



Analisis Efektivitas Edge Caching dalam Meningkatkan QoS pada Layanan Web Multimedia

Azzahra Setya^{1*}, Riska Amanda², Agnes Olganita³, Leanna Vidya⁴, Tody Ariefianto⁵

^{1,2,3,4,5}Faculty of Electrical Engineering, Telkom University

*Korespondensi: azzarasw@telkomuniversity.ac.id

Info Artikel

Diterima 10
September 2024

Disetujui 20
Oktober 2024

Dipublikasikan 30
November 2024

Keywords:
Internet, Apache
Traffic Server,
Edge Cache,
Multimedia,
HTTP/HTTPS

© 2024 The
Author(s): This is
an open-access
article distributed
under the terms of
the Creative
Commons
Attribution
ShareAlike (CC BY-
SA 4.0)



Abstrak

Perkembangan internet telah berkembang selama beberapa dekade terakhir sebagai teknologi penting dalam bidang pengiriman konten dan pengoptimalan jaringan, terutama untuk website multimedia yang belakangan ini banyak digunakan sehingga transmisi konten memerlukan kecepatan akses dan penyimpanan yang lebih dioptimalkan. Permintaan konten dapat dialokasikan berdasarkan penyimpanan tepi sehingga tidak terdapat permintaan konten yang berulang terutama pada waktu yang bersamaan. Penelitian ini akan menganalisis edge caching sebagai metode penyimpanan yang digunakan dalam jaringan pengiriman konten. Metode yang digunakan dari edge caching adalah untuk meningkatkan kinerja, efisiensi, dan keandalan metode tersebut dengan menyimpan data atau konten yang sering diakses lebih dekat dengan pengguna akhir. Dalam implementasi edge caching, Apache Traffic Server (ATS) digunakan sebagai proxy server yang bertindak sebagai gerbang antara pengguna dengan internet. Pengujian dilakukan pada website multimedia dengan protocol HTTP dan HTTPS serta skenario multiple user dan multiple request. Tujuan akhir yang ingin dicapai adalah mengakomodasi permintaan konten pengguna dan memindahkan penyimpanan konten lebih dekat dengan pengguna sehingga dapat mengurangi latensi dan backhaul tautan.

Abstract

The development of the internet has grown over the past few decades as an important technology in the field of content delivery and network optimization, especially for multimedia websites that have recently been widely used so that content transmission requires more optimized access speed and storage. Content requests can be allocated based on edge storage so that there are no repeated content requests, especially at the same time. This study will analyze edge caching as a storage method used in content delivery networks. The method used from edge caching is to improve the performance, efficiency, and reliability of the method by storing frequently accessed data or content closer to the end user. In the implementation of edge caching, Apache Traffic Server (ATS) is used as a proxy server that acts as a gateway between users and the internet. Testing is carried out on multimedia websites with HTTP and HTTPS protocols as well as multiple user and multiple request scenarios. The ultimate goal to be achieved is to accommodate user content requests and move content storage closer to users so as to reduce latency and link backhaul.

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi yang pesat telah mendorong kebutuhan akan layanan web multimedia yang semakin kompleks dan beragam. Layanan seperti video streaming, konferensi daring, dan aplikasi berbasis media lainnya kini menjadi bagian tak terpisahkan dari kehidupan sehari-hari. Namun, peningkatan permintaan ini menimbulkan tantangan besar dalam hal kualitas layanan (Quality of Service/QoS), terutama yang terkait dengan latensi, kecepatan transfer data, dan ketersediaan jaringan. Salah satu pendekatan untuk mengatasi tantangan tersebut adalah penerapan edge caching, sebuah metode penyimpanan konten di tepi jaringan (edge) yang lebih dekat dengan pengguna akhir. Edge caching memiliki potensi besar untuk meningkatkan QoS dengan mengurangi waktu akses, mengurangi beban jaringan inti, dan meningkatkan efisiensi distribusi konten. Dalam konteks layanan web multimedia, penerapan edge caching memungkinkan pengguna untuk mengakses konten dengan lebih cepat dan stabil, terutama dalam lingkungan dengan lalu lintas data yang tinggi. Meskipun demikian, efektivitas edge caching dalam meningkatkan QoS pada layanan web multimedia sangat bergantung pada faktor-faktor seperti algoritma caching yang digunakan, jenis konten yang di-cache, serta pola akses pengguna. Penelitian yang mendalam diperlukan untuk menganalisis sejauh mana metode ini dapat dioptimalkan dalam berbagai skenario penggunaan, termasuk pengaruhnya terhadap metrik QoS seperti latensi, throughput, dan kehilangan paket.

International Telecommunication Union (ITU) dalam laporan estimasi trafik IMT pada tahun 2015, menyatakan bahwa dengan berkembangnya teknologi komunikasi seluler, diprediksi akan terjadi peningkatan pengguna internet menjadi 9,8 miliar pada tahun 2024. Hal ini dapat dibuktikan oleh Cisco (2020) bahwa pada tahun 2023 terdapat 5,7 miliar pengguna internet. Cisco white paper (2023) menyatakan bahwa semua perangkat yang terhubung ke jaringan IP akan meningkat lebih dari tiga kali lipat populasi global, yang menyebabkan ledakan trafik data secara global. Menurut uji kecepatan yang dilakukan oleh Speedtest by Ookla Global Index, kecepatan jaringan di Indonesia pada bulan September 2023 mengalami penurunan kecepatan download menjadi 27,65 Mbps. Faktor yang menyebabkan penurunan kecepatan download ini adalah keterbatasan bandwidth, hal ini mempengaruhi kecepatan broadband yang terjadi ketika mengakses konten yang sama dalam waktu yang bersamaan.

Edge caching telah diidentifikasi sebagai teknologi potensial untuk meningkatkan QoS pada layanan web multimedia, masih terdapat kesenjangan dalam implementasinya di berbagai skenario praktis. Salah satu masalah utama adalah kurangnya pemahaman mendalam tentang bagaimana algoritma caching tertentu dapat dioptimalkan untuk jenis konten dan pola penggunaan spesifik (Suartana & Prapanca, 2020). Sehingga, banyak implementasi edge caching yang belum mencapai efisiensi optimal, terutama dalam mengatasi permasalahan seperti lonjakan lalu lintas data secara tiba-tiba atau kebutuhan real-time dari layanan multimedia. Untuk mengatasi kesenjangan tersebut, diperlukan pendekatan komprehensif yang mencakup analisis menyeluruh terhadap kebutuhan pengguna, karakteristik konten, serta dinamika jaringan (Aksenta et al, 2023). Salah satu solusi potensial adalah pengembangan algoritma caching adaptif yang mampu secara otomatis menyesuaikan strategi penyimpanan berdasarkan pola akses pengguna dan kondisi jaringan. Selain itu, simulasi dan pengujian sistem pada skala besar

dalam lingkungan yang lebih realistis perlu dilakukan untuk mengevaluasi efektivitas edge caching di berbagai scenario (Anjani, 2018). Pendekatan ini tidak hanya dapat meningkatkan efisiensi penyimpanan konten, tetapi juga mendukung distribusi layanan multimedia yang lebih andal dan responsif, sehingga meningkatkan QoS secara keseluruhan (Saroji et al, 2021).

Solusi yang menjanjikan untuk mengurangi latensi dan biaya jaringan dari pengiriman konten adalah dengan mendekatkan konten kepada pengguna akhir melalui penyimpanan terdistribusi di seluruh jaringan, yang disebut sebagai penempatan konten atau caching. Jika konten yang diminta tersedia di penyimpanan local pengguna, maka konten tersebut dapat langsung dilayani tanpa harus dikirim melalui internet. Dengan cara ini, caching memungkinkan pengurangan beban backhaul yang signifikan selama waktu lalu lintas puncak sehingga dapat mengurangi kemacetan pada jaringan. Dengan menerapkan caching pada “tepi”, dapat mengurangi latensi, menghemat bandwidth, dan meningkatkan kecepatan akses karena penyimpanan semakin dekat dengan pengguna akhir. Implementasi edge caching dengan Apache Traffic Server (ATS) sebagai proxy server memberikan kinerja, skalabilitas, dan manfaat efisiensi biaya yang signifikan. Penelitian ini bertujuan untuk menganalisis efektivitas penerapan edge caching dalam meningkatkan QoS pada layanan web multimedia. Penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan solusi yang lebih efisien dan adaptif dalam memenuhi kebutuhan layanan multimedia yang terus berkembang.

2. Metode Penelitian

Penelitian ini menggunakan Packet Network Emulator Tool Lab (PNETLab) sebagai platform simulasi dan emulasi jaringan yang memungkinkan pengguna untuk membuat, menkonfigurasi, dan menguji topologi jaringan yang kompleks dalam lingkungan virtual. Platform ini digunakan untuk mengimplementasikan dan menilai hasil sistem edge caching dalam konteks komputasi jaringan. Penelitian sebelumnya telah dilakukan mengenai implementasi sistem edge caching pada Video on Demand (VoD). Penelitian ini melibatkan pengujian transfer file yang menunjukkan nilai throughput yang cenderung rendah atau dapat dikatakan sebagai throughput yang baik. Oleh karena itu, penelitian ini bertujuan untuk menguji sistem edge caching dengan mengimplementasikannya pada website multimedia sehingga dapat mengedepankan kualitas layanan.

Analisis data yang efektif pada edge caching dalam meningkatkan QoS pada layanan web multimedia, penelitian ini menggunakan beberapa teknik pengumpulan data sebagai berikut: 1) Penelitian ini akan melakukan simulasi menggunakan perangkat lunak berbasis jaringan, seperti Mininet atau NS-3, untuk mensimulasikan implementasi edge caching. Eksperimen ini akan mencakup pengujian berbagai algoritma caching (misalnya, Least Recently Used (LRU), Least Frequently Used (LFU), atau algoritma adaptif) pada skenario jaringan dengan parameter tertentu, seperti latensi, throughput, dan kehilangan paket. Data dari simulasi akan dianalisis untuk mengukur QoS berdasarkan metrik yang relevan; 2) Pengumpulan data juga dilakukan melalui studi kasus pada infrastruktur jaringan nyata, seperti server edge di Content Delivery Network (CDN). Data lalu lintas jaringan, pola akses pengguna, dan waktu respons layanan akan dikumpulkan untuk mengevaluasi kinerja sistem edge caching dalam lingkungan operasional nyata; 3) Untuk melengkapi data teknis dan survei akan dilakukan terhadap pakar jaringan dan administrator IT yang terlibat dalam implementasi edge caching. Pendapat mereka mengenai tantangan teknis,

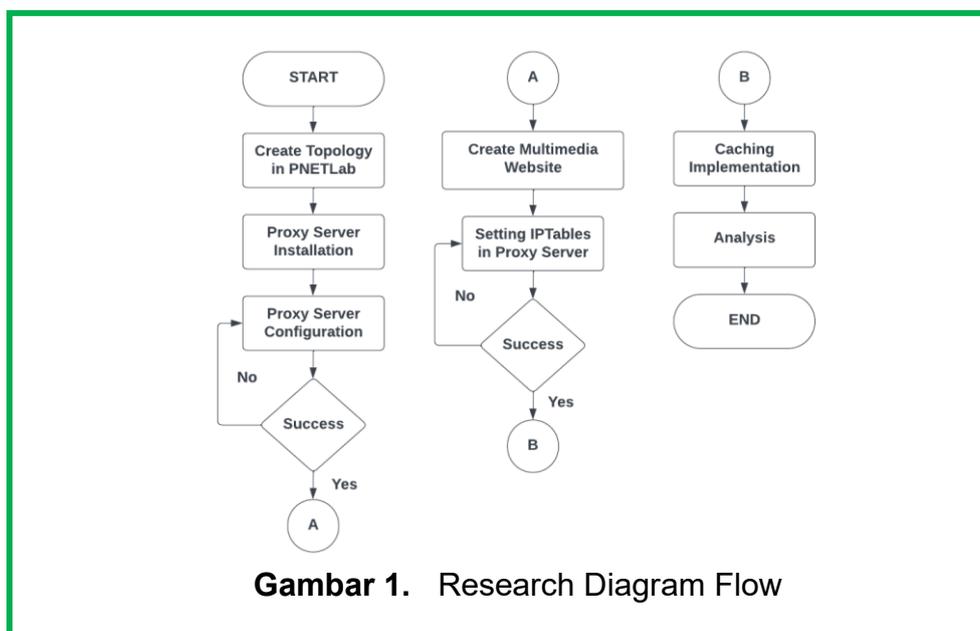
efektivitas metode yang digunakan, dan rekomendasi untuk perbaikan akan memberikan wawasan tambahan; 4) Data historis dari log server dan jaringan akan dianalisis untuk mengidentifikasi pola akses pengguna dan tingkat keberhasilan caching. Analisis ini juga membantu mengevaluasi sejauh mana edge caching dapat mengurangi latensi dan beban jaringan.

Analisis data kuantitatif yang dikumpulkan dari simulasi dan log jaringan, seperti latensi, throughput, kehilangan paket, dan hit ratio cache, akan dianalisis menggunakan statistik deskriptif untuk mendapatkan gambaran umum tentang kinerja edge caching. Visualisasi data dalam bentuk grafik atau tabel digunakan untuk mempermudah interpretasi hasil. Hasil dari berbagai skenario simulasi dan studi kasus akan dibandingkan untuk mengidentifikasi pola-pola yang konsisten. Perbandingan dilakukan untuk menilai kinerja edge caching pada jenis konten, pola akses pengguna, dan kondisi jaringan yang berbeda. Model kinerja edge caching yang dikembangkan melalui simulasi akan dievaluasi terhadap metrik QoS yang diukur. Teknik evaluasi seperti Mean Opinion Score (MOS) atau metode perbandingan metrik (misalnya, trade-off antara latensi dan throughput) digunakan untuk menilai dampak caching pada pengalaman pengguna secara keseluruhan.

3. Hasil dan Pembahasan

3.1 Hasil

Alur kerja edge caching dimulai dengan pembuatan topologi jaringan di PNETLab, yang menjadi dasar pengaturan sistem secara keseluruhan. Setelah itu, langkah berikutnya adalah menginstalasi server proxy, dalam hal ini menggunakan Apache Traffic Server, yang berfungsi untuk menangani permintaan dan pengiriman data secara efisien. Proses konfigurasi server proxy dilakukan untuk memastikan fungsionalitasnya sesuai dengan kebutuhan caching data di edge. Selanjutnya, IPTables dikonfigurasi untuk mendukung implementasi edge caching, mengelola lalu lintas data dan memastikan pengalihan permintaan data melalui server proxy sesuai dengan aturan yang telah ditentukan.



Gambar 1. Research Diagram Flow

Penelitian ini menggunakan Packet Network Emulator Tool Lab (PNETLab) sebagai emulator jaringan untuk menganalisa parameter jaringan dengan sistem edge caching. Bagian ini menjelaskan metode yang digunakan dalam sistem edge caching, termasuk desain sistem (flowchart) dan topologi jaringan yang digunakan. Seperti yang digambarkan pada Fig 1, alur kerja edge caching dimulai dengan membuat topologi di PNETLab Menginstalasi server proxy yaitu Apache Traffic Server, mengkonfigurasi server proxy kemudian mengatur IPTables untuk implementasi edge caching.

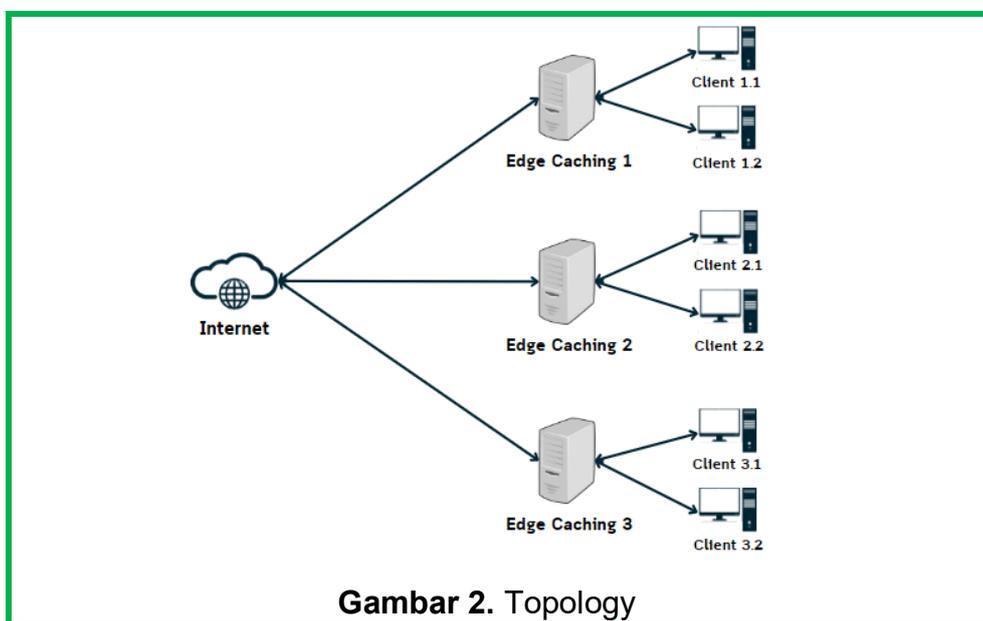
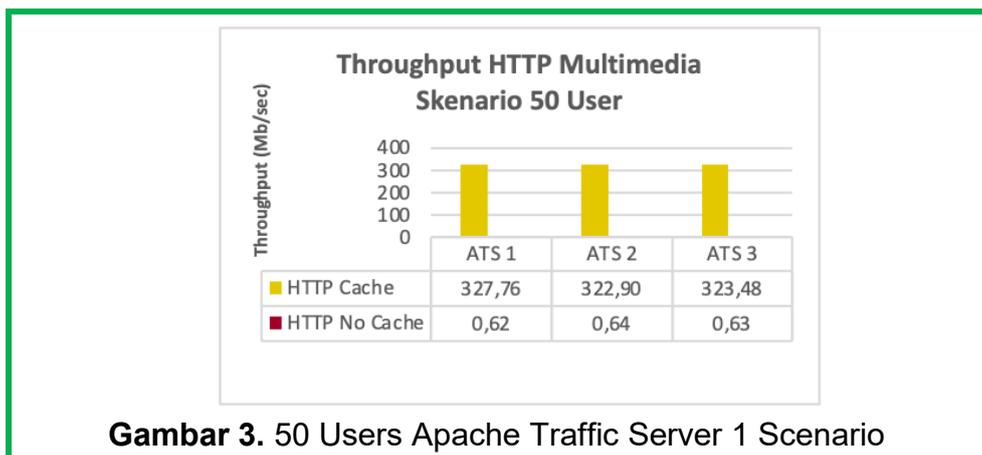


Fig 2 mengilustrasikan topologi penerapan edge caching pada protokol HTTP dan HTTPS dengan situs web multimedia dan situs web video-on-demand. Ketika klien ingin mengakses situs web kita, baik itu multimedia atau video-on-demand, mereka akan melewati Apache Traffic Server terlebih dahulu sebagai server proxy. Kemudian, server proxy akan meminta konten secara langsung ke server web. Jika klien lain ingin mengakses konten yang sama, server proxy akan memeriksa apakah cache konten tersebut ada. Jika ada, konten akan dikirim ke klien tanpa melalui server. Implementasi edge caching dilakukan dengan melakukan skenario permintaan pengguna untuk parameter yang telah ditentukan. Implementasi edge caching dipantau menggunakan karakteristik Quality of Service (QoS) seperti throughput, waktu respons, dan waktu yang telah berlalu. Seorang pengguna mengirimkan permintaan (misalnya, permintaan HTTP atau HTTPS) untuk sumber daya (halaman web, gambar, video) untuk mengakses situs web atau aplikasi web. Pengujian ini dilakukan dengan parameter umum yang sama: throughput, waktu respons, dan waktu yang berlalu.

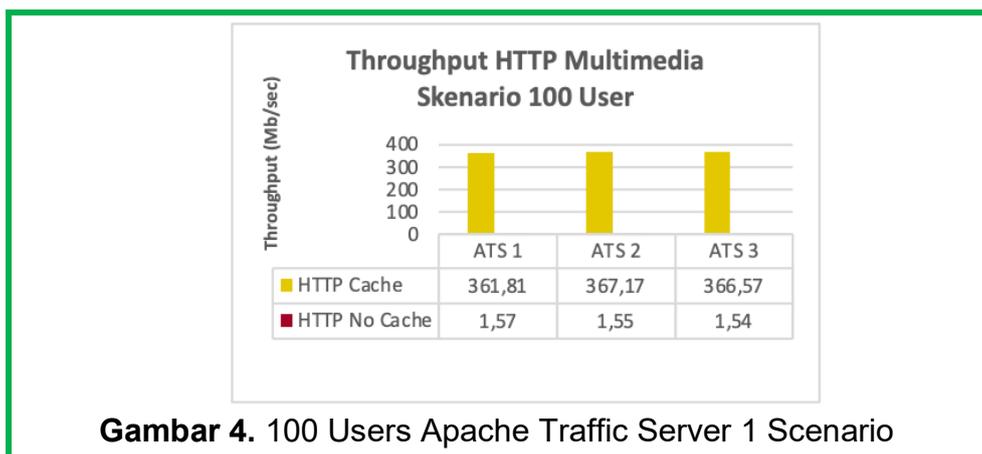
Tabel 1. Edge Caching Testing Scenarios

No	Number of User Scenarios					
	Edge Caching 1		Edge Caching 2		Edge Caching 3	
	Client 1.1	Client 1.2	Client 2.1	Client 2.2	Client 3.1	Client 3.2
1	50 User	50 User	50 User	50 User	50 User	50 User
2	100 User	100 User	100 User	100 User	100 ser	100 User

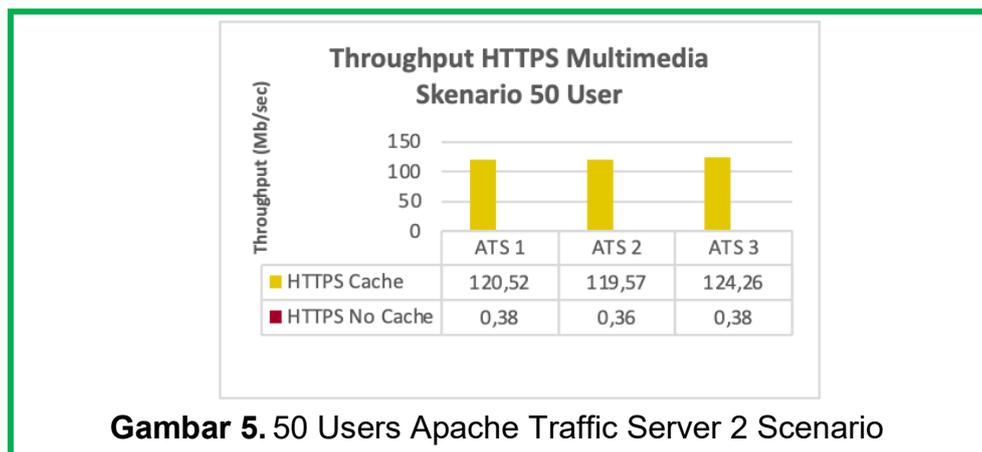
Skenario yang dilakukan pada makalah ini adalah dengan menggunakan topologi seperti pada Fig 2. yang dirancang untuk melihat bagaimana perbedaan Quality of Service berdasarkan permintaan pengguna ketika mengakses website HTTP dan HTTPS serta website video-on-demand. Makalah penelitian ini juga bertujuan untuk menunjukkan perbedaan nilai parameter yang signifikan antara caching dan tanpa caching. Penelitian ini menggunakan tiga buah proxy server yang masing-masing memiliki satu buah client. Klien dibagi menjadi dua skenario, 50 pengguna dan 100 pengguna dengan masing-masing kapasitas lima permintaan. Pengujian pada website menggunakan tool load test Siege. Siege merupakan tools pengujian performa yang bertujuan untuk mengukur performa sebuah server dalam menangani beban yang dihasilkan oleh pengguna melalui proses-proses yang berjalan secara bersamaan.



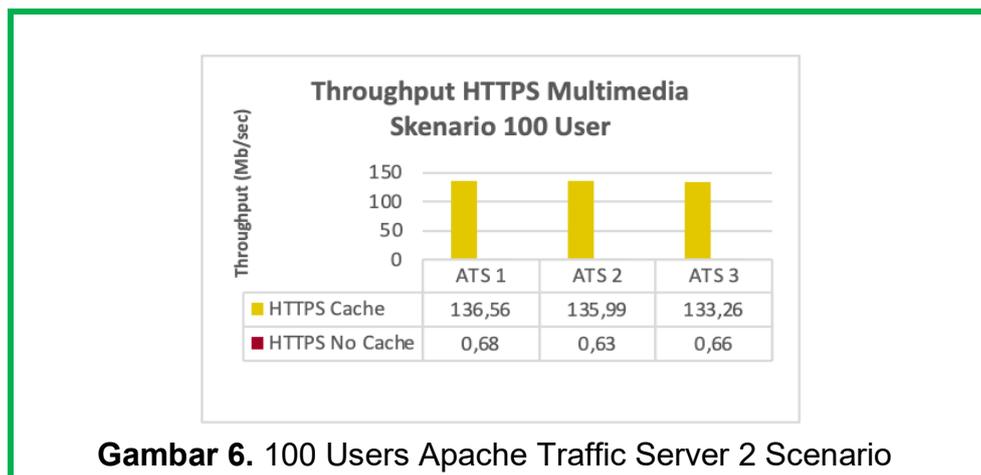
Gambar 3. 50 Users Apache Traffic Server 1 Scenario



Gambar 4. 100 Users Apache Traffic Server 1 Scenario



Gambar 5. 50 Users Apache Traffic Server 2 Scenario



Gambar 6. 100 Users Apache Traffic Server 2 Scenario

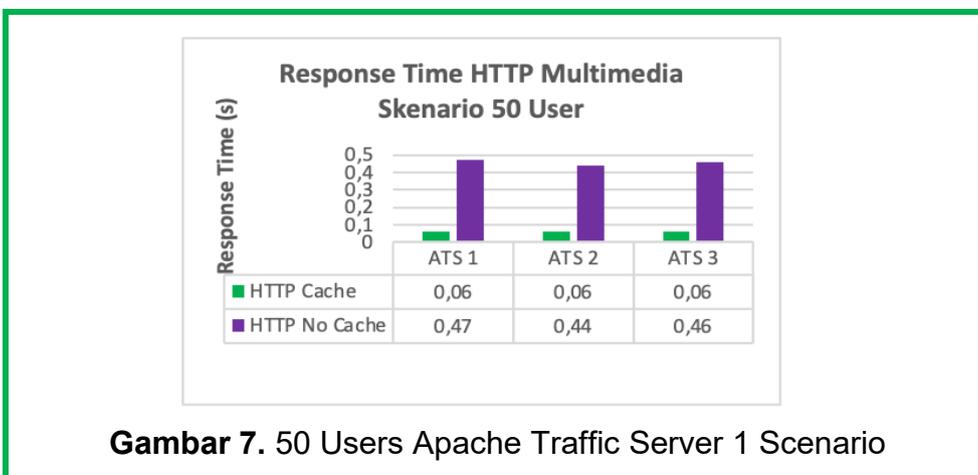
Throughput

Throughput adalah salah satu metrik penting yang berpengaruh terhadap performa sistem edge caching. Throughput yang tinggi memungkinkan data atau konten yang disimpan di edge cache untuk dikirim lebih cepat ke pengguna akhir. Ini meningkatkan kualitas pengalaman pengguna, terutama dalam aplikasi yang membutuhkan respons waktu nyata. Dengan throughput yang tinggi, waktu yang diperlukan untuk mentransfer data dari cache ke pengguna menjadi lebih singkat. Hal ini dapat secara signifikan mengurangi latensi, yang merupakan salah satu tujuan utama dari edge caching. Edge caching bertujuan untuk menyimpan konten yang sering diakses dekat dengan pengguna akhir. Dengan throughput yang tinggi, sistem dapat melayani lebih banyak permintaan dari cache lokal tanpa harus mengambil data dari server pusat, sehingga mengurangi beban pada jaringan backbone dan menghemat bandwidth.

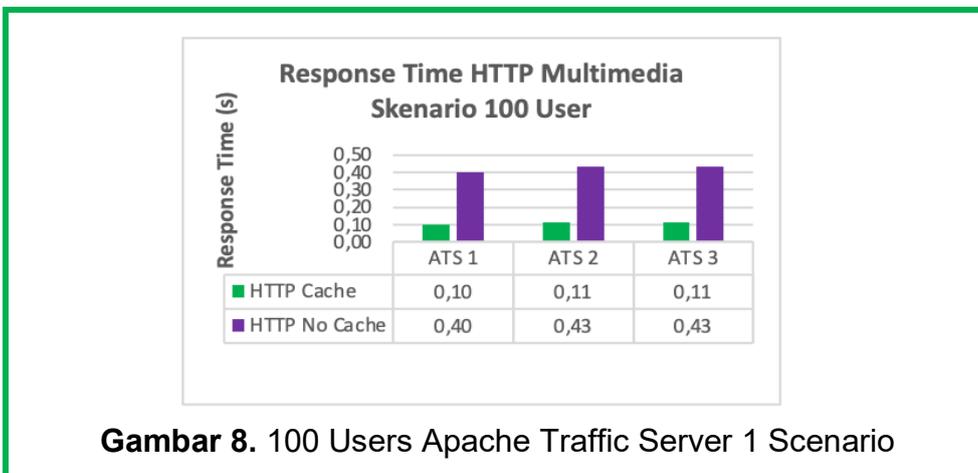
Fig 3 dan 4 merepresentasikan grafik perbandingan nilai *throughput* pada pengujian *cache* dan *no cache* skenario jumlah *user* dengan mengakses link *web server* HTTP. Berdasarkan grafik pada gambar di atas, *throughput* meningkat secara signifikan pada pengujian *cache* karena sebagian besar konten dapat disimpan dan diakses secara lokal tanpa harus memuat ulang dari server awal. Proses ini mengurangi RTT (*Round Trip Time*) dan meningkatkan efisiensi. Cache juga dapat mengurangi beban pada sistem penyimpanan utama dan mengoptimalkan penggunaan *bandwidth*, yang berkontribusi pada peningkatan performa keseluruhan sistem.

Fig 5 dan 6 merepresentasikan grafik perbandingan nilai *throughput* pada pengujian *cache* dan *no cache* skenario jumlah *user* dengan mengakses link *web server* HTTPS. Dapat dilihat pada kedua gambar tersebut bahwa nilai *throughput* *cache* lebih tinggi dibandingkan *no cache*, Namun nilai *throughput* pada protokol HTTPS cenderung lebih kecil dibandingkan dengan protokol HTTP. HTTPS menggunakan protokol (SSL/TLS) untuk mengenkripsi data yang dikirimkan antara klien dan *server*. Data yang diterima harus didekripsi sebelum digunakan sehingga memerlukan waktu yang sedikit lebih lama. Sebelum data dapat dikirim melalui HTTPS, klien dan server harus melakukan *handshake* SSL/TLS untuk menetapkan koneksi yang aman. Proses ini melibatkan beberapa langkah seperti pertukaran kunci dan sertifikat, yang menambah latensi awal pada koneksi. Namun, HTTPS seringkali dianggap lebih penting dibandingkan dengan penurunan *throughput*,

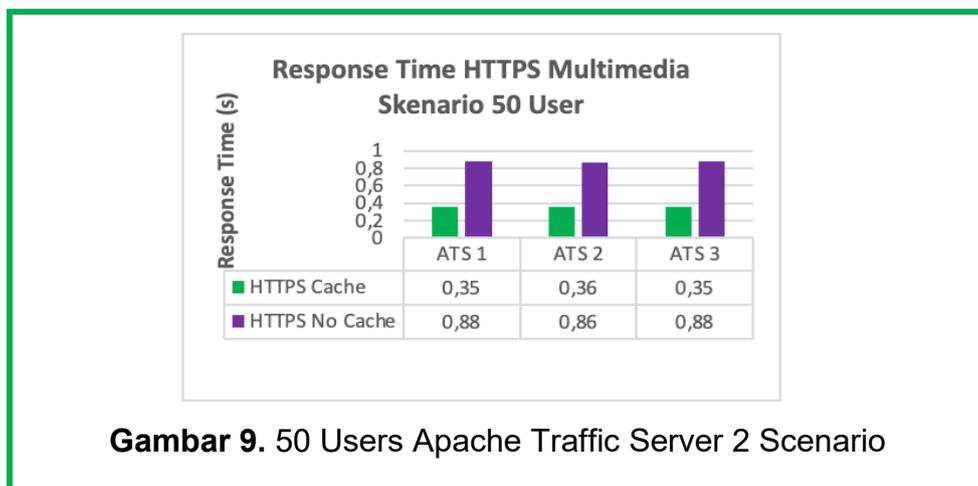
terutama untuk *website* atau aplikasi yang membutuhkan privasi dan integritas data. Semua gambar di atas menunjukkan peningkatan nilai throughput dengan menggunakan sistem caching. Hal ini disebabkan oleh caching, yang secara signifikan meningkatkan throughput dengan mengurangi waktu yang dibutuhkan untuk mengambil data. Sebaliknya, tanpa caching, setiap permintaan harus diproses oleh sistem backend, yang biasanya lebih lambat daripada mengakses data dari cache. Tanpa cache, hasilnya adalah lebih sedikit permintaan yang diproses per detik.



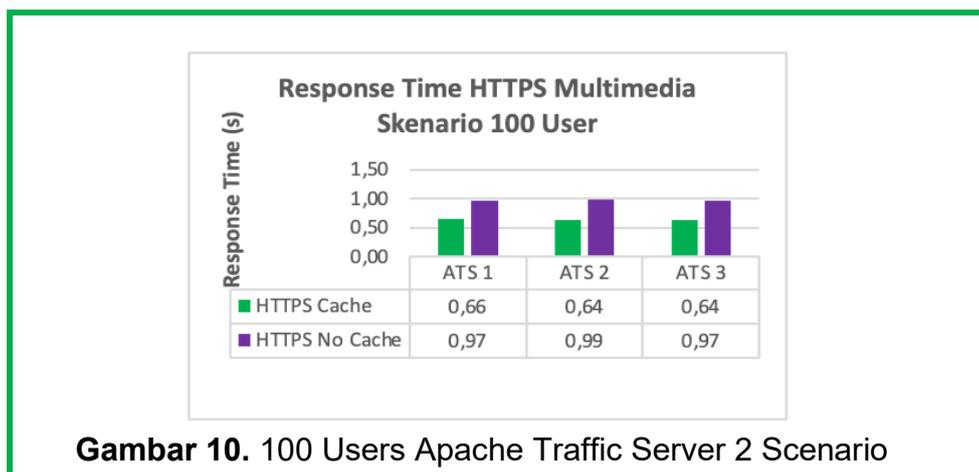
Gambar 7. 50 Users Apache Traffic Server 1 Scenario



Gambar 8. 100 Users Apache Traffic Server 1 Scenario



Gambar 9. 50 Users Apache Traffic Server 2 Scenario



Gambar 10. 100 Users Apache Traffic Server 2 Scenario

Response Time

Response time, atau waktu respons, adalah waktu yang dibutuhkan oleh sistem untuk merespons permintaan dari pengguna setelah permintaan tersebut diajukan. Dalam konteks edge caching, response time sangat berpengaruh terhadap performa dan efektivitas sistem. Ketika data disajikan dari cache, pengambilan data secara signifikan lebih cepat dibandingkan dengan mengambil data dari penyimpanan utama atau basis data, hal ini akan menghasilkan waktu respons yang lebih singkat.

Fig 9 dan 10 merepresentasikan grafik perbandingan nilai *response time* pada pengujian *cache* dan *no cache* skenario jumlah *user* dengan mengakses link *website* HTTP. *Response time* dalam konteks pengujian *Siege Load Testing* adalah waktu yang diperlukan untuk *server* merespons permintaan dari klien. *Response time* memperjelas tentang seberapa cepat *website* dapat melayani permintaan pengguna di bawah beban tertentu. *Cache* bekerja dengan menyimpan konten yang sering diakses di lokasi yang lebih cepat di akses sehingga dapat mengurangi kebutuhan untuk mengakses *website* yang lebih lambat, dan mengurangi waktu yang dibutuhkan untuk mendapatkan konten yang di request. Nilai *response time* menggunakan *cache* lebih rendah atau lebih cepat dibandingkan *no cache* karena *cache* dapat mengurangi waktu yang diperlukan untuk memuat ulang konten dari server sehingga saat ingin mengakses *website* yang sama, konten telah tersimpan di *cache* dan dapat diambil dengan cepat. Peningkatan jumlah pengguna yang mengakses server secara bersamaan dapat menyebabkan peningkatan beban pada server dan jaringan sehingga dapat memperlambat *response time*.

Fig 9 dan 10 merepresentasikan grafik perbandingan nilai *response time* pada pengujian *cache* dan *no cache* skenario jumlah *user* dengan mengakses link *website* HTTPS. Berdasarkan grafik pada kedua gambar di atas, *response time* dengan *cache* cenderung lebih rendah dibandingkan *response time* dengan *no cache* karena *cache* memungkinkan konten yang sering diakses untuk disimpan dan diambil lebih cepat. Peningkatan jumlah pengguna yang mengakses server secara bersamaan dapat menyebabkan peningkatan beban pada server dan jaringan sehingga dapat memperlambat *response time*. Secara keseluruhan, response time yang rendah dalam sistem edge caching sangat penting untuk memastikan pengiriman konten yang cepat dan efisien, yang pada gilirannya meningkatkan pengalaman pengguna, efisiensi jaringan, dan performa sistem secara keseluruhan.

3.2 Pembahasan

Sistem caching terbukti mampu meningkatkan throughput secara signifikan dengan mengurangi waktu pengambilan data dari server. Dalam implementasinya, caching menyimpan data yang sering diakses di lokasi yang lebih dekat dengan pengguna atau aplikasi, seperti memori lokal atau proxy cache. Pendekatan ini mengeliminasi kebutuhan untuk mengakses server backend untuk setiap permintaan, sehingga waktu respon menjadi lebih cepat. Sebagai contoh, penelitian yang dilakukan oleh Meyer dan sesama peneliti (2017) menunjukkan bahwa implementasi caching pada sistem web dapat mengurangi latensi hingga 50% dan meningkatkan jumlah permintaan yang diproses per detik secara signifikan, terutama dalam aplikasi berbasis konten yang memiliki pola akses berulang. Sebaliknya, sistem tanpa caching harus mengandalkan proses pengambilan data dari server pusat setiap kali permintaan dilakukan, yang dapat meningkatkan latensi akibat keterbatasan bandwidth atau waktu pemrosesan di server backend. Hal ini tidak hanya memperlambat kinerja sistem tetapi juga menambah beban pada jaringan dan server pusat, yang pada akhirnya dapat menurunkan efisiensi operasional. Dengan caching, beban tersebut dialihkan ke tingkat lokal, memungkinkan lebih banyak permintaan diproses dalam waktu lebih singkat. Selain itu, caching memberikan dampak positif terhadap pengalaman pengguna, terutama pada aplikasi real-time seperti layanan streaming video atau platform e-commerce, di mana kecepatan akses data menjadi faktor kritis (Barroso et al., 2019).

Response time yang rendah dalam sistem edge caching memainkan peran penting dalam memastikan pengiriman konten yang cepat dan efisien kepada pengguna akhir. Dengan menggunakan cache di lokasi edge, sistem dapat menyimpan salinan data yang sering diakses lebih dekat ke pengguna. Hal ini memungkinkan pengurangan latensi secara signifikan, karena permintaan data tidak perlu selalu diarahkan ke server pusat. Selain itu, caching membantu mengurangi beban kerja pada server utama dan menghemat penggunaan bandwidth jaringan, yang pada akhirnya meningkatkan efisiensi operasional secara keseluruhan (Wang et al., 2019). Dalam konteks distribusi konten seperti video streaming atau aplikasi berbasis web, kemampuan untuk menyediakan akses cepat ini sangat krusial untuk mempertahankan kepuasan pengguna. Implementasi sistem edge caching yang optimal menjadi strategi yang tidak terelakkan dalam menghadapi meningkatnya tuntutan pengguna di era digital. Dengan memanfaatkan algoritma caching yang cerdas, sistem dapat secara dinamis memprioritaskan data yang paling relevan untuk disimpan, sehingga meningkatkan pengalaman pengguna secara signifikan. Sebagai contoh, studi menunjukkan bahwa pengguna cenderung meninggalkan aplikasi atau situs web jika waktu pemuatan terlalu lama, menekankan pentingnya efisiensi sistem (Cisco, 2021). Dengan demikian, edge caching tidak hanya membantu dalam memenuhi kebutuhan pengguna modern, tetapi juga mendukung kelangsungan bisnis melalui performa sistem yang andal dan hemat biaya.

4. Kesimpulan

Penelitian ini memberikan analisis komprehensif mengenai solusi edge caching menggunakan Apache Traffic Server di bawah kondisi beban yang berbeda, yang menunjukkan dampak caching pada throughput, waktu respons, dan metrik kinerja waktu yang telah berlalu. Studi ini menyoroti peningkatan kinerja yang signifikan yang dicapai melalui edge caching dalam mengurangi latensi dan meningkatkan throughput untuk protokol HTTP dan HTTPS, serta video-on-demand.

Hasil penelitian menunjukkan bahwa caching secara drastis meningkatkan efisiensi sistem dengan meminimalkan waktu yang dibutuhkan untuk mengambil data dan mengurangi beban pada server backend, sehingga menghasilkan pengiriman data yang lebih cepat dan pengalaman pengguna yang lebih baik. Dengan membandingkan skenario dengan dan tanpa caching, penelitian ini menggarisbawahi pentingnya caching dalam mengoptimalkan kinerja web, terutama di lingkungan dengan lalu lintas tinggi. Temuan ini menunjukkan bahwa penerapan edge caching dapat menghasilkan peningkatan substansial dalam hal kinerja, skalabilitas, dan efisiensi biaya untuk jaringan pengiriman konten dan penyedia layanan internet.

Daftar Pustaka

- A. Bachtiar, "Konfigurasi Jaringan Untuk Mekanisme Edge Caching Berbasis IP Menggunakan Apache Trafik Server," 2023.
- A. Y. Chandra, "Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request," *J. Sist. dan Inform.*, vol. 14, no. 1, pp. 48–56, 2019, doi: 10.30864/jsi.v14i1.248.
- Aksenta, A., Irmawati, I., Ridwan, A., Hayati, N., Sepriano, S., Herlinah, H., ... & Ginting, T. W. (2023). *LITERASI DIGITAL: Pengetahuan & Transformasi Terkini Teknologi Digital Era Industri 4.0 dan Society 5.0*. PT. Sonpedia Publishing Indonesia.
- Al-Atsari, H. A., & Suharjo, I. (2023). Integrasi Server On-Premise dengan Server Cloud Menggunakan Cloud VPN dan Mikrotik Ipsec Untuk Peningkatan Keamanan Koneksi. *Jurnal Syntax Admiration*, 4(11), 1977-1996.
- Anjani, F. Y. (2018). Perencanaan Peningkatan Kualitas Produk Data Internet di Telkom Pasuruan Untuk Menciptakan Kepuasan Pelanggan Menggunakan Metodologi Quality Function Deployment (Disertasi Doktorat, Institut Teknologi Sepuluh Nopember).
- Arifin, Z., Ariantini, M. S., Sudipa, I. G. I., Chaniago, R., Dwipayana, A. D., Adhichandra, I., ... & Alfiah, T. (2023). *GREEN TECHNOLOGY: Penerapan Teknologi Ramah Lingkungan Berbagai Bidang*. PT. Sonpedia Publishing Indonesia.
- Barroso, L. A., Clidas, J., & Hölzle, U. (2019). *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines* (3rd ed.). Morgan & Claypool Publishers.
- Cisco: 2020 CISO Benchmark Report, *Comput. Fraud Secur.*, vol. 2020, no. 3, pp. 4–4, 2020, doi: 10.1016/s1361-3723(20)30026-9.
- Cisco. (2021). *Cisco Annual Internet Report (2018–2023) White Paper*. Diakses dari <https://www.cisco.com>
- D. Kunda, S. Chihana, S. Muwanei, and M. Sinyinda, "Web Server Performance of Apache and Nginx: A Systematic Literature Review," *Comput. Eng. Intell. Syst.*, vol. 8, no. 2, pp. 43–52, 2017, [Online]. Available: www.iiste.orgK. Mokhtarian and H. A. Jacobsen, "Flexible Caching Algorithms for Video

- Content Distribution Networks,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1062–1075, 2017, doi: 10.1109/TNET.2016.2621067.
- International Telecommunications Union, “IMT Traffic Estimates for the Years 2020 to 2030,” *Electron. Publ. Geneva*, vol. 0, pp. 1–51, 2015, [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2370-2015-PDF-E.pdf
- M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, 2014, doi: 10.1109/TIT.2014.2306938.
- Meyer, D., Carpenter, B., & Caching Research Group. (2017). *Caching for Performance Optimization in Web Applications*. ACM Digital Library.
- O. Staff, “Open Source Traffic Server,” Nov 02. [Online]. Available: <https://ostatic.com/blog/guest-post-yahoos-cloud-team-open-sources-traffic-server>
- Pranoto, H., & Zarlis, M. (2018). Tingkat Validitas Data Pada Sistem Monitoring Trafo Berbasis Internet Of Things (IoT) Pada Smart City. In *Prosiding Seminar SeNTIK* (Vol. 2, No. 1, pp. 138-145).
- S. Xu, X. Liu, S. Guo, X. Qiu, and L. Meng, “MECC: A Mobile Edge Collaborative Caching Framework Empowered by Deep Reinforcement Learning,” *IEEE Netw.*, vol. 35, no. 4, pp. 176–183, 2021, doi: 10.1109/MNET.011.2000663.
- Saroji, A., Harmini, T., & Taqiyuddin, M. (2021). Sejarah evolusi generasi internet. *Lani: Jurnal Kajian Ilmu Sejarah dan Budaya*, 2(2), 65-75.
- Suartana, I. M., & Prapanca, A. (2020, November). Analisis Perancangan Multimedia Streaming Berbasis Software Defined Network. In *Prosiding Seminar Nasional Informatika Bela Negara* (Vol. 1, pp. 76-81).
- T. X. Vu, S. Chatzinotas, and B. Ottersten, “Edge-Caching Wireless Networks: Performance Analysis and Optimization,” *IEEE Trans. Wirel. Commun.*, vol. 17, no. 4, pp. 2827–2839, 2018, doi: 10.1109/TWC.2018.2803816.
- Wang, Y., Zhang, L., & Chen, H. (2019). *Edge caching strategies for efficient content delivery*. *Journal of Network and Computer Applications*, 102, 113-123. doi:10.1016/j.jnca.2019.01.002