

PENGEMBANGAN WEB (TEORI)

LAPORAN EKSPERIMEN: NODE.JS VS RUBY (RUBY O RAILS) PADA APLIKASI WEB FULL-STACK

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
A. IDENTIFIKASI PROBLEM	1
B. DESKRIPSI PROBLEM	1
C. METODOLOGI EKSPERIMEN	1
D. PELAKSANAAN EKSPERIMEN	2
E. ANALISIS HASIL EKSPERIMEN.....	2
F. KESIMPULAN.....	3

A. IDENTIFIKASI PROBLEM

Pada eksperimen ini, tujuan utama adalah membandingkan kinerja antara Node.js dan Ruby dalam konteks aplikasi web yang melibatkan pengambilan data dan pemrosesan asinkron. Indikator yang akan dianalisis meliputi waktu muat (load time), waktu koneksi (connect time), latensi, ukuran data yang dikirim dan diterima, serta kode respons dari server.

B. DESKRIPSI PROBLEM

Node.js dan Ruby (khususnya Ruby on Rails) sering digunakan untuk pengembangan aplikasi web full-stack. Namun, ada perdebatan mengenai efisiensi masing-masing dalam menangani tugas asinkron seperti I/O operasi, pengambilan data, dan pengolahan respon dari server. Tujuan eksperimen ini adalah untuk mengukur waktu respons dan performa dari Node.js serta Ruby dalam situasi serupa. Hasil eksperimen akan menunjukkan perbedaan kinerja kedua platform dalam menangani permintaan HTTP.

C. METODOLOGI EKSPERIMEN

1) Tujuan

Tujuan dari eksperimen ini adalah untuk membandingkan efisiensi concurrency antara **Node.js** dan **Ruby** dalam menjalankan tugas-tugas asynchronous. Efisiensi diukur berdasarkan throughput, latency, dan penggunaan CPU dalam skenario pemrosesan data secara bersamaan.

2) Desain Eksperimen

Desain eksperimen ini melibatkan pembuatan dua aplikasi yang melakukan operasi CRUD sederhana dan menangani permintaan HTTP secara bersamaan. Aplikasi pertama dibangun menggunakan **Node.js**, dan aplikasi kedua menggunakan **Ruby (Ruby on Rails)**. Setiap aplikasi diuji dengan beban yang sama, yaitu sejumlah request yang dijalankan secara bersamaan untuk mensimulasikan tugas asynchronous yang intensif.

3) Parameter Eksperimen

Eksperimen menggunakan beberapa parameter yang sama untuk kedua platform:

- **Jumlah request:** Menggunakan skenario dengan 100, 500, dan 1000 request secara bersamaan.
- **Jenis operasi:** Operasi CRUD sederhana pada data berbasis JSON.

- **Tools pengukuran:** Menggunakan JMeter untuk mengirim request secara serentak dan mencatat hasil pengukuran.

4) Metode Pengukuran

Pengukuran dilakukan dengan menggunakan beberapa metrik:

- **Throughput:** Mengukur jumlah request yang berhasil diselesaikan per detik.
- **Latency:** Waktu yang dibutuhkan oleh sistem untuk merespons setiap request.
- **Error rate:** Mengukur jumlah error yang terjadi selama pemrosesan.

D. PELAKSANAAN EKSPERIMEN

1) Node.js

- Implementasi server Node.js dengan endpoint HTTP GET yang merespons dengan JSON.
- Data permintaan diambil dan diukur menggunakan JMeter untuk mendapatkan waktu muat, waktu koneksi, latensi, serta ukuran data yang dikirim dan diterima.
- Hasil dari eksperimen Node.js sebagai berikut:

```
Thread Name:Test Node.js 1-1
Sample Start:2024-09-13 16:29:10 WIB
Load time:144
Connect Time:5
Latency:144
Size in bytes:491
Sent bytes:121
Headers size in bytes:336
Body size in bytes:155
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:OK
```

2) Ruby:

- Tidak dapat dilakukan pengujian karena terdapat error instalasi library psych pada Ruby, yang merupakan dependensi penting untuk menjalankan Ruby on Rails. Beberapa percobaan instalasi telah dilakukan, tetapi kesalahan ini terus muncul, sehingga pengujian Ruby tidak dapat dilanjutkan.

E. ANALISIS HASIL EKSPERIMEN

1) Node.js

- Waktu koneksi yang dihasilkan sebesar 5 ms menunjukkan bahwa Node.js mampu melakukan negosiasi koneksi dengan cepat.

- Waktu muat total sebesar 144 ms, yang mengindikasikan bahwa Node.js memiliki latensi yang rendah dalam merespons permintaan HTTP.
- Tidak ada error (Error Count: 0) dan kode respons adalah 200 (OK), yang menunjukkan bahwa permintaan HTTP berhasil dijalankan dengan baik.
- Ukuran data total yang diterima adalah 491 bytes, dengan body response sebesar 155 bytes dan header sebesar 336 bytes, menandakan bahwa data JSON dikirim dengan efisien.

2) Ruby

- Tidak dapat dianalisis karena error instalasi psych yang terus terjadi. Library ini diperlukan untuk menangani parsing YAML, yang merupakan bagian penting dari Ruby on Rails.

F. KESIMPULAN

Berdasarkan eksperimen ini, Node.js menunjukkan hasil yang sangat baik dengan latensi rendah, waktu koneksi yang cepat, dan tidak ada error selama pengujian. Sayangnya, untuk Ruby, pengujian tidak dapat dilakukan karena masalah instalasi library psych. Kesimpulan ini hanya berlaku untuk Node.js, dan pengujian Ruby harus diulang setelah masalah instalasi diselesaikan.