

ANALISIS PERFORMA SERVER PADA IMPLEMENTASI MIGRASI PROTOKOL HTTP KE HTTPS MENGGUNAKAN NGINX PROXY MANAGER

Fitrah Habibullah ^{1*}, Antoni Haikal ^{2**}

* Teknik Informatika, Politeknik Negeri Batam

** Rekayasa Keamanan Siber, Politeknik Negeri Batam

fitrahhabibullah0@gmail.com ¹, antoni@polibatam.ac.id ²

Article Info

Article history:

Received ...

Revised ...

Accepted ...

Keyword:

HTTPS, Nginx Proxy Manager, Analisis Performa Server, Response Time, Throughput

ABSTRACT

Migrasi protokol Hypertext Transfer Protocol (HTTP) ke Hypertext Transfer Protocol Secure (HTTPS) diperlukan untuk meningkatkan keamanan komunikasi data melalui enkripsi SSL/TLS. Namun, proses enkripsi berpotensi menambah beban komputasi yang dapat memengaruhi performa server. Penelitian ini bertujuan untuk menganalisis dampak migrasi HTTP ke HTTPS menggunakan Nginx Proxy Manager sebagai reverse proxy terhadap performa server. Metode yang digunakan adalah pendekatan eksperimental kuantitatif dengan membandingkan response time, throughput, serta penggunaan sumber daya server pada tiga virtual machine. Hasil pengujian menggunakan Apache Benchmark dilakukan sebanyak dua kali percobaan dan nilai yang disajikan merupakan rata-rata dari kedua percobaan tersebut, sehingga hasil yang diperoleh lebih representatif. Hasil tersebut menunjukkan bahwa rata-rata response time pada VM 1 sebesar 107,60 ms pada HTTP dan 114,38 ms pada HTTPS, pada VM 2 sebesar 167,81 ms pada HTTP dan 159,89 ms pada HTTPS, serta pada VM 3 sebesar 143,43 ms pada HTTP dan 167,75 ms pada HTTPS. Pengujian menggunakan Apache JMeter menunjukkan bahwa nilai throughput pada VM 1 untuk HTTP sebesar 9,6 request/sec, 4,8 request/sec, dan 5,5 request/sec, serta HTTPS sebesar 9,4 request/sec, 5,4 request/sec, dan 5,5 request/sec pada skenario 100, 200, dan 300 pengguna. Pada VM 2, throughput HTTP tercatat sebesar 9,3 request/sec, 5,9 request/sec, dan 5,9 request/sec, sedangkan HTTPS sebesar 9,0 request/sec, 6,6 request/sec, dan 6,5 request/sec. Pada VM 3, throughput HTTP sebesar 8,2 request/sec, 6,1 request/sec, dan 6,3 request/sec, sementara HTTPS sebesar 9,8 request/sec, 6,5 request/sec, dan 6,4 request/sec, di mana pada beberapa kondisi HTTPS menunjukkan nilai yang sebanding bahkan lebih tinggi dibandingkan HTTP. Selain itu, pemantauan performa server menunjukkan bahwa rata-rata penggunaan CPU pada kondisi HTTP tercatat sebesar 60,75% pada VM 1, 61,50% pada VM 2, dan 61,50% pada VM 3, sedangkan pada kondisi HTTPS masing-masing sebesar 62,50%, 62,75%, dan 62,75%. Rata-rata penggunaan memori pada kondisi HTTP tercatat sebesar 1,15 GB pada VM 1, 1,18 GB pada VM 2, dan 1,10 GB pada VM 3, sedangkan pada kondisi HTTPS masing-masing sebesar 1,20 GB pada ketiga virtual machine. Hasil penelitian ini menunjukkan bahwa implementasi HTTPS menggunakan Nginx Proxy Manager mampu meningkatkan keamanan komunikasi tanpa menimbulkan perbedaan konsumsi sumber daya yang berarti terhadap performa server.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

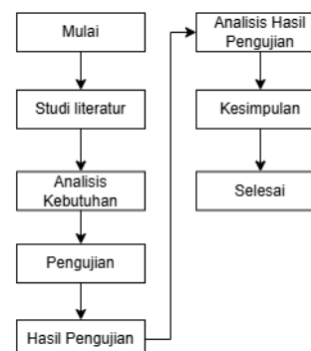
Perkembangan layanan berbasis web menuntut sistem server yang tidak hanya andal tetapi juga aman. Salah satu aspek penting dalam keamanan komunikasi data adalah penggunaan protokol yang mendukung enkripsi. Hypertext Transfer Protocol (HTTP) masih banyak digunakan pada sistem internal organisasi karena sederhana dan efisien, namun protokol ini tidak menyediakan mekanisme enkripsi sehingga rentan terhadap serangan seperti man-in-the-middle dan penyadapan data[1]. Sebagai solusi, Hypertext Transfer Protocol Secure (HTTPS) memanfaatkan Secure Socket Layer (SSL) dan Transport Layer Security (TLS) untuk mengenkripsi komunikasi antara client dan server. Implementasi HTTPS mampu menjamin kerahasiaan, integritas, dan autentikasi data selama transmisi[2]. Meskipun demikian, proses enkripsi dan dekripsi memerlukan sumber daya tambahan yang berpotensi memengaruhi performa server[2], khususnya pada parameter response time, throughput, serta penggunaan CPU dan memori.

Beberapa penelitian sebelumnya menekankan pentingnya migrasi ke HTTPS untuk meningkatkan keamanan, namun belum banyak yang secara empiris menganalisis dampak performa server setelah migrasi menggunakan Nginx Proxy Manager sebagai reverse proxy. Nginx Proxy Manager menyediakan kemudahan pengelolaan sertifikat SSL/TLS melalui antarmuka grafis[3], sehingga proses migrasi menjadi lebih praktis dibandingkan konfigurasi manual. Namun, penggunaan reverse proxy tambahan serta lapisan enkripsi perlu dievaluasi untuk memastikan tidak terjadi penurunan performa yang signifikan.

Penelitian ini dilakukan untuk menganalisis dampak migrasi protokol HTTP ke HTTPS menggunakan Nginx Proxy Manager terhadap performa server pada sistem monitoring di PT Omni Cloud Indonesia. Analisis dilakukan dengan membandingkan response time, throughput, penggunaan CPU, dan memori pada tiga virtual machine dalam berbagai skenario beban. Hasil penelitian diharapkan dapat memberikan gambaran mengenai keseimbangan antara peningkatan keamanan dan efisiensi kinerja server.

II. METODE

A. Tahapan Penelitian



Gambar 1. Diagram Alur Penelitian

Penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif[4] untuk membandingkan performa server sebelum dan sesudah migrasi dari HTTP ke HTTPS.

Tahap pertama adalah identifikasi masalah, yaitu menganalisis kondisi awal sistem yang masih menggunakan HTTP serta menentukan parameter performa yang akan diuji, meliputi response time, throughput, penggunaan CPU, dan memori. Tahap kedua adalah studi literatur, yang dilakukan untuk memahami konsep HTTP, HTTPS, SSL/TLS, serta peran reverse proxy dalam arsitektur jaringan, sekaligus mengkaji penelitian terdahulu terkait dampak HTTPS terhadap performa server. Tahap ketiga adalah analisis kebutuhan dan perancangan sistem, yang mencakup penentuan spesifikasi perangkat keras dan perangkat lunak serta perancangan lingkungan pengujian menggunakan tiga virtual machine dan satu server reverse proxy berbasis Nginx Proxy Manager. Tahap keempat adalah implementasi dan pengujian sistem, yang dilakukan dalam dua kondisi, yaitu HTTP dan HTTPS. Pengujian performa menggunakan Apache Benchmark untuk mengukur response time dan Apache JMeter untuk mengukur throughput, sedangkan penggunaan CPU dan memori dimonitor menggunakan htop. Tahap terakhir adalah analisis data, yaitu membandingkan hasil pengujian kedua protokol secara deskriptif-komparatif untuk mengetahui dampak migrasi terhadap performa server.

B. Analisis Kebutuhan

Kebutuhan sistem dalam penelitian ini dibagi menjadi dua kategori utama, yaitu perangkat keras dan perangkat lunak. Perangkat keras disiapkan untuk mendukung operasional tiga virtual machine dan satu server reverse proxy, sedangkan perangkat lunak meliputi sistem operasi, aplikasi reverse proxy, serta tools pengujian dan monitoring performa server. Spesifikasi detail perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini disajikan pada tabel berikut:

Kebutuhan Hardware	Detail Spesifikasi
Prosesor	8 core (2 core, 4 socket)
Ram	16 Gb
Storage	100 Gb SSD

Tabel 1. Kebutuhan Hardware

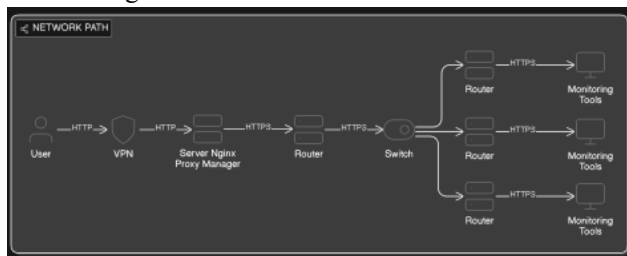
Tabel 1 di atas merupakan kebutuhan hardware yang di gunakan oleh sistem pada penelitian ini.

Kebutuhan Software	Detail Spesifikasi	Versi
Operating System	Operating System	24.04 LTS
System Monitoring	Libre NMS	26.1.0
Reverse Proxy	Nginx Proxy Manager	2.13.6
Tools Test	Apache Benchmark	2.3
	Jmeter	5.6.3
	HTOP	3.3.0

Tabel 2. Kebutuhan Software

Tabel 2 di atas merupakan kebutuhan perangkat lunak yang dibutuhkan oleh sistem, perangkat lunak apa saja yang akan digunakan pada penelitian ini.

C. Perancangan Sistem



Gambar 2. Topologi Sistem

Berdasarkan Gambar 2, ditunjukkan rancangan topologi jaringan yang digunakan dalam penelitian ini. Lingkungan pengujian terdiri dari tiga server virtual yang ditempatkan pada lokasi berbeda. Seluruh server tersebut terhubung melalui satu server utama yang berfungsi sebagai reverse proxy dengan memanfaatkan Nginx Proxy Manager (NPM). Server NPM berperan sebagai pusat pengelolaan lalu lintas jaringan sekaligus pengaturan sertifikat SSL/TLS dalam proses migrasi protokol. Alur komunikasi dimulai dari client yang mengirimkan permintaan (request) menuju server Nginx Proxy Manager. Selanjutnya, permintaan tersebut diteruskan ke salah satu dari tiga server virtual sesuai dengan konfigurasi yang telah ditetapkan. Dengan demikian, seluruh akses dari client harus melalui server reverse proxy terlebih dahulu sebelum mencapai server tujuan, sehingga mekanisme pengamanan dan pengelolaan koneksi dapat dilakukan secara terpusat dan terkontrol.

D. Implementasi Sistem

Tahap implementasi sistem merupakan tahap di mana seluruh perangkat keras dan perangkat lunak yang telah dirancang sebelumnya diinstal dan dikonfigurasi untuk memastikan sistem berjalan sesuai dengan desain arsitektur yang telah ditetapkan. Pada fase ini, setiap virtual machine dikonfigurasi menggunakan sistem operasi Ubuntu Server serta dipersiapkan untuk menjalankan layanan monitoring berbasis web. Implementasi dimulai dengan konfigurasi layanan web menggunakan protokol HTTP sebagai kondisi awal pengujian. Selanjutnya, dilakukan instalasi dan konfigurasi Nginx Proxy Manager pada server utama yang berfungsi sebagai reverse proxy. Nginx Proxy Manager dikonfigurasi untuk mengelola host, mengarahkan lalu lintas ke masing-masing server virtual, serta mengaktifkan sertifikat SSL/TLS guna mengimplementasikan protokol HTTPS. Proses migrasi dilakukan dengan menambahkan sertifikat SSL/TLS pada setiap host yang telah dikonfigurasi, sehingga komunikasi antara client dan server berlangsung secara terenkripsi. Setelah konfigurasi selesai, dilakukan pengujian awal untuk memastikan bahwa layanan dapat diakses dengan baik pada kedua kondisi, yaitu HTTP dan HTTPS, sebelum dilakukan pengujian performa lebih lanjut menggunakan tools benchmarking.

E. Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi performa server sebelum dan sesudah migrasi dari protokol HTTP ke HTTPS menggunakan Nginx Proxy Manager sebagai reverse proxy. Pengujian dilakukan pada tiga virtual machine dalam dua kondisi, yaitu saat layanan menggunakan HTTP dan setelah diimplementasikan HTTPS dengan SSL/TLS. Parameter yang dianalisis meliputi response time, throughput, serta penggunaan sumber daya CPU dan memori.

1. Apache Benchmark

Apache Benchmark Apache Benchmarking Tool adalah sebuah alat yang dibuat oleh Apache Organization yang dapat bekerja untuk mengukur performansi pada Hypertext Transfer Protocol (HTTP) web Server. Tool ini dapat menghitung banyak request per second yang dapat dilayani oleh web Server yang digunakan[5]. Pengujian dilakukan dengan variasi jumlah permintaan dan jumlah koneksi paralel yang konsisten. Variasi jumlah permintaan yang diuji meliputi 1.000 permintaan, 5.000 permintaan, 10.000 permintaan, dan 20.000 permintaan. Jumlah koneksi paralel yang digunakan dalam setiap pengujian ditetapkan sebanyak 10 koneksi (-c 10) untuk menjaga konsistensi dalam proses analisis kinerja server[6]. Pengujian dilakukan sebanyak dua kali percobaan pada

masing-masing virtual machine dalam kondisi HTTP dan HTTPS, dan nilai response time yang digunakan dalam analisis merupakan hasil rata-rata dari kedua percobaan tersebut untuk memperoleh data yang lebih representatif dan meminimalisir pengaruh fluktuasi yang terjadi pada satu kali pengujian. Nilai response time yang diperoleh kemudian dianalisis berdasarkan rata-rata pada setiap skenario beban untuk mengetahui dampak implementasi enkripsi SSL/TLS terhadap kecepatan respons server.

2. Apache JMeter

Apache JMeter digunakan untuk mensimulasikan akses pengguna secara simultan guna mengukur throughput server[7]. Skenario pengujian dilakukan dengan tiga variasi jumlah pengguna (thread). Parameter utama yang diukur adalah throughput (request per second), yang menunjukkan jumlah permintaan yang mampu diproses server dalam satuan detik. Pengujian dilakukan pada kedua kondisi protokol, yaitu HTTP dan HTTPS, untuk menganalisis apakah migrasi ke HTTPS memengaruhi kapasitas server dalam menangani beban akses secara bersamaan.

3. HTOP

HTOP digunakan sebagai alat monitoring untuk mengamati penggunaan sumber daya server secara real-time selama proses pengujian berlangsung. Parameter yang diamati meliputi persentase penggunaan CPU dan memori pada masing-masing virtual machine. Data penggunaan CPU dan memori dicatat pada setiap skenario pengujian untuk mengidentifikasi adanya peningkatan beban sistem akibat proses enkripsi dan dekripsi SSL/TLS. Informasi ini digunakan sebagai pendukung dalam analisis dampak migrasi protokol terhadap efisiensi penggunaan sumber daya server[8].

Htop merupakan tool yang digunakan untuk memonitor program yang sedang berjalan dengan interaktif. Htop dapat menyampaikan informasi gambaran tentang prosesor, CPU dan memori yang digunakan[9].

Namun, HTOP hanya menampilkan kondisi sistem secara langsung tanpa menyediakan mekanisme penyimpanan data historis, sehingga kurang optimal untuk kebutuhan analisis lebih lanjut. Oleh karena itu, dilakukan integrasi dengan Prometheus sebagai sistem monitoring yang berfungsi untuk mengumpulkan, menyimpan, dan mengolah data performa server. Grafana digunakan sebagai platform open-source untuk monitoring dan visualisasi data yang mendukung integrasi dengan

berbagai sumber data seperti Prometheus, serta memungkinkan pembuatan dashboard interaktif dalam bentuk grafik, tabel, maupun visualisasi lainnya[10].

F. Analisis Hasil Pengujian

Analisis dilakukan untuk membandingkan performa server pada dua kondisi pengujian, yaitu sebelum dan sesudah migrasi dari protokol HTTP ke HTTPS. Parameter yang dianalisis meliputi response time, throughput, penggunaan CPU, dan penggunaan memori pada masing-masing virtual machine. Hasil pengujian Apache Benchmark dievaluasi berdasarkan nilai rata-rata response time pada setiap variasi jumlah request, di mana nilai rata-rata yang digunakan merupakan hasil rata-rata dari dua kali percobaan pengujian yang dilakukan pada masing-masing skenario untuk memperoleh data yang lebih representatif[8]. Hasil pengujian Apache JMeter dianalisis berdasarkan nilai throughput pada setiap skenario jumlah pengguna simultan. Data penggunaan CPU dan memori digunakan untuk mengidentifikasi adanya peningkatan beban sistem akibat proses enkripsi SSL/TLS. Pendekatan yang digunakan dalam analisis ini adalah deskriptif-komparatif, yaitu dengan membandingkan selisih nilai rata-rata antara kondisi HTTP dan HTTPS pada setiap parameter yang diukur. Analisis tersebut bertujuan untuk mengetahui apakah migrasi ke HTTPS memberikan dampak signifikan terhadap performa server, serta untuk menilai keseimbangan antara peningkatan keamanan dan efisiensi kinerja sistem. Hasil analisis ini selanjutnya menjadi dasar dalam penarikan kesimpulan mengenai efektivitas implementasi HTTPS menggunakan Nginx Proxy Manager pada lingkungan server monitoring.

III. HASIL DAN PEMBAHASAN

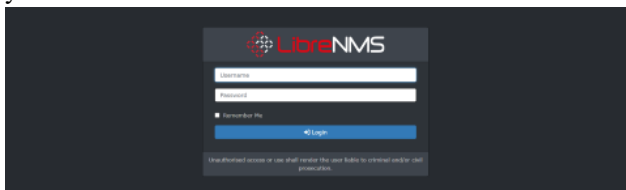
A. Implementasi Sistem

Pada tahap implementasi, sistem migrasi protokol dari HTTP ke HTTPS diterapkan menggunakan Nginx Proxy Manager sebagai reverse proxy untuk mengelola lalu lintas jaringan dan sertifikat SSL/TLS. Implementasi dilakukan pada tiga virtual machine yang menjalankan layanan monitoring berbasis web dan terhubung melalui satu server utama yang menjalankan Nginx Proxy Manager. Proses implementasi diawali dengan konfigurasi layanan menggunakan protokol HTTP sebagai kondisi awal pengujian, kemudian dilanjutkan dengan aktivasi HTTPS melalui penerapan sertifikat SSL/TLS pada reverse proxy. Dengan konfigurasi tersebut, komunikasi antara client dan server dapat berlangsung secara terenkripsi sehingga memungkinkan dilakukan pengujian performa sebelum dan sesudah migrasi protokol.

1. Implementasi vm System Monitoring

Pada tahap implementasi system monitoring, pengujian dilakukan menggunakan sistem operasi Ubuntu Server

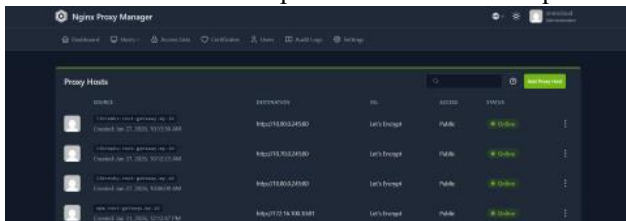
24.04 LTS pada tiga Virtual Machine (VM) yang berfungsi sebagai server monitoring dan ditempatkan pada lokasi yang berbeda. Ketiga VM tersebut dikonfigurasi untuk menjalankan layanan monitoring berbasis web yang dapat diakses oleh client melalui server reverse proxy. Seluruh VM terhubung dengan satu server utama yang menjalankan Nginx Proxy Manager sebagai reverse proxy untuk mengatur lalu lintas koneksi serta pengelolaan sertifikat SSL/TLS. Dengan konfigurasi ini, setiap permintaan dari client akan diarahkan terlebih dahulu ke server reverse proxy sebelum diteruskan ke masing-masing VM system monitoring, sehingga memungkinkan dilakukan pengujian performa server pada dua kondisi protokol, yaitu HTTP dan HTTPS.



Gambar 3. Sistem Monitoring

2. Implementasi Nginx Proxy Manager

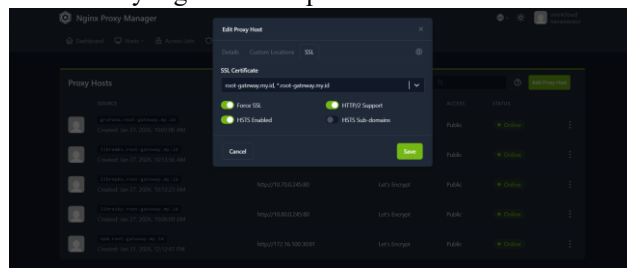
Implementasi sistem reverse proxy dilakukan menggunakan Nginx Proxy Manager yang dijalankan pada server utama. Nginx Proxy Manager berfungsi sebagai pusat pengelolaan koneksi antara client dan tiga virtual machine yang menjalankan layanan system monitoring. Melalui antarmuka berbasis web, Nginx Proxy Manager dikonfigurasi untuk mengatur proxy host, mengarahkan permintaan client ke masing-masing virtual machine sesuai konfigurasi yang telah ditetapkan, serta mengelola sertifikat SSL/TLS untuk mengaktifkan protokol HTTPS. Dengan konfigurasi tersebut, seluruh permintaan dari client akan melewati server reverse proxy terlebih dahulu sebelum diteruskan ke server tujuan, sehingga pengelolaan lalu lintas jaringan dan keamanan komunikasi dapat dilakukan secara terpusat.



Gambar 4. Dashboard Nginx Proxy Manager

- Implementasi HTTPS dilakukan dengan menambahkan sertifikat SSL/TLS pada layanan yang dikelola melalui Nginx Proxy Manager. Proses ini dilakukan dengan mengaktifkan fitur pengelolaan sertifikat pada setiap *proxy host* yang telah dikonfigurasi sebelumnya melalui tab SSL pada antarmuka Nginx Proxy Manager. Sertifikat SSL/TLS yang digunakan diterbitkan oleh Let's Encrypt dengan nama domain *root-gateway.my.id*. Sertifikat SSL/TLS digunakan untuk mengenkripsi komunikasi antara *client* dan server sehingga data yang ditransmisikan menjadi lebih aman[11].

Selain penerapan sertifikat, beberapa konfigurasi tambahan juga diaktifkan untuk memperkuat keamanan komunikasi. Fitur *Force SSL* diaktifkan untuk memastikan seluruh permintaan HTTP secara otomatis dialihkan ke HTTPS. Fitur *HTTP/2 Support* diaktifkan untuk meningkatkan efisiensi transmisi data antara *client* dan server. Fitur *HSTS (HTTP Strict Transport Security) Enabled* diaktifkan untuk menginstruksikan peramban agar selalu menggunakan koneksi HTTPS pada domain tersebut. Dengan seluruh konfigurasi tersebut, layanan yang sebelumnya diakses menggunakan protokol HTTP telah sepenuhnya dialihkan ke HTTPS dengan lapisan keamanan yang lebih komprehensif.



Gambar 5. Konfigurasi Implementasi SSL Sertifikat

B. Pengujian Sistem dan Hasil

Pengujian sistem dilakukan untuk mengevaluasi performa server setelah implementasi migrasi protokol dari HTTP ke HTTPS menggunakan Nginx Proxy Manager sebagai reverse proxy. Pengujian ini bertujuan untuk mengukur kinerja server dalam menangani permintaan client pada berbagai skenario beban. Proses pengujian dilakukan dengan membandingkan dua kondisi sistem, yaitu saat menggunakan protokol HTTP dan setelah diimplementasikan protokol HTTPS dengan sertifikat SSL/TLS. Parameter yang dianalisis dalam pengujian ini meliputi response time, throughput, serta penggunaan sumber daya server berupa CPU dan memori pada tiga virtual machine yang menjalankan layanan system monitoring.

1. Skenario Pengujian Apache Benchmark

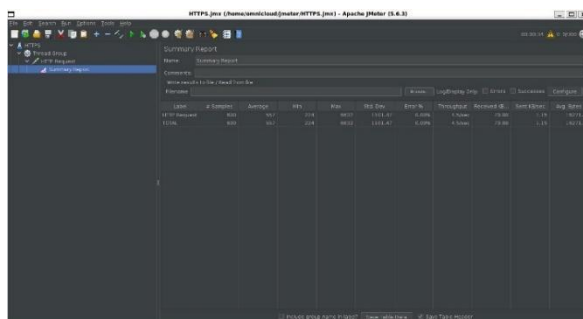
Pengujian menggunakan Apache Benchmark dilakukan untuk mengukur waktu respons (response time) server dalam menangani sejumlah permintaan dari client. Pengujian dilakukan pada tiga virtual machine dengan dua kondisi protokol, yaitu HTTP dan HTTPS. Variasi jumlah permintaan yang digunakan dalam pengujian meliputi 1.000, 5.000, 10.000, dan 20.000 permintaan dengan jumlah koneksi paralel yang ditetapkan sebanyak 10 koneksi (-c 10). Konfigurasi ini digunakan untuk mensimulasikan tingkat beban akses yang berbeda secara konsisten pada setiap skenario pengujian[6]. Setiap skenario pengujian dilakukan sebanyak dua kali percobaan, dan nilai response time yang digunakan dalam analisis merupakan hasil rata-rata dari kedua percobaan tersebut guna memperoleh data yang lebih representatif dan meminimalisir pengaruh fluktuasi yang dapat terjadi.

Pengujian menggunakan Apache JMeter dilakukan untuk mensimulasikan akses pengguna secara bersamaan guna mengukur throughput server[12]. Skenario pengujian seperti table di bawah ini.

Skenario	Jumlah Thread (Users)	Ramp-Up Period (detik)	Loop Count
1	100	10	1
2	200	20	1
3	300	30	1

Tabel 3. Skenario Pengujian Apache Jmeter

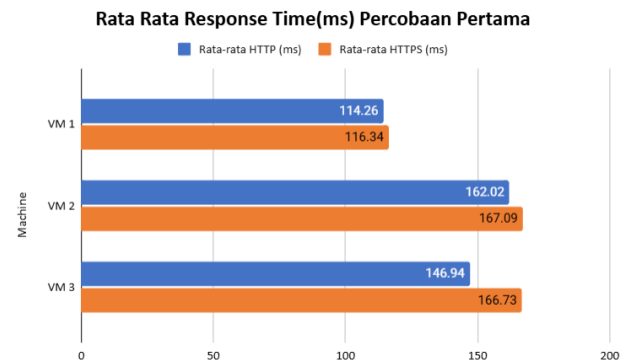
Pengujian ini bertujuan untuk mengetahui kemampuan server dalam menangani permintaan dari banyak pengguna secara simultan pada kondisi HTTP dan HTTPS.



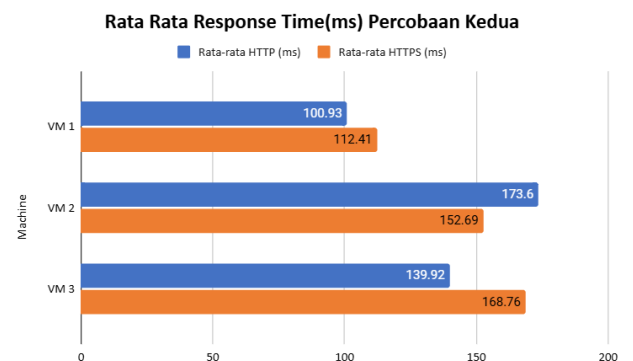
Gambar 6. Implemtasi Apache Jmeter

2. Hasil Pengujian Apache Benchmark

Berikut adalah hasil pengujian menggunakan Apache Benchmark dengan skenario yang sudah di tentukan, dengan sumbu X = "Jumlah Request" dan sumbu Y = "Response Time (ms)".



Gambar 7. Hasil Rata Rata Pengujian Response time (ms) pada percobaan pertama



Gambar 8. Hasil Rata Rata Pengujian Response time (ms) pada percobaan kedua

3. Hasil Pengujian Apache Jmeter

Berikut adalah hasil pengujian menggunakan Apache Jmeter dengan skenario yang sudah di tentukan.

Jumlah Thread (Users)	Ramp-Up Period (detik)	HTTP (req/sec)	HTTPS (req/sec)
100	10	9.6	9.4
200	20	4.8	5.4
300	30	5.5	5.5

Tabel 4. Hasil Pengujian Apache Jmeter ((req/sec) VM 1

Jumlah Thread (Users)	Ramp-Up Period (detik)	HTTP (req/sec)	HTTPS (req/sec)
100	10	9.3	9
200	20	5.9	6.6
300	30	5.9	6.5

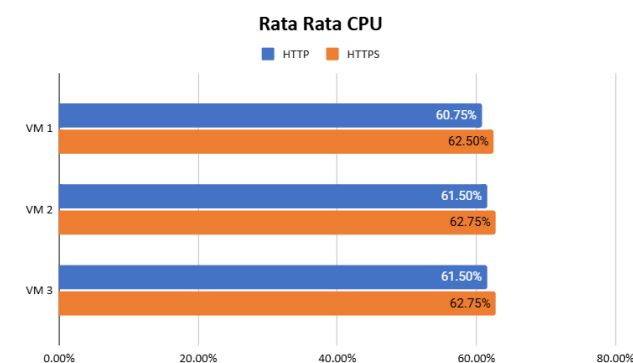
Tabel 5. Hasil Pengujian Apache Jmeter (req/sec) VM 2

Jumlah Thread (Users)	Ramp-Up Period (detik)	HTTP (req/sec)	HTTPS (req/sec)
100	10	8.2	9.8
200	20	6.1	6.5
300	30	6.3	6.4

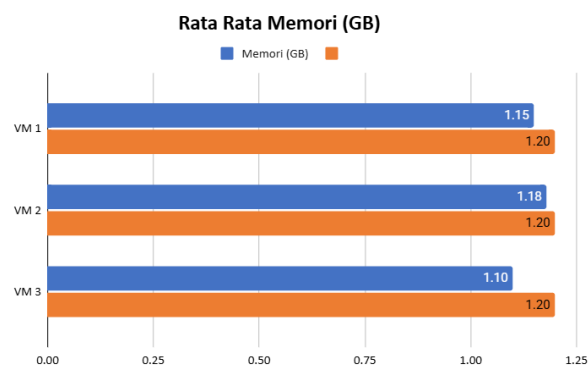
Tabel 6. Hasil Pengujian Apache Jmeter (req/sec) VM 3

4. Hasil Pengujian CPU dan Memory Server

Selain melakukan pengujian response time dan throughput, penelitian ini juga melakukan pemantauan terhadap penggunaan sumber daya server selama proses pengujian berlangsung. Parameter yang diamati meliputi penggunaan CPU dan Memori (RAM) pada saat sistem menangani permintaan dari client dengan variasi jumlah request yang berbeda. Pemantauan dilakukan menggunakan sistem monitoring HTOP untuk melihat perubahan penggunaan sumber daya server pada kondisi HTTP dan HTTPS, dengan sumbu X = "Jumlah Request" dan sumbu Y = "Penggunaan CPU (%)" dan "Penggunaan Memori (GB)".



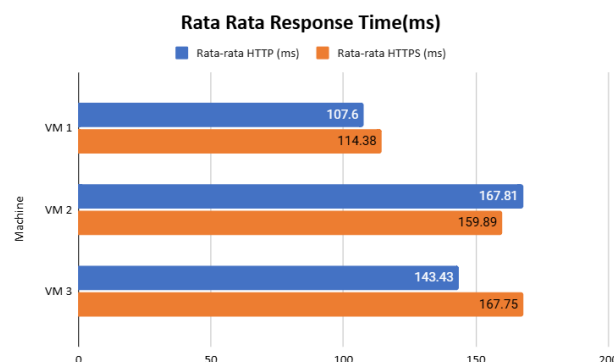
Gambar 9. Hasil Analisis Performa CPU(%)



Gambar 10. Hasil Analisis Performa Memori(GB)

C. Analisis Hasil Pengujian

Pengujian dilakukan untuk menganalisis performa server setelah implementasi migrasi protokol dari HTTP ke HTTPS menggunakan Nginx Proxy Manager sebagai reverse proxy. Pengujian dilakukan pada tiga virtual machine yang menjalankan layanan system monitoring dengan beberapa skenario beban. Parameter yang dianalisis meliputi response time yang diperoleh dari pengujian Apache Benchmark serta throughput yang diperoleh dari pengujian Apache JMeter.



Gambar 11. Hasil Rata – Rata Response time (ms) Apache Benchmark

Berdasarkan hasil pengujian Apache Benchmark yang ditunjukkan pada gambar di atas, diperoleh nilai rata-rata response time pada masing-masing virtual machine untuk kondisi HTTP dan HTTPS. Nilai rata-rata yang disajikan merupakan hasil rata-rata dari dua kali percobaan pengujian yang dilakukan, sehingga nilai yang diperoleh lebih representatif dan dapat meminimalisir pengaruh fluktuasi yang terjadi pada satu kali pengujian. Pada VM 1 diperoleh rata-rata response time sebesar 107.60 ms untuk HTTP dan 114.38 ms untuk HTTPS. Pada VM 2 diperoleh rata-rata response time sebesar 167.81 ms untuk HTTP dan 159.89 ms untuk HTTPS. Sementara itu pada VM 3 diperoleh rata-rata response time sebesar 143.43 ms untuk HTTP dan 167.75 ms untuk HTTPS. Data tersebut menunjukkan adanya perbedaan nilai response time antara kedua protokol pada masing-

masing virtual machine pada skenario pengujian yang dilakukan..

Selanjutnya, pengujian menggunakan Apache JMeter dilakukan untuk mengukur throughput server dalam menangani akses pengguna secara simultan. Pengujian dilakukan pada tiga skenario jumlah pengguna, yaitu 100, 200, dan 300 pengguna. Pada VM 1, nilai throughput HTTP masing-masing sebesar 9.6 request/sec, 4.8 request/sec, dan 5.5 request/sec, sedangkan pada HTTPS sebesar 9.4 request/sec, 5.4 request/sec, dan 5.5 request/sec. Pada VM 2, throughput HTTP tercatat sebesar 9.3 request/sec, 5.9 request/sec, dan 5.9 request/sec, sementara pada HTTPS sebesar 9.0 request/sec, 6.6 request/sec, dan 6.5 request/sec. Sementara itu, pada VM 3, throughput HTTP tercatat sebesar 8.2 request/sec, 6.1 request/sec, dan 6.3 request/sec, sedangkan pada HTTPS sebesar 9.8 request/sec, 6.5 request/sec, dan 6.4 request/sec. Data tersebut menunjukkan variasi nilai throughput pada masing-masing skenario pengujian baik pada kondisi HTTP maupun HTTPS dalam menangani peningkatan jumlah pengguna secara bersamaan. Pada beberapa skenario pengujian, nilai *throughput* HTTPS bahkan melampaui HTTP, yang secara teknis disebabkan oleh dukungan HTTP/2 dan mekanisme *TLS session resumption* pada Nginx Proxy Manager yang mampu mengoptimalkan penanganan permintaan secara paralel

Selain parameter response time dan throughput, pemantauan juga dilakukan terhadap penggunaan sumber daya server selama proses pengujian berlangsung. Berdasarkan hasil pemantauan yang ditunjukkan pada Gambar 10, penggunaan CPU meningkat secara bertahap seiring dengan meningkatnya jumlah permintaan yang diproses oleh server. Rata-rata penggunaan CPU pada kondisi HTTP tercatat sebesar 60,75% pada VM 1, 61,50% pada VM 2, dan 61,50% pada VM 3, sedangkan pada kondisi HTTPS masing-masing sebesar 62,50%, 62,75%, dan 62,75%. Selisih rata-rata penggunaan CPU antara HTTP dan HTTPS secara keseluruhan hanya sebesar 1,42%, yang menunjukkan bahwa proses enkripsi SSL/TLS tidak menimbulkan peningkatan beban komputasi yang berarti terhadap server.

Sementara itu, pemantauan penggunaan memori (RAM) menunjukkan bahwa penggunaan memori selama proses pengujian cenderung stabil. Rata-rata penggunaan memori pada kondisi HTTP tercatat sebesar 1,150 GB pada VM 1, 1,175 GB pada VM 2, dan 1,100 GB pada VM 3, sedangkan pada kondisi HTTPS masing-masing sebesar 1,20 GB pada ketiga virtual machine. Hal ini mengindikasikan bahwa peningkatan jumlah permintaan yang dikirimkan ke server tidak mengakibatkan perbedaan konsumsi memori yang berarti baik pada kondisi HTTP maupun HTTPS. Apabila ditinjau dari perspektif trade-off antara keamanan dan performa, peningkatan overhead akibat migrasi ke HTTPS pada ketiga virtual machine dinilai proporsional mengingat

manfaat keamanan yang diperoleh, yaitu perlindungan terhadap serangan man-in-the-middle, penyadapan data, serta jaminan integritas dan autentikasi komunikasi. Dengan demikian, dampak performa yang ditimbulkan dari penerapan lapisan keamanan HTTPS pada lingkungan ini berada pada batas yang wajar dan dapat diterima secara teknis maupun operasional.

Secara keseluruhan, hasil pengujian *response time*, *throughput*, serta pemantauan penggunaan CPU dan memori memberikan gambaran mengenai performa server selama proses pengujian berlangsung pada kedua protokol yang diuji.

IV. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, implementasi migrasi protokol dari HTTP ke HTTPS menggunakan Nginx Proxy Manager sebagai reverse proxy tidak menunjukkan penurunan performa server yang berarti. Hasil pengujian menggunakan Apache Benchmark yang dilakukan sebanyak dua kali percobaan menunjukkan bahwa nilai response time pada kondisi HTTP dan HTTPS berada pada rentang yang tidak jauh berbeda pada ketiga virtual machine yang diuji. Pengujian menggunakan Apache JMeter juga menunjukkan bahwa nilai throughput pada kondisi HTTPS tetap mampu menangani akses pengguna secara simultan pada berbagai skenario jumlah pengguna. Pada beberapa skenario pengujian, nilai throughput HTTPS bahkan melampaui HTTP, yang secara teknis disebabkan oleh dukungan HTTP/2 dan mekanisme *TLS session resumption* pada Nginx Proxy Manager yang mampu mengoptimalkan penanganan permintaan secara paralel.

Selain itu, hasil pemantauan performa server menunjukkan bahwa proses enkripsi SSL/TLS tidak menimbulkan peningkatan beban komputasi yang berarti, baik dari sisi penggunaan CPU maupun memori pada ketiga virtual machine yang diuji. Hal ini mengindikasikan bahwa peningkatan jumlah permintaan tidak mengakibatkan perbedaan konsumsi sumber daya yang berarti antara kondisi HTTP maupun HTTPS. Berdasarkan hasil tersebut, implementasi HTTPS menggunakan Nginx Proxy Manager mampu meningkatkan keamanan komunikasi data melalui enkripsi SSL/TLS tanpa mengorbankan efisiensi kinerja server secara keseluruhan.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada PT Omni Cloud Indonesia atas dukungan fasilitas dan lingkungan pengujian yang diberikan dalam pelaksanaan penelitian ini. Ucapan terima kasih juga penulis sampaikan kepada dosen pembimbing, rekan-rekan, serta seluruh pihak yang telah memberikan bantuan, arahan, dan dukungan, baik secara langsung maupun tidak langsung, sehingga penelitian ini dapat terselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] H. Endah Wahanani, F. Prima Aditiawan, and R. Mumpuni, "UJI COBA SERANGAN MAN IN THE MIDDLE PADA KEAMANAN SSL PROTOKOL HTTP."
- [2] C. Nova, P. Ridwan, M. Sinun, and H. A. Malaiga, "PERKEMBANGAN DALAM PROTOKOL HTTP: MENGATASI TANTANGAN PERFORMA DAN KEAMANAN DI ERA INTERNET YANG TERUS BERKEMBANG," 2024.
- [3] M. Y. S. Putra and A. S. Affandi, "Analisis Tingkat Efektivitas Cloudflare, Docker dan Nginx Proxy Manager Sebagai Sarana untuk Meningkatkan Keamanan Web," *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 19, no. 1, pp. 24–30, Mar. 2025, doi: 10.32815/jitika.v19i1.1070.
- [4] A. Hasiholan, M. Fani, R. K. Siber, and P. N. Batam, "Analisis Komparatif Deteksi Malware pada SIEM Wazuh Melalui Integrasi VirusTotal dan Yara: Studi Kasus File Upload di Rootguards." [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [5] "PERBANDINGAN PERFORMANSI WEB SERVER APACHE DAN NGINX DENGAN MENGGUNAKAN IPV6".
- [6] S. Akva etal, S. Akva, Y. Lorinez, and J. Tampubolon, "Analisis Penggunaan Load Balancing dengan HAProxy terhadap Kinerja Web Server pada Ubuntu-Analisis Penggunaan Load Balancing dengan HAProxy terhadap Kinerja Web Server pada Ubuntu," vol. 07, 2025, doi: 10.54209/jatilima.v7i01.989.
- [7] Riswandi, Kasim, and Muh. F. Raharjo, "Evaluasi Kinerja Web Server Apache menggunakan Protokol HTTP2," *Journal of Engineering, Technology, and Applied Science*, vol. 2, no. 1, pp. 19–31, Apr. 2020, doi: 10.36079/lamintang.jetas-0201.92.
- [8] M. R. Ramadhan and R. Rachmatika, "Analisis Performa Sistem Informasi Akademik: Pengaruh Struktur Memori dan Manajemen Proses pada Sistem Operasi Linux Menggunakan Metode Eksperimental Kuantitatif," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 5, no. 3, pp. 1009–1018, Nov. 2025, doi: 10.51454/decode.v5i3.1272.
- [9] Y. N. Khamidah, B. Arifwidodo, and M. A. Amanaf, "PENGARUH CLIENT REQUEST PADA WEB SERVER APACHE DAN NGINX DENGAN IPV6 MENGGUNAKAN APACHE BENCHMARK," vol. 25, no. 1, 2024.
- [10] F. Saory, J. H. Jaman, and C. Author, "SISTEM MONITORING SERVER MENGGUNAKAN PROMETHEUS DAN GRAFANA DENGAN INTEGRASI SLACK DI PT. ATLAS LINTAS INDONESIA," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.8024.
- [11] A. Setiawan, M. A. Satrio, I. Madani, R. D. Rachmat, and S. H. Sukma, "Meningkatkan Keamanan Sertifikat Digital dengan Pengaktifan HTTPS," *Journal of Internet and Software Engineering*, vol. 1, no. 4, p. 9, Aug. 2024, doi: 10.47134/pjise.v1i4.3170.
- [12] N. Rahma, A. Karima, A. Ulya, I. Sahputra, and M. Ula, "Analisis Performa Website Perpustakaan Universitas Malikussaleh Menggunakan Metode Load Testing dengan Apache JMeter," *Julyxxxx*, vol. x, No.x, pp. 1–5.