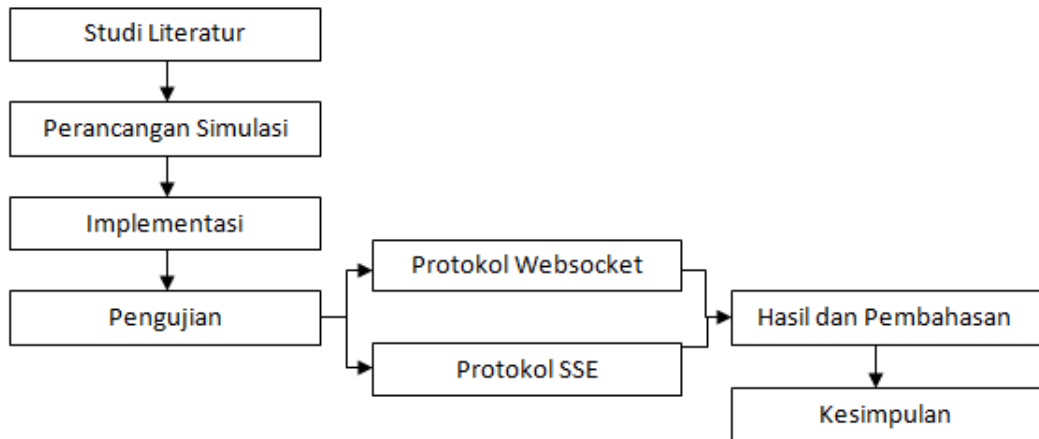


BAB 3 METODOLOGI

Pada bab ini akan dijelaskan langkah-langkah perancangan, implementasi, pengujian, serta analisa yang akan diterapkan pada penelitian ini. Hasil pengujian yang telah dianalisa akan diambil kesimpulan untuk menentukan tingkat efisien antara protokol Websocket dengan protokol server-sent event. Tahap-tahap dari keseluruhan proses penelitian dapat dilihat pada gambar 3.1.



Gambar 3.1 Skema Tahapan Metodologi Penelitian

3.1 Studi Literatur

Penelitian yang akan dilakukan bersifat analitik sehingga diperlukan studi literatur untuk mempelajari dan membangun simulasi perbandingan efisiensi antara kinerja protokol Websocket dengan SSE. Teori – teori pendukung yang akan digunakan antara lain:

1. *Push notification*
2. Protokol Websocket
3. Protokol SSE
4. Bahasa Pemrograman Python dan Javascript
5. Tools Psutil

Referensi yang digunakan untuk perancangan simulasi perbandingan efisiensi antara kinerja protokol Websocket dengan protokol SSE didapatkan dari berbagai sumber seperti buku, jurnal ilmiah, serta beberapa situs resmi.

3.2 Perancangan

Simulasi yang akan dirancang oleh penulis adalah simulasi mekanisme kerja pengiriman notifikasi yang dikirimkan oleh *server* ke *client* dengan merujuk pada tugas akhir Andrias Yudianto Putra yang berjudul “Pengembangan *Push notification* Menggunakan Websocket”. Pada penelitian ini, penulis akan

menambahkan perancangan dengan menggunakan protokol SSE untuk dapat membandingkan hasil berupa *delay* dan penggunaan CPU antara kedua protokol.

3.2.1 Kebutuhan Server dan Client

Penelitian ini dibagi menjadi 2 yaitu analisis kinerja protokol Websocket dan protokol SSE pada teknologi *push notification* yang diterapkan di jaringan lokal yang sama (WLAN). Berikut ini adalah tabel kebutuhan *server* dan *client* yang digunakan oleh penulis pada penelitian ini:

Tabel 3.1 Tabel Kebutuhan Server dan Client

	<i>Server</i>	<i>Client</i>
	Laptop	Smartphone
Bahasa Pemrograman	Python	Javascript
Framework	Flask	-
Library	Socket.IO	-
Sistem Operasi	Ubuntu 16.04	Android

Pada tabel 3.1 dapat dilihat bahwa perangkat yang berperan sebagai *server* adalah laptop dengan menggunakan sistem operasi Linux Ubuntu 16.04. Implementasi pengiriman notifikasi menggunakan Websocket dan SSE pada sisi *server* menggunakan bahasa pemrograman Python. Selain itu juga menggunakan *framework* Flask dan *library* Socket.IO. *Framework* Flask berfungsi untuk merancang *server* yang didukung oleh *library* Socket.IO.

Untuk perangkat yang berperan sebagai *client* adalah *smartphone* dengan menggunakan sistem operasi Android. Bahasa pemrograman yang digunakan adalah Javascript. Javascript berfungsi untuk merancang halaman situs yang akandiakses oleh *client* melalui *browser*.

Untuk pengiriman notifikasi menggunakan protokol Websocket umumnya digunakan perantara atau *broker*, akan tetapi pada penelitian ini *server* dirancang untuk sekaligus berperan menjadi *broker* karena *client* yang digunakan untuk pengujian tidak terlalu banyak. Sehingga peran *broker* eksternal tidak terlalu berpengaruh.

Sedangkan untuk pengiriman notifikasi menggunakan protokol SSE tidak menggunakan *broker* seperti pada Websocket. *Client* yang terhubung ke *server* akan mendapatkan notifikasi dari waktu ke waktu tiap kali terdapat pembaruan pada sisi server selama koneksi tidak terputus atau *session* pada *browser* berakhir.

3.3 Implementasi

Tahap selanjutnya adalah implementasi rancangan yang telah dibuat sebelumnya pada tahap perancangan. Tahap ini berisi penjelasan tentang proses implementasi dari rancangan penelitian. Implementasi yang dilakukan adalah implementasi protokol Websocket dan SSE pada server, implementasi *client* pada perangkat Android, dan implementasi *tools* untuk mendapatkan *resource* kinerja pengiriman masing-masing protokol pada sisi server.

Berikut ini penjelasan dari implementasi yang dilakukan:

1. Implementasi Server

Implementasi pada server mencakup proses instalasi *framework* flask, instalasi *library* Socket.IO, konfigurasi *push service* menggunakan pada protokol Websocket dan protokol SSE, dan implementasi halaman antar muka untuk admin

2. Implementasi Client

Implementasi pada client menggunakan javascript untuk membuat halaman antarmuka yang berisi notifikasi saat pengujian pengiriman notifikasi protokol Websocket dan protokol SSE. Halaman antarmuka berupa halaman situs yang diakses pada *browser* masing-masing client

3. Implementasi Tools Pengujian

Implementasi *tools pengujian* pada penelitian ini mencakup proses instalasi modul *timestamp* pada sisi server dan instalasi modul *psutil* pada sisi server.

3.4 Pengujian

Tahap ini berisi penjelasan tentang proses pengujian setelah melakukan tahap implementasi untuk pengiriman notifikasi menggunakan Websocket dan SSE agar diperoleh hasil berdasarkan pada tujuan penelitian ini. Pengujian mencakup pada kinerja kedua protokol untuk pengiriman notifikasi dan mencari seberapa banyak *resource* yang digunakan oleh *server* untuk melakukan pengiriman. Pengujian dilakukan dengan tujuan untuk membandingkan kinerja antara protokol Websocket dengan protokol SSE. Pengujian dilakukan di jaringan lokal dengan menggunakan komponen pengujian seperti perangkat *smartphone* yang menggunakan sistem operasi android (berperan sebagai *client*) dan laptop yang menggunakan sistem operasi Linux (berperan sebagai *server*). Parameter uji yang digunakan adalah parameter *delay* yang berfungsi sebagai alat ukur waktu pengiriman notifikasi dari kedua protokol yang akan diuji.

Pada penelitian ini dilakukan dua jenis pengujian yaitu pengiriman notifikasi dengan jumlah *client* yang berbeda untuk mendapatkan nilai delay dari kedua protokol. *Client* yang digunakan adalah sejumlah 1 *client*, 3 *client* dan 6 *client*.

Pengujian kedua adalah pengujian *monitoring* pada *resource server* untuk mengetahui seberapa banyak *resource* yang digunakan oleh *server*. *Resource* yang digunakan adalah penggunaan CPU. Tabel 3.2 adalah tabel skenario pengujian yang dilakukan pada penelitian ini.

Tabel 3.2 Tabel skenario pengujian

Jenis Pengujian	Skenario pengujian	Parameter Pengujian
Perulangan Pengiriman Informasi berupa Notifikasi	<ul style="list-style-type: none"> - Pengiriman pada 1 <i>client</i> - Pengiriman pada 3 <i>client</i> - Pengiriman pada 6 <i>client</i> 	<ul style="list-style-type: none"> - Delay pengiriman notifikasi
<i>Monitoring Resource</i> pada sisi <i>Server</i>	<ul style="list-style-type: none"> - Pengiriman pada 1 <i>client</i> - Pengiriman pada 3 <i>client</i> - Pengiriman pada 6 <i>client</i> 	<ul style="list-style-type: none"> - Penambahan nilai penggunaan CPU pada sisi <i>server</i> (dalam persen)

Skenario pengujian pengiriman notifikasi dengan jumlah *client* yang berbeda dilakukan dengan cara melakukan pengiriman notifikasi sebanyak 10 notifikasi, 100 notifikasi dan 1000 notifikasi. Perbedaan jumlah notifikasi yang diukur bertujuan untuk mendapatkan hasil rata-rata *delay* pada tiap pengujian. Notifikasi yang dikirimkan berupa pesan teks dengan ukuran 4 karakter, yaitu "Test". Notifikasi akan dikirimkan pada 1 *client*, 3 *client* dan 6 *client*. Hasil yang didapatkan dari pengujian ini adalah rata-rata *delay* dari protokol Websocket dan protokol SSE pada tiap skenario pengujian.

Untuk skenario pengujian *monitoring resource* pada sisi *server*, hasil yang didapatkan berupa penambahan CPU *usage*. Hasil yang diperoleh akan dianalisis untuk mengetahui pengaruh penambahan jumlah *client* dengan nilai CPU *usage* pada tiap pengujian. Pengujian ini dilakukan dengan cara melakukan instalasi modul psutil pada sisi *server* dengan menerapkan interval waktu sehingga *resource* yang terekam tidak langsung ditampilkan dalam jumlah yang banyak.

3.5 Kesimpulan

Pada tahap kesimpulan akan dicantumkan hasil analisis dari pengujian terhadap sistem yang telah dibuat serta hasil perbandingan kinerja antara protokol Websocket dengan protokol SSE. Data yang didapat dari pengujian untuk masing – masing protokol dianalisa sehingga akan diperoleh kesimpulan protokol mana yang mempunyai kinerja lebih efisien.