**LAPORAN EKSPERIMEN MENGENAI PROBLEM PERBANDINGAN SWR DAN REACT QUERY DALAM PENGELOLAAN DATA FETCHING DAN CACHE**

*Laporan ini disusun untuk memenuhi tugas 3 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

Sarah 221524059

Septyana Agustina 221524060

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](#_Toc175471873)

[A. IDENTIFIKASI PROBLEM 1](#_Toc175471874)

[B. DESKRIPSI PROBLEM 1](#_Toc175471875)

[C. METODOLOGI EKSPERIMEN 1](#_Toc175471876)

[D. PELAKSANAAN EKSPERIMEN 2](#_Toc175471877)

[E. ANALISIS HASIL EKSPERIMEN 3](#_Toc175471878)

# **IDENTIFIKASI PROBLEM**

Saat membangun aplikasi web modern, terutama yang memerlukan pengambilan data dari API eksternal, dua aspek utama yang harus dipertimbangkan adalah kecepatan respon dan penggunaan bandwidth. Keduanya mempengaruhi pengalaman pengguna dan efisiensi sistem secara keseluruhan. Pengelolaan data fetching dan cache menjadi penting untuk memastikan performa aplikasi tetap optimal meskipun jumlah permintaan API yang tinggi. Dua library populer yang sering digunakan untuk menangani masalah ini adalah SWR dan React Query.

Problem yang diidentifikasi adalah:

* Seberapa baik SWR dan React Query dalam menangani pengelolaan data fetching dan cache pada aplikasi web?
* Bagaimana perbandingan kedua library ini dalam hal waktu respon dan penggunaan bandwidth?

# **DESKRIPSI PROBLEM**

Dalam pengembangan aplikasi React, pengelolaan pengambilan data (fetching) dari API merupakan salah satu aspek yang sangat penting. Library seperti SWR dan React Query menyediakan alat bantu untuk mempermudah proses ini, namun masing-masing memiliki pendekatan yang berbeda. Masalah yang dihadapi oleh developer adalah memilih library yang tepat berdasarkan kecepatan, efisiensi, dan kemudahan penggunaan.

Waktu respon adalah indikator utama performa aplikasi. Semakin cepat data bisa diambil dan ditampilkan kepada pengguna, semakin baik pula pengalaman pengguna.

Penggunaan bandwidth berhubungan dengan efisiensi jaringan. Penggunaan data yang lebih rendah berarti aplikasi lebih hemat dalam hal transfer data, yang penting terutama bagi pengguna dengan jaringan terbatas.

Untuk memahami perbedaan performa antara SWR dan React Query, perlu dilakukan eksperimen komparatif yang mengukur waktu respon dan penggunaan bandwidth dalam skenario pengambilan data dari API.

# **METODOLOGI EKSPERIMEN**

Metode eksperimen ini dirancang untuk mengukur waktu respon dan penggunaan bandwidth dari aplikasi React yang menggunakan SWR dan React Query untuk melakukan data fetching dari API eksternal.

Langkah-langkah:

1. **Set Up Aplikasi**:

* Dibuat dua aplikasi React terpisah: satu menggunakan SWR dan yang lain menggunakan React Query.
* Kedua aplikasi mengambil data dari API yang sama, yaitu daftar isu dari repositori GitHub React (<https://api.github.com/repos/facebook/react/issues>).

1. **Pengukuran Waktu Respon**:

* Gunakan **Chrome DevTools** untuk mengukur waktu respon (dalam milidetik) yang diperlukan untuk mendapatkan data dari API.
* Lakukan beberapa kali refresh untuk mendapatkan hasil yang konsisten.

1. **Pengukuran Bandwidth**:

* Gunakan tab **Network** di Chrome DevTools untuk mengukur jumlah data yang ditransfer (dalam kilobyte) saat pengambilan data dari API.

1. **Simulasi Jaringan Lambat**:

* Pengujian dilakukan dalam kondisi normal tanpa throttling untuk mendapatkan hasil default dari kedua library.

Alat Pengukuran:

* **Chrome DevTools** digunakan untuk mengukur waktu respon dan penggunaan bandwidth.
* **React Profiler** digunakan untuk memastikan bahwa proses rendering berjalan lancar tanpa ada kendala signifikan.

# **PELAKSANAAN EKSPERIMEN**

Eksperimen dilaksanakan dengan cara berikut:

1. Pembuatan Aplikasi SWR:

* Aplikasi React dengan SWR dipasang dan data dari GitHub API diambil menggunakan SWR.
* Aplikasi diakses dan waktu respon serta penggunaan bandwidth diukur menggunakan Chrome DevTools.

1. Pembuatan Aplikasi React Query:

* Aplikasi kedua dibuat menggunakan React Query dengan pengambilan data dari API yang sama.
* Sama seperti pada aplikasi SWR, waktu respon dan bandwidth diukur melalui Chrome DevTools.

# **ANALISIS HASIL EKSPERIMEN**

1. Waktu Respon

|  |  |
| --- | --- |
| Library | Waktu Respon (ms) |
| SWR | 112.672 |
| React Query | 135.288 |

Dari hasil pengujian, SWR menunjukkan waktu respon yang lebih cepat dibandingkan React Query. Waktu respon SWR adalah **112.672 ms**, sedangkan React Query adalah **135.288 ms**. Artinya, dalam pengambilan data dari API, SWR memberikan waktu respon yang lebih baik sekitar 17% lebih cepat.

1. Penggunaan Bandwidth

|  |  |
| --- | --- |
| Library | Penggunaan Bandwidth (kB) |
| SWR | 413 |
| React Query | 451 |

Dari hasil pengujian penggunaan bandwidth, SWR juga lebih efisien dibandingkan React Query. Penggunaan bandwidth SWR adalah **413 kB**, sementara React Query mengonsumsi **451 kB**, yang berarti React Query mengonsumsi lebih banyak data.

1. Analisis Cache

SWR dan React Query sama-sama memiliki mekanisme caching yang baik, namun **React Query** memiliki fitur caching yang lebih canggih dengan opsi konfigurasi seperti staleTime dan cacheTime. Sementara itu, **SWR** lebih minimalis dalam pengaturan cache, tetapi cukup efisien untuk kasus penggunaan sederhana.

1. Simulasi Jaringan Lambat

Dalam simulasi jaringan normal (tanpa throttling), kedua library menunjukkan performa yang baik. Namun, pada jaringan lambat (Slow 3G), cache dari kedua library memainkan peran penting. Ketika data sebelumnya sudah di-cache, kedua aplikasi dapat menampilkan data lebih cepat meskipun koneksi lambat. Namun, waktu respon awal tetap lebih cepat pada SWR.