Implementasi Mekanisme IP Edge Caching menggunakan Apache Traffic Server pada Web Server Website Multimedia dan Website Video on Demand

1st Muhammad Raihan Baswara
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
rbaswara@student.telkomuniversity.ac.id

2nd Sofia Naning Hertiana
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sofiananing@telkomuniversity.ac.id

3rd Leanna Vidya Yovita
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
leanna@telkomuniversity.ac.id

***Abstrak* — Seiring dengan meningkatnya jumlah pengguna internet, klien memiliki permintaan yang semakin tinggi untuk layanan baru dengan delay yang rendah, konsumsi daya yang kecil, dan performa yang tinggi, sehingga sangat mempengaruhi arsitektur jaringan komunikasi yang ada. Oleh karena itu, masalah yang sering dijumpai pada website adalah performa yang kurang optimal, yang meliputi waktu respon yang lambat, beban server yang berlebihan, skalabilitas yang terbatas, dan ketidakmampuan untuk menangani trafik yang tinggi. Pada Proxy Server, caching dilakukan di edge dengan menggunakan Apache Traffic Server (ATS) sebagai metode yang akan memberikan waktu respon yang lebih cepat, karena request dari pengguna tidak perlu memakan banyak waktu untuk terhubung ke server asli, dan ATS bekerja dengan menerima request dari client. Dengan solusi caching menggunakan metode ATS, penulis melakukan uji coba dengan menggunakan pengujian pada 2 Web Server, yaitu Website Multimedia dan Website Video on Demand yang berjalan pada protokol HTTPS. Pengujian dilakukan dengan parameter seperti Hit Ratio, Throughput, Delay, dan Round-Trip Time (RTT). Terdapat perbedaan yang sangat signifikan antara menggunakan ATS (Cache) untuk hasil Throughput, Delay, dan RTT yang lebih baik dibandingkan tanpa menggunakan ATS (tanpa cache).**

Kata kunci*— *Proxy Server, Apache Traffic Server, Edge Caching, Web Server, Website Multimedia, Website Video on Demand

I. PENDAHULUAN

Semakin pesatnya pertumbuhan trafik data di internet, volume dan variasi konten pun terus bertambah, dan Cisco (2020) mencatat bahwa trafik IP meningkat sangat pesat menjelang tahun 2023, dan diprediksi akan ada lebih banyak trafik IP di tahun 2023, di mana 66 persen populasi global akan menjadi pengguna internet. Dengan meningkatnya penggunaan internet, jumlah konten video dan multimedia pun berkembang pesat. Namun, jumlah halaman situs web yang diakses menurun seiring dengan waktu respons karena penundaan 1 detik menyebabkan penurunan 7% [1]. Cisco Visual Network Index memperkirakan bahwa trafik konten multimedia saja akan mencapai 82% dari penggunaan internet global pada tahun 2023 [1]. Trafik internet juga tumbuh secara signifikan dalam dua dekade terakhir karena meningkatnya jumlah pengguna internet yang menuntut layanan berkualitas tinggi, meluasnya penggunaan media sosial dengan berbagai pilihan konten, dan perkembangan situs web yang canggih dan interaktif. Situs web telah menjadi salah satu elemen kunci dalam ekosistem internet, menyediakan platform bagi individu dan bisnis untuk berbagi informasi, berinteraksi, dan mengakses berbagai layanan secara online. Karena pertumbuhan jaringan yang cepat, pengguna memiliki permintaan yang lebih tinggi untuk layanan baru dengan delay yang rendah, konsumsi daya yang kecil, dan performa yang tinggi, sehingga arsitektur jaringan komunikasi yang ada sangat terpengaruh dan dibatasi dan pekerjaan pemeliharaan bidang jaringan komunikasi menghadapi persyaratan yang semakin ketat.

II. KAJIAN TEORI

Sehubungan dengan kemunculan variasi konten pada sebuah website akan terus meningkat hingga 82% dari pengguna global pada tahun 2023, maka diperlukan sebuah sistem agar web server dapat menangani beban trafik yang tinggi. Pada topik ini, implementasi yang dilakukan adalah dengan menggunakan IP Edge Caching menggunakan metode Apache Traffic Server pada Web Server Website Multimedia dan Website Video on Demand.

A. Web Server

Web server digunakan untuk menyediakan layanan web, seperti webmail, web chat, atau layanan web lainnya yang dapat diakses melalui protokol HTTP. Web server juga bertugas untuk menerima dan mengolah permintaan HTTP dari klien, web server akan mengambil data yang diminta dan mengirimkannya kembali ke pengguna melalui protokol HTTP. Web server access log berisi kumpulan file data yang berasal aktivitas server yang dipakai [2].

B. Website Multimedia

Website multimedia dijalankan pada platform InfinityFree yang berfungsi sebagai penyedia hosting gratis sehingga memungkinkan penulis memiliki domain dan ruang penyimpanan yang dapat diakses secara gratis. Website multimedia dibuat menggunakan HTML dan CSS sehingga dapat membuat halaman web yang interaktif, menarik, dan sesuai dengan kebutuhan dan keinginan.

C. Website Video on Demand

Sistem Video on Demand (VoD) memungkinkan pengguna untuk memilih dan menyaksikan video yang hendak diakses dalam jaringan sebagai bagian dari sistem interaktif. Video on demand (VoD) dapat memanfaatkan proses streaming dan download. Video on demand (VoD) juga memungkinkan pengguna untuk melakukan kendali, seperti pause, fast forward, fast rewind, slow forward, dan slow rewind. Website VoD dijalankan pada platform Microsoft Azure. Microsoft Azure, sering disebut sebagai Azure, adalah platform cloud computing yang dijalankan oleh Microsoft. Virtual Machine di Azure digunakan sebagai server untuk Website Video on Demand.

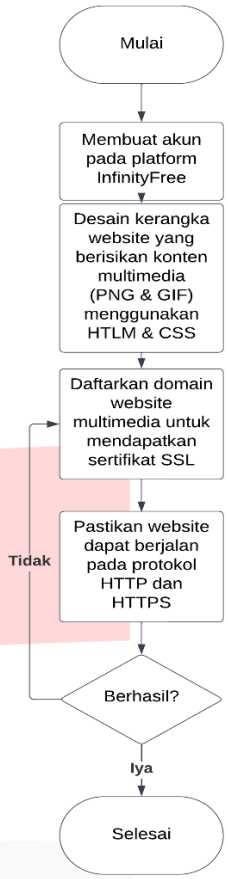
D. Protokol HTTP/HTTPS

Hypertext Transfer Protocol (HTTP) adalah suatu protokol permintaan yang berjalan di atas TCP. Standar HTTP formal pertama adalah HTTP/1.1 yang dirilis pada Januari 1997 sebagai RFC 2086 [3]. HTTPS adalah protokol HTTP berbasis Secure Sockets Layer (SSL) untuk pertukaran informasi, yang bertujuan untuk membangun channels yang aman (secure) di suatu jaringan yang tidak aman [4]. Hal tersebut dikarenakan user mulai lebih memperhatikan keamanan layanan dan jaringan internet membuat banyak situs web mengadopsi Protokol HTTPS untuk melindungi data privasi pengguna.

III. METODE

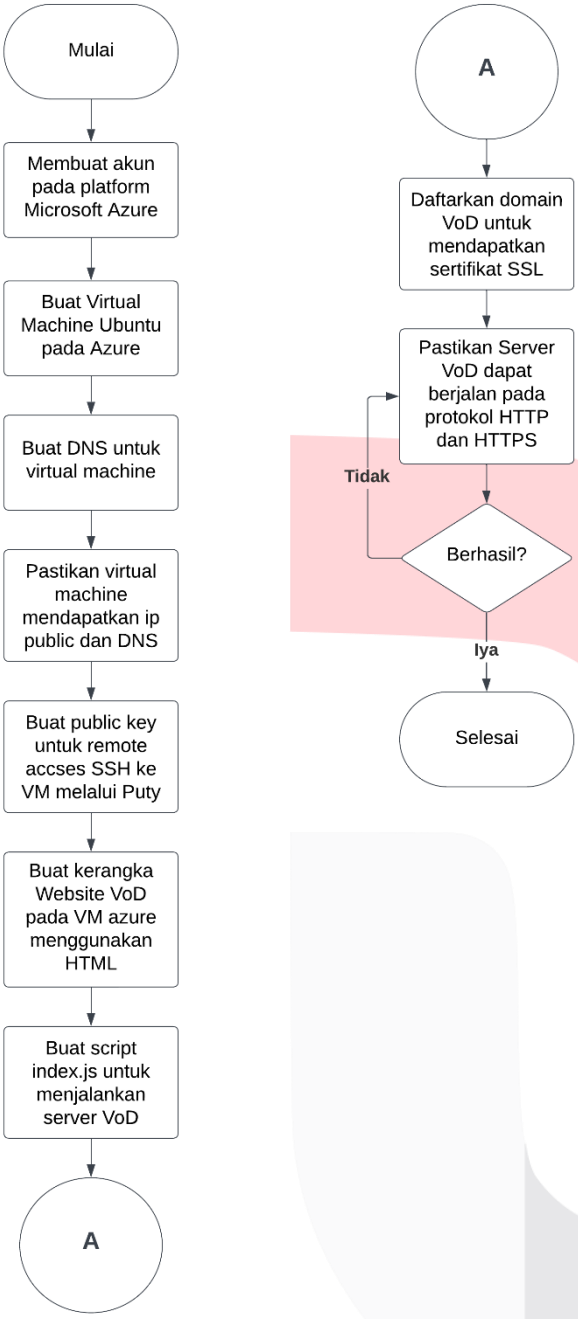
Topik ini menggunakan Packet Network Emulator Tool Lab (PNETLab) sebagai implementasi Sistem Topologi IP Edge Caching menggunakan metode Apache Traffic Server dengan skenario pengujian menggunakan berbagai macam ukuran halaman web, ukuran resolusi video, dan jumlah user serta menggunakan fitur load testing tools siege dan inspect network traffic untuk pengolahan data.

A. Desain Sistem Web Server



GAMBAR 1:
Flowchart Alur Pembuatan Website Multimedia

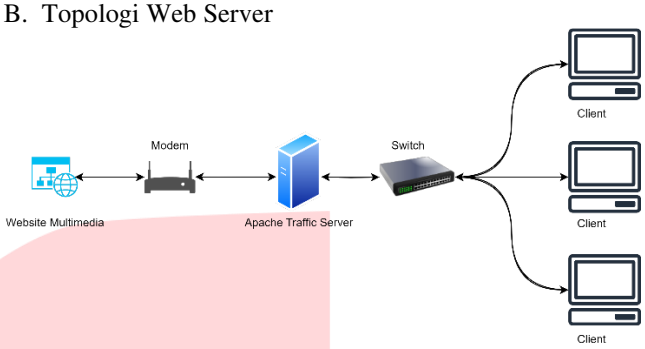
Pada Gambar 1 diatas menggambarkan langkah-langkah untuk membuat dan meng-hosting website multimedia menggunakan platform InfinityFree. Kita perlu membuat akun di platform InfinityFree, yang merupakan sebuah layanan hosting gratis yang menyediakan infrastruktur untuk meng-hosting website. Setelah itu, merancang kerangka website dengan menggunakan bahasa pemrograman HTML dan CSS. Dalam proses desain ini, konten multimedia seperti gambar PNG dan animasi GIF dimasukkan ke dalam website multimedia. Selanjutnya mendaftarkan domain untuk website multimedia agar dapat diakses melalui URL yang lebih mudah diingat. Selain itu, untuk meningkatkan keamanan, perlu dilakukan pendaftaran sertifikat SSL agar website bisa berjalan pada protokol HTTPS.



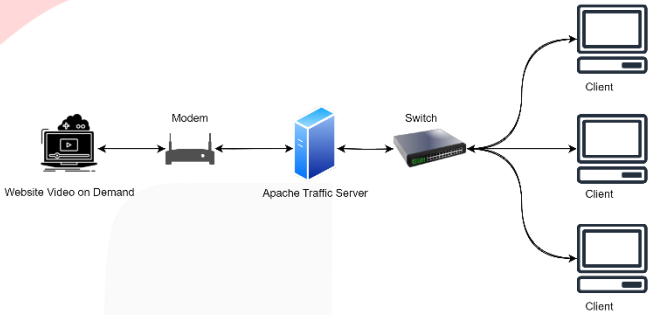
GAMBAR 2:
Flowchart Alur Pembuatan Website VoD

Pada Gambar 2 diatas menggambarkan langkah-langkah untuk membuat dan meng-hosting sebuah Website Video on Demand (VoD) pada platform Microsoft Azure. Azure adalah sebuah platform cloud computing yang menyediakan berbagai layanan, termasuk Virtual Machine (VM), untuk meng-hosting aplikasi dan website di lingkungan cloud. Selanjutnya membuat Virtual Machine dengan sistem operasi Ubuntu di dalam platform Azure, lalu membuat DNS untuk Virtual Machine agar website VoD dapat diakses melalui alamat domain yang lebih mudah diingat. Pastikan Virtual Machine sudah memiliki IP Publik dan DNS agar dapat diakses melalui internet. Selanjutnya, buatlah Public Key untuk remote access ke Virtual Machine menggunakan protokol SSH melalui Putty. SSH adalah protokol yang digunakan untuk mengakses, mengelola, dan berkomunikasi dengan server secara aman melalui koneksi terenkripsi. Setelah mengatur koneksi dan akses ke Virtual Machine, berikutnya adalah membuat kerangka Website VoD menggunakan bahasa pemrograman HTML, lalu buat script index.js untuk menjalankan server VoD yang akan digunakan untuk meng-hosting dan menyajikan video on demand kepada pengguna.

B. Topologi Web Server



GAMBAR 3:
Topologi IP Edge Caching Website Multimedia



GAMBAR 4:
Topologi IP Edge Caching Website VoD

Pada Gambar 3 dan Gambar 4 diatas merupakan Topologi jaringan yang menggambarkan alur data dari Client ke Website Multimedia dan Website Video on Demand (VoD) melalui beberapa perangkat jaringan. Alur dimulai dari Client, yaitu perangkat pengguna seperti komputer atau smartphone yang ingin mengakses situs web. Data dari Client kemudian melewati sebuah Switch, yang bertindak sebagai perangkat penghubung dalam jaringan lokal. Setelah melewati Switch, data dikirim ke Apache Traffic Server, yang berfungsi sebagai sebuah reverse proxy server. Apache Traffic Server berperan sebagai perantara antara Client dan Website Multimedia serta Website VoD. Reverse proxy server ini membantu mempercepat akses dan mengoptimalkan distribusi konten dengan menyimpan cache dan memproses permintaan dari Client sebelum diteruskan ke server tujuan. Selanjutnya, data melewati Modem, yang merupakan perangkat yang menghubungkan jaringan lokal dengan jaringan internet. Setelah melewati Modem, data menuju Website Multimedia dan Website VoD, yang merupakan server-server yang menyediakan konten yang diminta oleh Client. Website VoD berisi konten video on demand yang dapat diakses oleh pengguna sesuai permintaan, sedangkan Website Multimedia menyediakan berbagai jenis konten multimedia seperti gambar dan animasi gif. Secara keseluruhan, topologi ini menunjukkan jalur data dari Client melalui Switch, Apache Traffic Server, Modem, hingga mencapai Website VoD dan Website Multimedia.

C. Parameter Pengujian

Penulis melakukan uji coba menggunakan pengujian pada website multimedia dan website video on demand dengan beberapa skenario yang berjalan pada Protokol HTTPS. Pengujian yang dilakukan menggunakan parameter seperti Throughput dan Delay.

1. Throughput

Throughput merupakan jumlah total kedatangan paket (data) sukses menuju tujuan selama interval waktu tertentu yang merupakan kecepatan transfer data yang diukur dalam satuan MegaByte per Second (MBps). Pengukuran throughput menggunakan load testing siege dan inspect elemen network untuk melihat dan menghitung data throughput.

$$Throughput = \frac{jumlah\ data\ yang\ dikirim}{waktu\ pengiriman\ data} \quad (1)$$

Pada persamaan (1) throughput dapat dihitung dengan membagi jumlah data yang dikirim dengan waktu pengiriman data.

2. Delay

Delay merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal hingga dapat diterima oleh penerima. Pengukuran delay menggunakan inspect network pada browser klien untuk melihat dan menghitung data delay.

$$Rata - rata\ delay = \frac{Total\ Delay}{Jumlah\ paket\ yang\ diterima} \quad (2)$$

Pada persamaan (2) delay dapat dihitung dengan membagi total delay yang didapat dengan jumlah paket yang diterima.

D. Skenario Pengujian

Tabel 1:
Skenario Ukuran Halaman Website Multimedia

Skenario Ukuran Halaman Web (HTTPS)	
No-Cache	**Cache**
2.4 MB	2.4 MB
4.8 MB	4.8 MB
7.2 MB	7.2 MB

Tabel 2:
Skenario Jumlah User Website VoD

Skenario Jumlah User (HTTPS)	
No-Cache	**Cache**
2 User	2 User
4 User	4 User
6 User	6 User

Tabel 3:
Skenario Ukuran Resolusi Video Website VoD

Skenario Resolusi Video (HTTPS)	
No-Cache	**Cache**
480p	480p
720p	720p
1080p	1080p

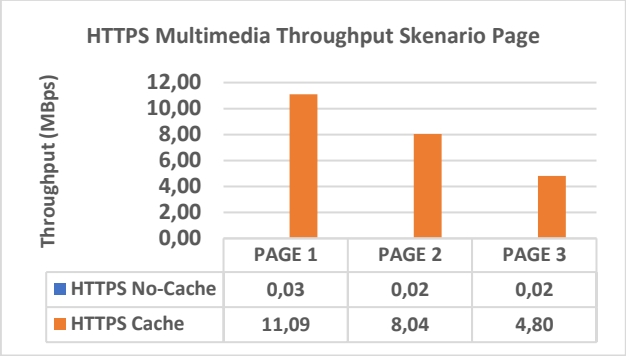
IV. HASIL DAN PEMBAHASAN

Implementasi Mekanisme Edge Caching Berbasis IP Menggunakan Apache Traffic Server untuk Web Server Website Multimedia dan Website Video on Demand diukur dengan menggunakan karakteristik Quality of Service (QoS) seperti Throughput, Round-Trip Time (RTT), dan Delay untuk parameter pengujian jaringan. Hit Cache Ratio juga digunakan dalam penelitian ini untuk parameter pengujian server.

A. Hasil Pengujian Website Multimedia

1. Skenario Ukuran Halaman Page

a. Throughput

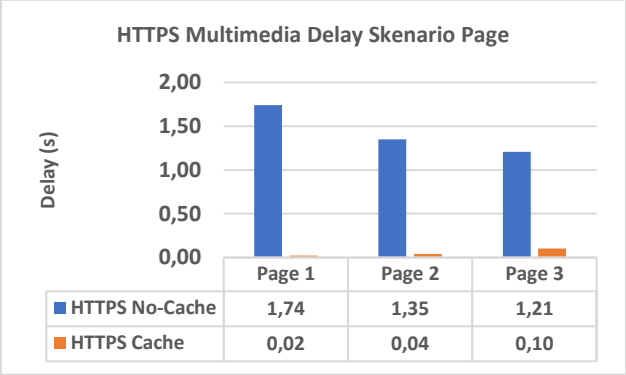


GAMBAR 5:
HTTPS Multimedia Throughput Skenario Page

Pada Gambar 5 diatas, hasil pengujian menunjukkan skenario pengujian halaman ukuran web dengan parameter Throughput (MBps) pada protokol HTTPS. Terdapat dua kondisi pengujian, yaitu HTTPS No-Cache dan HTTPS Cache, dengan tiga halaman web yang diuji, yaitu Page 1, Page 2, dan Page 3. Pada kondisi HTTPS No-Cache, rata-rata waktu untuk memuat Page 1 adalah sekitar 0.03 detik, sedangkan untuk Page 2 dan Page 3 masing-masing adalah sekitar 0.02 detik. Hal ini menunjukkan bahwa ketika tidak ada data yang di-cache, waktu yang dibutuhkan untuk memuat halaman web relatif cepat dan konsisten di semua halaman yang diuji.

Pada kondisi HTTPS Cache, rata-rata waktu untuk memuat Page 1 meningkat secara signifikan menjadi sekitar 11.09 detik. Namun, waktu untuk memuat Page 2 dan Page 3 mengalami peningkatan yang lebih rendah, yaitu sekitar 8.04 detik untuk Page 2 dan 4.80 detik untuk Page 3. Hal ini menunjukkan bahwa pada kondisi ini, waktu untuk memuat halaman web lebih lama karena data di-cache terlebih dahulu sebelum ditampilkan, sehingga waktu yang diperlukan menjadi lebih bervariasi. Secara keseluruhan, hasil pengujian menunjukkan bahwa HTTPS Cache dapat mengoptimalkan waktu muat halaman web dalam beberapa situasi dengan memanfaatkan caching data. Namun, hasilnya dapat bervariasi tergantung pada ukuran dan kompleksitas halaman web yang diakses serta ketersediaan data di-cache.

b. Delay



GAMBAR 6:
HTTPS Multimedia Delay Skenario Page

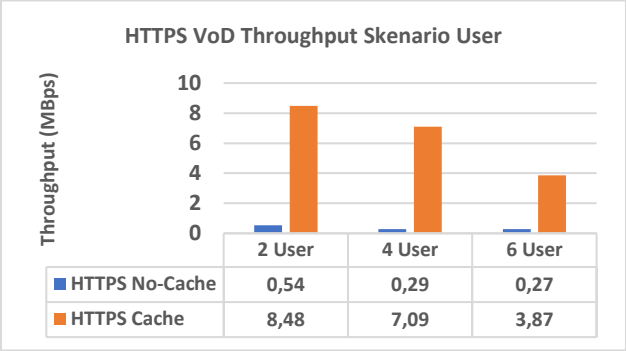
Pada Gambar 6 diatas, hasil pengujian menunjukkan skenario pengujian halaman ukuran web dengan parameter Delay (s) pada protokol HTTPS. Pengujian dilakukan dalam dua kondisi, yaitu HTTPS No-Cache dan HTTPS Cache, dengan tiga halaman web yang diuji, yaitu Page 1, Page 2, dan Page 3. Pada kondisi HTTPS No-Cache, rata-rata waktu delay (s) untuk memuat Page 1 adalah sekitar 1.74 detik, sedangkan untuk Page 2 dan Page 3 masing-masing adalah sekitar 1.35 detik dan 1.21 detik. Hal ini menunjukkan bahwa ketika tidak ada data yang di-cache, waktu delay yang dibutuhkan untuk memuat halaman web relatif tinggi dan konsisten di semua halaman yang diuji.

Di sisi lain, pada kondisi HTTPS Cache, rata-rata waktu delay (s) untuk memuat Page 1 menurun secara signifikan menjadi sekitar 0.02 detik. Namun, waktu delay untuk memuat Page 2 dan Page 3 mengalami kenaikan menjadi sekitar 0.04 detik dan 0.10 detik. Hal ini menunjukkan bahwa pada kondisi ini, penggunaan cache dapat mengurangi waktu delay untuk memuat halaman web secara signifikan. Secara keseluruhan, hasil pengujian menunjukkan bahwa HTTPS Cache dapat membantu mengurangi waktu delay dalam memuat halaman web. Ada beberapa faktor yang mempengaruhi efisiensi cache, seperti ukuran halaman web, kompleksitas, dan kesesuaian data yang di-cache, yang dapat menyebabkan peningkatan waktu delay pada halaman lain.

B. Hasil Pengujian Website Video on Demand

1. Skenario Jumlah User

a. Throughput

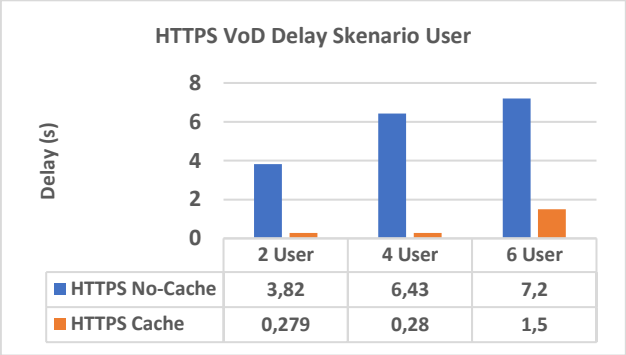


GAMBAR 7:
HTTPS VoD Throughput Skenario User

Pada Gambar 7 diatas, hasil pengujian menunjukkan skenario pengujian jumlah user dengan parameter Throughput (MBps) pada protokol HTTPS. Pengujian dilakukan dalam dua kondisi, yaitu HTTPS No-Cache dan HTTPS Cache, dengan tiga skenario jumlah user yang diuji, yaitu 2 User, 4 User, dan 6 User. Pada kondisi HTTPS No-Cache, throughput yang didapat untuk 2 User adalah 0.54 MBps, sementara untuk 4 User dan 6 User masing-masing adalah 0.29 MBps dan 0.27 MBps. Hasil ini menunjukkan bahwa ketika tidak ada data yang di-cache, throughput yang dapat dicapai oleh setiap user menurun ketika jumlah user yang mengakses server semakin meningkat.

Di sisi lain, pada kondisi HTTPS Cache, hasil throughput untuk 2 User meningkat secara signifikan menjadi 8.48 MBps. Lalu, throughput untuk 4 User dan 6 User mengalami penurunan menjadi sekitar 7.09 MBps dan 3.87 MBps. Hal ini menunjukkan bahwa pada kondisi ini, penggunaan cache dapat meningkatkan throughput untuk jumlah user yang lebih sedikit, tetapi throughput cenderung menurun ketika jumlah user semakin banyak. Penurunan throughput ini terjadi karena adanya persaingan akses ke cache dan pembagian sumber daya yang lebih terbatas ketika ada lebih banyak user yang mengakses server. Secara keseluruhan, hasil pengujian menunjukkan bahwa HTTPS Cache dapat meningkatkan throughput ketika jumlah user yang mengakses server terbatas. Namun, jika jumlah user semakin banyak, throughput yang dicapai oleh setiap user cenderung menurun.

b. Delay



GAMBAR 8:
HTTPS VoD Delay Skenario User

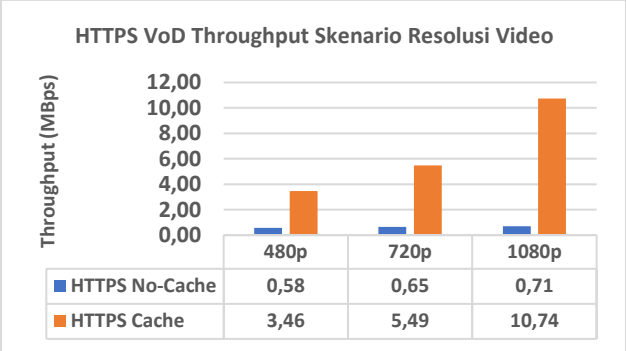
Pada Gambar 8 diatas, hasil pengujian menunjukkan skenario pengujian jumlah user dengan parameter Delay (s) pada protokol HTTPS. Pengujian dilakukan dalam dua kondisi, yaitu HTTPS No-Cache dan HTTPS Cache, dengan tiga tingkat jumlah user yang diuji, yaitu 2 User, 4 User, dan 6 User. Pada kondisi HTTPS No-Cache, rata-rata waktu delay (s) untuk 2 User adalah 3.82 detik, sementara untuk 4 User dan 6 User masing-masing adalah 6.43 detik dan 7.2 detik. Hasil ini menunjukkan bahwa ketika tidak ada data yang di-cache, waktu delay yang dibutuhkan untuk memuat halaman web meningkat secara signifikan ketika jumlah user yang mengakses server semakin banyak.

Di sisi lain, pada kondisi HTTPS Cache, rata-rata waktu delay (s) untuk 2 User sangat rendah, yaitu 0.279 detik. Namun, waktu delay untuk 4 User dan 6 User hampir tidak berubah, yaitu sekitar 0.28 detik dan 1.5 detik. Hal ini menunjukkan bahwa pada kondisi ini, penggunaan cache dapat secara signifikan mengurangi waktu delay untuk memuat halaman web, terutama pada jumlah user yang sedikit. Namun, ketika jumlah user semakin banyak, waktu

delay cenderung stabil atau mengalami sedikit peningkatan, salah satunya dikarenakan adanya pembagian sumber daya yang lebih terbatas ketika ada lebih banyak user yang mengakses server. Secara keseluruhan, hasil pengujian menunjukkan bahwa HTTPS Cache dapat membantu mengurangi waktu delay dalam memuat halaman web, terutama pada jumlah user yang sedikit. Namun, ketika jumlah user semakin banyak, waktu delay mungkin akan meningkat atau stabil karena pembagian sumber daya yang lebih terbatas.

2. Skenario Resolusi Video

a. Throughput

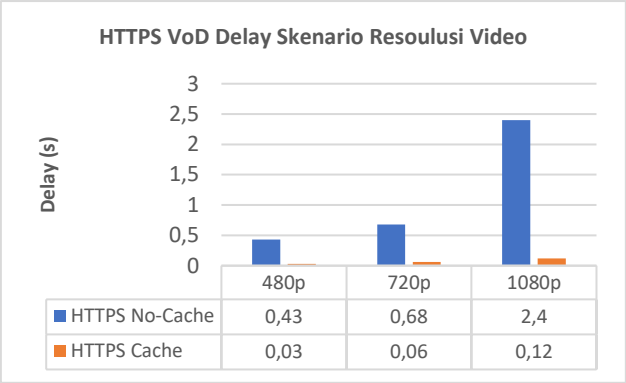


GAMBAR 9:
HTTPS VoD Throughput Skenario Resolusi Video

Pada Gambar 9 diatas, hasil pengujian menunjukkan skenario pengujian resolusi video dengan parameter Throughput (MBps) pada protokol HTTPS. Pengujian dilakukan dalam dua kondisi, yaitu HTTPS No-Cache dan HTTPS Cache, dengan tiga tingkat resolusi video yang diuji, yaitu 480p, 720p, dan 1080p. Pada kondisi HTTPS No-Cache, rata-rata throughput untuk video dengan resolusi 480p adalah 0.58 MBps, sementara untuk video 720p dan 1080p masing-masing adalah 0.65 dan 0.71 MBps. Hasil ini menunjukkan bahwa ketika tidak ada data yang di-cache, throughput yang dibutuhkan untuk memuat video dengan resolusi lebih tinggi seperti 720p dan 1080p cenderung lebih tinggi daripada resolusi lebih rendah seperti 480p.

Di sisi lain, pada kondisi HTTPS Cache, rata-rata throughput untuk video 480p meningkat secara signifikan menjadi 3.46 MBps. Selain itu, throughput untuk video 720p dan 1080p juga meningkat menjadi 5.49 MBps dan 10.74 MBps. Hal ini menunjukkan bahwa pada kondisi ini, penggunaan cache dapat mengoptimalkan throughput untuk semua tingkatan resolusi video. Dengan adanya cache, server dapat menyimpan data video yang telah diunduh sebelumnya, sehingga mengurangi waktu yang dibutuhkan untuk mengunduh data saat video dengan resolusi yang sama diakses kembali. Secara keseluruhan, hasil pengujian menunjukkan bahwa HTTPS Cache dapat membantu meningkatkan throughput untuk semua tingkatan resolusi video, sehingga memungkinkan pengguna untuk memperoleh pengalaman menonton yang lebih lancar dan responsif.

b. Delay



GAMBAR 10:
HTTPS VoD Delay Skenario Resolusi Video

Pada Gambar 10 diatas, hasil pengujian menunjukkan skenario pengujian resolusi video dengan parameter Delay (s) pada protokol HTTPS. Pengujian dilakukan dalam dua kondisi, yaitu HTTPS No-Cache dan HTTPS Cache, dengan tiga tingkat resolusi video yang diuji, yaitu 480p, 720p, dan 1080p. Pada kondisi HTTPS No-Cache, rata-rata waktu delay (s) untuk video dengan resolusi 480p adalah 0.43 detik, sementara untuk video 720p dan 1080p masing-masing adalah 0.68 detik dan 2.4 detik. Hasil ini menunjukkan bahwa ketika tidak ada data yang di-cache, waktu delay yang dibutuhkan untuk memuat video dengan resolusi lebih tinggi seperti 720p dan 1080p cenderung lebih tinggi daripada resolusi lebih rendah seperti 480p.

Di sisi lain, pada kondisi HTTPS Cache, rata-rata waktu delay (s) untuk video 480p menurun secara signifikan menjadi 0.03 detik. Selain itu, waktu delay untuk video 720p dan 1080p juga menurun menjadi 0.06 detik dan 0.12 detik. Hal ini menunjukkan bahwa pada kondisi ini, penggunaan cache dapat secara drastis mengurangi waktu delay untuk memuat video dengan semua tingkatan resolusi. Dengan adanya cache, server dapat menyimpan data video yang telah diunduh sebelumnya, sehingga pengguna mendapatkan akses yang lebih cepat ketika mengakses video dengan resolusi yang sama kembali. Secara keseluruhan, hasil pengujian menunjukkan bahwa HTTPS Cache sangat efektif dalam mengurangi waktu delay untuk memuat video dengan semua tingkatan resolusi. Penggunaan cache dapat membantu mempercepat respons server, memberikan pengalaman menonton yang lebih lancar, dan mengurangi waktu tunggu bagi pengguna ketika mengakses video dengan kualitas yang berbeda.

V. KESIMPULAN

Dari pengujian yang telah dilakukan, implementasi mekanisme IP Edge Caching menggunakan Apache Traffic Server untuk Web Server Website Multimedia dan Website Video on Demand menghasilkan perbedaan yang sangat signifikan untuk parameter pengujian jaringan dan server seperti Throughput, Round-Trip Time, Delay, dan Hit Ratio antara kondisi No-Cache dan Cache.

Penelitian ini hanya dilakukan dengan menggunakan software, yaitu dengan Packet Network Emulator Tool Lab (PNETLab). Penulis menyarankan untuk penelitian selanjutnya agar implementasi menggunakan perangkat yang sesungguhnya (fisik) dengan harapan implementasi

selanjutnya dapat dilakukan secara langsung pada infrastruktur jaringan internet. Penulis juga menyarankan untuk menggunakan Apache Traffic Control sebagai metode. Apache Traffic Control memungkinkan untuk membuat Content Delivery Network berskala besar dengan menggunakan open source. Dengan Apache Traffic Control, operator dapat membuat Content Delivery Network untuk mengirimkan konten dengan efektif dan efisien kepada pengguna.

REFERENSI

[1] K. Raja, B. Abilash, S. Anbalagan, K. Dev, and A. Ganapathisubramaniyan, "Popularity based content caching of YouTube data in cellular networks," *Multimed Tools Appl*, vol. 81, no. 26, pp. 37165–37182, Nov. 2022, doi: 10.1007/s11042-022-13528-1.

[2] E. Zagan and M. Danubianu, "Data Lake Architecture for Storing and Transforming Web Server Access Log Files," *IEEE Access*, vol. 11, pp. 40916–40929, 2023, doi: 10.1109/ACCESS.2023.3270368.

[3] IEEE Staff, *2017 IFIP IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017.

[4] C. Dian zi ke ji da xue (Chengdu and Institute of Electrical and Electronics Engineers, *2017 9th International Conference on Advanced Infocomm Technology (ICAIT) : Nov. 22 - Nov. 24, 2017, Chengdu, China.*