinvestigasi dalam penelitian, tujuan yang menjelaskan tujuan utama dari penelitian, manfaat yang menguraikan manfaat yang diharapkan dari hasil penelitian, dan struktur penulisan

2. BAB II: KAJIAN PUSTAKA

Pada bagian ini dijelaskan pengertian dari objek-objek yang akan dibahas pada penelitian. Selain pengertian, terdapat juga konsep dan teori yang berhubungan dengan penelitian. Pada bagian ini juga dipaparkan hasil dari penelitian terdahulu yang relevan dan sesuai dengan topik bahasan.

3. BAB III: METODE PENELITIAN

Pada bagian ini mendeskripsikan secara rinci metode yang akan digunakan dalam penelitian dari desain penelitian, partisipan, populasi dan sampel, instrumen penelitian, prosedur penelitian, dan analisis data.

4. BAB IV: TEMUAN DAN BAHASAN

Bab ini akan menyajikan hasil temuan penelitian bersama dengan analisis yang relevan, sesuai dengan tujuan dan masalah penelitian yang telah dijelaskan sebelumnya.

5. BAB V: KESIMPULAN, IMPLIKASI, DAN REKOMENDASI

Bab ini akan menyampaikan kesimpulan dari hasil penelitian yang dilakukan berdasarkan rumusan masalah penelitian serta memberikan saran untuk penelitian selanjutnya

6. DAFTAR PUSTAKA

Daftar pustaka mencantumkan semua referensi yang digunakan dalam penelitian, termasuk buku, jurnal, artikel, atau sumber-sumber lainnya yang menjadi dasar teori dan penelitian.