**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN PERBANDINGAN TEKNIK CACHING DENGAN METODE ETAG DAN CACHE CONTROL**

*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

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](#_Toc176232797)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc176232798)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc176232799)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc176232800)

[**D.** **PELAKSANAAN EKSPERIMEN** 1](#_Toc176232801)

[**E.** **ANALISIS HASIL EKSPERIMEN** 2](#_Toc176232802)

# **IDENTIFIKASI PROBLEM**

Masalah yang diidentifikasi adalah efisiensi teknik caching HTTP, khususnya dalam perbandingan antara ETag dan Cache-Control. Fokus eksperimen ini adalah untuk menentukan mana dari kedua teknik tersebut yang lebih efektif dalam mengelola caching untuk aplikasi web dalam hal penggunaan bandwidth dan waktu respons.

# **DESKRIPSI PROBLEM**

Caching adalah teknik penting dalam pengembangan web untuk meningkatkan performa aplikasi dan mengurangi beban pada server. Dua teknik caching yang umum digunakan adalah ETag dan Cache-Control:

* ETag: Merupakan header HTTP yang digunakan untuk menentukan apakah konten telah berubah sejak terakhir kali di-cache. Server mengirimkan ETag dalam respons dan klien menyertakan ETag dalam permintaan berikutnya. Jika ETag masih valid, server dapat mengembalikan status 304 Not Modified, yang berarti konten tidak perlu diunduh ulang.
* Cache-Control: Merupakan header HTTP yang mengontrol bagaimana dan berapa lama konten dapat di-cache. Ini dapat mencakup instruksi seperti max-age, no-cache, dan no-store untuk mengelola durasi cache dan kebijakan penyimpanan.

# **METODOLOGI EKSPERIMEN**

Eksperimen ini dilakukan untuk membandingkan efektivitas ETag dan Cache-Control dalam hal:

1. Penggunaan Bandwidth: Mengukur total data yang diterima dan dikirim selama pengujian.
2. Waktu Respons: Mengukur rata-rata waktu respons server untuk setiap permintaan.
3. Jumlah Permintaan dan Iterasi: Mengukur jumlah total permintaan dan iterasi yang dilakukan selama pengujian.

Pengujian dilakukan menggunakan alat k6 untuk simulasi beban, dengan 100 pengguna virtual (VUs) selama 1 menit. Dua skrip pengujian terpisah digunakan, satu untuk ETag dan satu untuk Cache-Control.

# **PELAKSANAAN EKSPERIMEN**

1. Implementasi API

* ETag: API dikonfigurasi untuk menggunakan header ETag dalam respons. Jika resource tidak berubah, server mengirimkan respons 304 Not Modified.
* Cache-Control: API dikonfigurasi untuk menggunakan header Cache-Control dengan pengaturan cache yang sesuai.

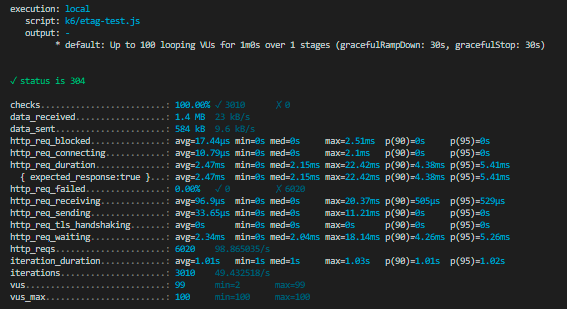
1. Pengujian dengan k6

* Parameter Pengujian:
  + Jumlah Pengguna Virtual: 100
  + Durasi: 1 menit
* Skrip yang Digunakan:
  + etag-test.js: Menguji API dengan ETag.
  + cache-control-test.js: Menguji API dengan Cache-Control.

# **ANALISIS HASIL EKSPERIMEN**

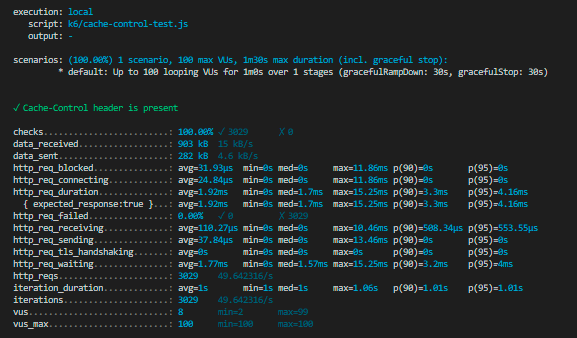
1. Hasil:

Etag:



* Keberhasilan Cek: 100.00%
* Data Diterima: 1.4 MB
* Data Dikirim: 584 kB
* Waktu Respons:
* Rata-rata: 2.47 ms
* P90: 4.38 ms
* P95: 5.41 ms
* Jumlah Permintaan: 6020
* Jumlah Iterasi: 3010

Cache Control:



* Keberhasilan Cek: 100.00%
* Data Diterima: 903 kB
* Data Dikirim: 282 kB
* Waktu Respons:
* Rata-rata: 1.92 ms
* P90: 3.3 ms
* P95: 4.16 ms
* Jumlah Permintaan: 3029
* Jumlah Iterasi: 3029

1. Analisis Perbandingan

* Penggunaan Bandwidth:
  + ETag: Lebih banyak data yang diterima (1.4 MB) karena header ETag sering menyebabkan permintaan untuk memeriksa apakah konten telah berubah. Namun, banyak data yang tidak perlu dikirim ulang (584 kB).
  + Cache-Control: Mengurangi data yang dikirim dan diterima karena pengaturan cache yang mengurangi jumlah permintaan ke server.
* Waktu Respons:
  + ETag: Rata-rata waktu respons sedikit lebih tinggi (2.47 ms) dibandingkan Cache-Control. Ini mungkin karena overhead tambahan dalam memproses ETag.
  + Cache-Control: Rata-rata waktu respons lebih cepat (1.92 ms), menunjukkan bahwa caching sisi klien lebih efisien dalam konteks ini.
* Jumlah Permintaan dan Iterasi:
  + ETag: Memproses lebih banyak permintaan (6020) dengan lebih sedikit iterasi (3010), yang menunjukkan lebih banyak permintaan untuk memeriksa apakah konten telah berubah.
  + Cache-Control: Mengelola lebih sedikit permintaan (3029) dengan iterasi yang sama (3029), menandakan penggunaan cache yang lebih efisien.

1. Pros dan Cons

ETag:

* Pros:
  + Menyediakan kontrol lebih granular atas cache.
  + Berguna untuk konten yan gsering berubah.
* Cons:
  + Mungkin menambah overhead karena perlu memeriksa status Etag pada setiap permintaan.
  + Lebih banyak data yang diterima dan dikirim dibandingkan dengan cache control.

Cache Control:

* Pros:
  + Mengurangi penggunaan bandwidth dengan mengandalkan cache sisi klien.
  + Waktu respon lebih cepat dalam pengujian ini.
* Cons:
  + Kurang kontrol granular jika dibandingkan dengan Etag.
  + Tidak selalu cocok untuk konten yang sering diperbarui.

1. Kesimpulan

Berdasarkan hasil eksperimen, Cache-Control menunjukkan performa yang lebih baik dalam hal waktu respons dan penggunaan bandwidth. Namun, ETag mungkin lebih sesuai untuk aplikasi yang membutuhkan validasi yang lebih granular dan sering memperbarui konten. Pilihan antara keduanya sebaiknya didasarkan pada kebutuhan spesifik aplikasi dan pola akses data.