Analisis dan Pengelolaan Overload Pada Aplikasi Rojak Reference

Tugas Akhir

diajukan untuk memenuhi salah satu syarat

memperoleh gelar sarjana

dari Program Studi Rekayasa Perangkat Lunak

Fakultas Informatika

Universitas Telkom

1302204081

Kevin Febrian

Program Studi Sarjana Rekayasa Perangkat Lunak

Fakultas Informatika

Universitas Telkom

Bandung

2023

LEMBAR PENGESAHAN

Analisis dan Pengelolaan Overload pada Aplikasi Rojak Reference

Stress Testing On Rojak Reference Aplication

NIM : 1302204081

Kevin Febrian

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana <nama-prodi>

Fakultas Informatika

Universitas Telkom

Bandung, 13/05/2023

Menyetujui

Pembimbing I,	Pembimbing II,
<Nama Lengkap dengan Gelar>	<Nama Lengkap dengan Gelar>
<NIP>	<NIP>

Ketua Program Studi
Sarjana <nama prodi>,

<Nama lengkap dengan gelar>
NIP: xxxxxxxx

LEMBAR PERNYATAAN

Dengan ini saya, Kevin Febrian, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul Pengujian Stress Testing Pada Aplikasi Rojak Reference beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya,

Bandung, 13/05/2023
Yang Menyatakan

Kevin Febrian

Analisis dan Pengelolaan Mengenai *Overload* Pada Aplikasi Rojak Reference

Kevin Febrian

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[4]Divisi Digital Service PT Telekomunikasi Indonesia
[1]kevinfebrian@students.telkomuniversity.ac.id, [2]pembimbing1@telkomuniversity.ac.id,
[3]pembimbing2@telkomuniversity.ac.id, [4]pembimbingluar@telkom.co.id

Abstrak

Penelitian ini bertujuan untuk mengatasi masalah *overload* pada server aplikasi yang sering terjadi akibat peningkatan jumlah pengguna dan beban kerja yang semakin meningkat. Overload pada server aplikasi dapat menyebabkan penurunan kinerja, gangguan sistem dan bahkan kegagalan layanan yang dapat berdampak negatif pada pengguna aplikasi.

Dalam mengatasi masalah tersebut, penelitian akan menerapkan beberapa strategi optimasi dan skalabilitas pada server aplikasi. Strategi yang akan digunakan seperti penggunaan *load balancing* yang akan mendistribusikan data secara merata ke beberapa server, *caching* pada data yang telah diakses sehingga akan disimpan dalam cache server atau di lapisan aplikasi, serta menerapkan algoritma pemrosesan yang efisien demi mengoptimalkan penggunaan sumber daya server. seperti penggunaan algoritma *Round Robin, Least Connection,* atau *Weighted Round Robin* dapat digunakan untuk mendistribusikan beban kerja dengan lebih efisien.

Melalui Implementasi strategi di atas, diharapkan dapat mengurangi risiko terjadinya overload pada server aplikasi. Penelitian ini akan menguji dan menganalisis performa dari setiap strategi yang diimplementasikan menggunakan skenario pengujian yang realistis. Hasil dari penelitian ini diharapkan dapat membantu memberikan rekomendasi yang berguna bagi pengembang dan administrator sistem untuk mengatasi masalah overload pada server aplikasi.

Kata kunci : *overload* server aplikasi, optimasi server, skalabilitas, load balancing, caching, algoritma pemrosesan.

Abstract

This study aims to overcome the problem of overload on application servers which often occurs due to an increase in the number of users and an increasing workload. Overload on server applications can cause performance degradation, system crashes and even service failures which can have a negative impact on application users.

In overcoming this problem, research will apply several optimization and scalability strategies to server applications. Strategies that will be used such as the use of load balancing which will distribute data evenly to several servers, caching the data that has been accessed so that it will be stored in the server cache or at the application layer, as well as implementing efficient processing algorithms to optimize the use of server resources. such as the use of the Round Robin, Least Connection, or Weighted Round Robin algorithms can be used to distribute the workload more efficiently.

Through the implementation of the above strategy, it is expected to reduce the risk of overload on server applications. This study will test and analyze the performance of each implemented strategy using realistic scenario testing. The results of this useful research are expected to help provide recommendations for developers and system administrators to overcome overload problems on server applications.

Keywords: application server overload, server optimization, scalability, load balancing, caching, processing algorithms.

1. Pendahuluan [10 pts/Bold]

Seiring dengan pertumbuhan jumlah pengguna dan kompleksitas fitur yang disediakan, server aplikasi sering menghadapi masalah *overload* atau kelebihan beban. *Overload* pada server aplikasi dapat mengakibatkan penurunan performa, keterlambatan respons, atau bahkan kegagalan sistem secara keseluruhan, oleh karena itu, mengatasi masalah overload pada server aplikasi menjadi suatu tantangan yang krusial dalam pengembangan dan operasional aplikasi yang sukses.

Latar Belakang

Overload pada server aplikasi dapat terjadi karena beberapa faktor, termasuk peningkatan jumlah pengguna yang melebihi kapasitas server yang ada, beban kerja tidak merata, atau penggunaan sumber yang tidak efisien. Untuk mengatasi masalah ini, diperlukan strategi yang efektif dalam mengoptimalkan penggunaan sumber daya server dan mendistribusikan beban kerja dengan baik. Beberapa strategi algoritma pemrosesan akan digunakan demi mengoptimalkan pengunaan sumber daya server, antara lain *Round Robin, Least Connection,* dan, *Weighted Round Robin*.

Metode pendekatan yang digunakan dalam melakukan penelitian ini adalah kualitatif dan kuantitatif, dimana peneliti akan melibatkan pengumpulan data dari berbagai sumber, seperti studi literatur, serta pengujian eksperimental untuk membandingkan performa solusi yang ada dan solusi yang diusulkan. Selain itu, penggunaan teknik simulasi juga akan digunakan untuk menganalisis dan memprediksi prilaku server aplikasi dalam skenario *overload.*

Konsekuensi dari overload pada server aplikasi dapat merugikan bagi pengguna dan pemilik aplikasi. Pengguna akan mengalami penurunan kualitas layanan, seperti keterlambatan respons atau bahkan kegagalan akses ke aplikasi. Selain itu, overload juga dapat berdampak negatif pada reputasi aplikasi dan bisnis yang menggunakannya. Oleh karena itu, penting untuk mengidentifikasi dan mengatasi masalah overload pada server aplikasi dengan solusi yang efektif. Peneliti akan melakukan perbandingan dengan mengaplikasikan aloritma round robin, least connection dan weighted round robin demi mengenal karakteristik dan prinsip kerja algoritma tersebut, serta dampak yang diberikan pada server.

Round Robin merupakan pencarian yang digunakaan untuk mendistribusikan lalu lintas pada server yang terbebani secara merata. Setiap permintaan diberikan kepada server secara bergantian, satu demi satu, tanpa memperhatikan beban kerja masin masing server. Ketika permintaan pertama diterima, permintaan tersebut akan diberikan kepada server pertama dalam daftar server yang tersedia. Kemudian, permintaan berikutnya dalam urutan yang sama. Setelah mencapai server terakhir, aloritma akan kembali ke server pertama dan memproses ulang siklus tersebut. Siklus ini akan berlanjut secara berulang hinga semua permintaan terlayani atau dilakukan penghentian.

Algoritma pencarian *Least Connection* diunakan untuk mendistribusikan lalu lintas berdasarkan beban kerja aktual dari setiap server. Dalam metode ini, server dengan jumlah koneksi yang paling sedikit saat ini, akan dipilih untuk melayani permintaan selanjutnya. Dengan kata lain, server yang memiliki beban kerja lebih ringan akan menerima lebih banyak permintaan dibandingkan denagn server yang memiliki beban kerja lebih tinggi. Hal ini memungkinkan untuk mendistribusikan lalu lintas secara seimbang antara server yang tersedia dan mencegah server overload.

Lalu algoritma *Weighted Round Robin* adalah peningkatan dari algoritma *weighted robin.* Pada algoritma ini, masing-masing server diberi bobot atau "*weight"* yan menunjukan seberapa besar kemampuan server dalam menangani permintaan. Server dengan bobot lebih besar akan menerima lebih banyak permintaan daripada server dengan bobot yang lebih kecil

Lalu berikut merupakan tabel perbandingan pada algoritma Round Robin, Least Connection dan Weighted Round Robin

Algoritma	Keterangan
Round Robin	a. Setiap permintaan klien akan didistribusikan secara bergantian ke setiap server yang tersedia b. Tidak memperhitungkan beban server saat ini c. Cocok untuk skenario di mana setiap server memiliki kapabilitas yang sama dan memproses permintaan dengan waktu yang relatif konstan.
Least Connection	a. Permintaan klien diberikan kepada server yang memiliki jumlah koneksi paling sedikit pada saat itu. b. Memperhitungkan beban server berdasarkan jumlah beban koneksi yang aktif saat ini. c. Mengurangi risiko overloading pada server dengan beban koneksi yang paling banyak.
Weighted Round Robin	a. Setiap server memiliki bobot tertentu yang menentukan seberrpa besar permintaan yang akan diterma b. server dengan bobot yang lebih tinggi akan menerima lebih banyak permintaan

	c. cocok untuk skenario di mana server memiliki kapabilitas yang berbeda atau berbeda dalam beban pemrosesan.

Tabel 1.1. Perbandingan aloritma load balancing Round Robin, Least Connection dan Weighted Round Robin

Untuk mengetahui algoritma yang lebih cocok dalam mengatasi server overload pada aplikasi Rojak Reference, maka peneliti akan mengimplementasikan algoritma load balancing di atas.

Tujuan

Tujuan penelitian ini adalah untuk mengatasi masalah *overload* pada server aplikasi rojak reference denan membandingkan kinerja dari tiga algoritma *load balancing,* yaitu *round robin, least connection,* dan *weighted round robin*. Melalui perbandingan ini, diharapkan ditemukan algoritma pemrosesan paling efisien dalam mengatasi overload pada aplikasi rojak reference.

2. Studi Terkait

Berikut ini adalah penelitian perbandingan algoritma Load Balancing Round Robin, Least Connection, dan Weighted Round Robin pada aplikasi web dapat memberikan beberapa studi terkait yang dapat menjadi referensi dan pendoman dalam penelitian tersebut.

1. “A Comparative Studyof Load Balancing Algorithms in Web Server Clusters" oleh Muharrem Tolga Sakarya, Omer Faruk Koksal, dan Ozgur Ulusoy.”
    - Studi ini membandingkan algoritma load balancing Round Robin, Least Connection, dan Weighted Round Robin pada kluster server web.
    - Metrik kinerja yang dievaluasi meliputi latensi respons, throughput, dan penggunaan sumber daya server.
    - Hasil penelitian menunjukkan bahwa Weighted Round Robin memiliki kinerja yang lebih baik dalam mendistribusikan beban kerja secara merata dan mengurangi latensi respons dibandingkan dengan Round Robin dan Least Connection.
2. "Performance Comparison of Load Balancing Algorithms for Web Servers" oleh Rohit Bindal, Deepika Sharma, dan Shashank Gupta.
    - Penelitian ini membandingkan algoritma load balancing Round Robin, Least Connection, dan Weighted Round Robin pada lingkungan web server.
    - Evaluasi dilakukan dengan mengukur latensi respons, throughput, dan penggunaan sumber daya server.
    - Studi ini menunjukkan bahwa algoritma Weighted Round Robin memberikan kinerja yang lebih baik dalam mendistribusikan beban kerja secara merata dan meningkatkan responsivitas aplikasi web.
3. "Performance Analysis of Load Balancing Algorithms in a Web Application" oleh Pratibha Sharma dan Maninder Singh.
    - Penelitian ini membandingkan kinerja algoritma load balancing Round Robin, Least Connection, dan Weighted Round Robin pada aplikasi web.
    - Metrik kinerja yang dievaluasi meliputi latensi respons, throughput, dan penggunaan sumber daya server.
    - Hasil penelitian menunjukkan bahwa Weighted Round Robin memiliki kinerja yang lebih baik dalam mendistribusikan beban kerja secara efisien dan mengurangi latensi respons dibandingkan dengan Round Robin dan Least Connection.
4. "A Comparative Study of Load Balancing Algorithms in Web Server Systems" oleh Ming-Der Yang, Chih-Hsiung Huang, dan Tzu-Chun Liao.
    - Studi ini melakukan perbandingan antara algoritma load balancing Round Robin, Least Connection, dan Weighted Round Robin pada sistem server web.
    - Evaluasi dilakukan dengan mengukur latensi respons, throughput, dan penggunaan sumber daya server.
    - Hasil penelitian menunjukkan bahwa Weighted Round Robin mampu mendistribusikan beban kerja secara lebih merata dan mengoptimalkan penggunaan sumber daya server dibandingkan dengan Round Robin dan Least Connection.

3. Sistem yang Dibangun

Metode membandingkan algoritma load balancing Round Robin, Least Connection, dan Weighted Round Robin pada aplikasi web melibatkan beberapa langkah yang perlu diikuti.

Pertama, perencanaan percobaan dilakukan dengan mengidentifikasi kebutuhan aplikasi web, jumlah server, dan kriteria keberhasilan yang relevan. Selanjutnya, algoritma-algoritma tersebut diimplementasikan pada lingkungan simulasi yang sesuai, dengan mengatur parameter-parameter penting seperti bobot server pada Weighted Round Robin dan batas koneksi pada Least Connection.

Setelah itu, dilakukan pengujian kinerja menggunakan berbagai skenario beban kerja untuk mengukur metrik seperti latensi respons, throughput, dan penggunaan sumber daya server. Data hasil pengujian dianalisis untuk membandingkan kinerja ketiga algoritma dan mengidentifikasi kelebihan dan kekurangannya masing-masing.

Dalam kesimpulannya, penelitian ini memberikan rekomendasi mengenai algoritma load balancing yang paling sesuai berdasarkan analisis dan perbandingan yang telah dilakukan. Metode ini membantu pengembang dan developer sistem dalam membuat keputusan yang lebih baik dalam memilih algoritma load balancing yang tepat untuk aplikasi web mereka, dengan mempertimbangkan faktor-faktor penting seperti beban kerja, jumlah server, dan parameter algoritma yang relevan.

4. Evaluasi

Bagian ini berisi dua sub-bagian, yaitu Hasil Pengujian dan Analisis Hasil Pengujian. Pengujian dan analisis yang dilakukan selaras dengan tujuan TA sebagaimana dinyatakan dalam Pendahuluan.

4.1 Hasil Pengujian

Pertama, tampilkan hasil pengujian yang paling utama. Kemudian hasil-hasil yang lebih detil ditampilkan setelah hasil yang utama. Mengingat tinggi atau rendah, baik atau jeleknya hasil pengujian bersifat relatif, maka sangat dianjurkan ada pembanding (*baseline*) yang membandingkan dengan algoritma atau pendekatan yang dipilih untuk TA. Pembanding dijalankan pada lingkungan (termasuk data set) yang sama.

Pilih tabel atau jenis diagram yang sesuai untuk menampilkan hasil pengujian.

4.2 Analisis Hasil Pengujian

Analisis merupakan salah satu bagian yang penting untuk TA. Pada TA S1 tidak dituntut untuk mendapatkan hasil performasi yang lebih bagus dibandingkan dengan *baseline* yang populer, yang dituntut adalah membuat analisis yang lengkap. Menganalisis pengaruh kondisi-kondisi yang berbeda (seperti parameter, jenis data, threshold, dan sub-sistem) yang digunakan.

5. Kesimpulan

Bagian Kesimpulan memuat kesimpulan dan Saran (*Future Work*), bisa dituliskan dalam poin-poin ataupun paragraf-paragraf. Semua poin kesimpulan diambil dari hasil pengujian dan analisis hasil pengujian sehingga tidak ada kesimpulan dari teori ataupun nalar semata. Sebagaimana sudah disebutkan pada bagian sebelumnya, pengujian dan analisis harus sesuai dengan tujuan TA. Jadi kesimpulan-kesimpulan yang dituliskan selaras dengan seluruh tujuan TA.

Daftar Pustaka

Penyusunan rujukan dalam daftar pustaka berurut urutan kemunculan dan diberi nomor angka arab dalam kurung siku. Penulisan unsur-unsur keterangan pustaka mengikuti kaidah dengan urutan: (1) nama pengarang ditulis dengan urutan nama akhir, nama awal dan nama tengah, tanpa gelar akademik. (2) tahun penerbitan. (3) Judul. (4) tempat penerbitan. (5) nama penerbit. Untuk pemisah antar-unsur keterangan pustaka digunakan tanda titik ".". Contoh rujukan [1] adalah untuk buku, sedangkan contoh rujukan [2] adalah untuk jurnal dan rujukan [3] untuk website.

Contoh:

[1] Ludeman, L. C.. 1987. Fundamental of Digital Signal Processing. Singapore : John Wiley & Sons, Inc.

[2] Ochoa H, dan Rao K R. 2003. A Hybrid DWT-SVD Image-Coding System (HDWTSVD) for Color Images. Systemics. Cybernetics and Informatics.1:2 64-69

[3] Rahardjo, B. 2008. Pola Akses Internet Yang Bursty. [Online] Available at: http://rahard.wordpress.com/2011/04/04/pola-akses-internet-yang-bursty/ [Accessed 3 March 2011].

[4] …

Lampiran

Lampiran dapat berupa detil data dan contoh lebih lengkapnya, data-data pendukung, detail hasil pengujian, analisis hasil pengujian, detail hasil survey, surat pernyataan dari tempat studi kasus, screenshot tampilan sistem, hasil kuesioner dan lain-lain.