Analisis Kinerja Wireguard Dan Openvpn Dengan Reverse Proxy Untuk Akses Home Server

Maulana Rafi Nurdiansyah
Teknik Informatika
Telkom University Purwokerto
Purwokerto, Indonesia
hipsterweeds@student.telkomuniversity.ac.id

Alon Jala Tirta Segara, S.Kom., M.Kom
Teknik Informatika
Purwokerto, Indonesia
alonhs@telkomuniversity.ac.id

Abstrak--Pengembangan teknologi informasi telah meningkatkan kebutuhan akan akses data *home server* secara fleksibel dan aman dari jarak jauh, namun kombinasi penggunaan *Virtual Private Network* dan *reverse proxy* menimbulkan tantangan terkait kinerja karena mekanisme keamanan seperti enkripsi dan *tunneling* dapat menimbulkan *overhead* pada paket data yang meningkatkan *latency* dan mengurangi *throughput*. Penelitian ini penting karena berbagai protokol *VPN* memiliki karakteristik berbeda dalam menangani *overhead* dan efisiensi pemrosesan yang berdampak langsung pada kinerja sistem, sementara kondisi saat ini menunjukkan bahwa pemilihan protokol *VPN* dan konfigurasi sistem masih berdasarkan asumsi teoretis tanpa evaluasi empiris yang komprehensif. Penelitian ini mengimplementasikan pendekatan analisis empiris terhadap kinerja konfigurasi akses home server menggunakan kombinasi *VPN* dengan membandingkan protokol *WireGuard* dan *OpenVPN* serta penerapan *reverse proxy* berbasis *Nginx*, di mana sistem dirancang dengan topologi jaringan yang terdiri dari dua *Virtual Private Server* sebagai gateway dan pengujian serta dua unit *Raspberry Pi* sebagai home server target dengan empat skenario pengujian yang difokuskan pada parameter *Quality of Service* yaitu *response time*, *latency*, dan penggunaan sumber daya *CPU* serta *RAM*. Hasil pengujian menunjukkan bahwa *OpenVPN* dengan *reverse proxy* menghasilkan *response time* terbaik sebesar 9ms dan *latency* terendah 69,656ms, sementara *WireGuard* tanpa *reverse proxy* paling efisien dalam penggunaan *CPU* dengan konsumsi hanya 16,7%, di mana implementasi *reverse proxy* terbukti memberikan dampak positif terhadap *response time* pada kedua protokol dengan *OpenVPN* menunjukkan adaptasi yang lebih baik terhadap arsitektur *reverse proxy*, sehingga penelitian ini berkontribusi memberikan panduan empiris dalam pemilihan protokol *VPN* yang sesuai untuk implementasi home server dengan arsitektur *reverse proxy*.

Kata kunci Wireguard, Open vpn, *Reverse Proxy*, *Home Server*, VPN

I. PENDAHULUAN

Pengembangan teknologi informasi dan komunikasi telah secara signifikan meningkatkan kebutuhan akan akses data dan layanan secara fleksibel dan aman, tidak terbatas pada lokasi fisik [1]. Fenomena ini mendorong peningkatan adopsi solusi yang memungkinkan pengguna untuk terhubung ke jaringan pribadi dari lokasi mana pun secara aman, termasuk akses ke sumber daya personal seperti *home server*. Kebutuhan akan kemampuan mengakses data dan aplikasi di *home server* dari luar jaringan lokal semakin umum, baik untuk keperluan kerja jarak jauh maupun pengelolaan data pribadi [3].

Akses jarak jauh yang aman ke jaringan pribadi, *Virtual Private Network (VPN)* adalah solusi yang sudah mapan untuk menciptakan kanal komunikasi yang aman melalui infrastruktur publik seperti internet [4]. Dengan teknik enkripsi dan *tunneling*, *VPN* menjamin kerahasiaan, integritas, dan otentikasi data yang ditransmisikan [5]. Selain *VPN*, *reverse proxy* juga makin populer sebagai lapisan di depan *home server* untuk mengelola koneksi masuk, memberikan fitur keamanan tambahan seperti terminasi *SSL/TLS*, otentikasi, dan *load balancing* [6]. Penggunaan *reverse proxy* juga menyederhanakan akses ke berbagai layanan internal melalui satu titik masuk *public* [7].

Kombinasi penggunaan *VPN* dan *reverse proxy* menimbulkan tantangan kinerja. Mekanisme keamanan yang diterapkan oleh *VPN*, seperti enkripsi dan *tunneling*, secara alami menimbulkan *overhead* pada paket data, yang dapat meningkatkan latensi dan mengurangi *throughput* [8]. Penambahan lapisan *reverse proxy* di atas koneksi *VPN* berpotensi menambah kompleksitas dalam alur data dan memengaruhi kinerja secara keseluruhan [9]. Hal ini terlihat pada peningkatan waktu *response*, peningkatan latensi dalam transmisi data, serta peningkatan konsumsi sumber daya sistem pada perangkat yang terlibat. Setiap protokol *VPN* memiliki karakteristik berbeda dalam efisiensi pemrosesan yang memengaruhi kinerja [10]. Oleh karena itu, evaluasi empiris terhadap kinerja berbagai konfigurasi menjadi krusial untuk mengidentifikasi solusi yang paling optimal.

Solusi umum mengatasi masalah kinerja potensial melibatkan pemilihan protokol *VPN* yang efisien dan konfigurasi sistem yang tepat. Dua protokol *VPN* yang saat ini banyak digunakan dan diakui karena

efisiensi dan keamanannya adalah *WireGuard* [11] dan *OpenVPN* [12]. Keduanya menawarkan pendekatan berbeda terhadap *tunneling* dan kriptografi, yang berdampak pada kinerja. Penggunaan *reverse proxy* bersamaan dengan VPN merupakan salah satu pendekatan arsitektur yang banyak diimplementasikan untuk meningkatkan keamanan dan fungsionalitas akses *home server* dari jarak jauh [13].

Penelitian ini mengusulkan pendekatan analisis empiris terhadap kinerja konfigurasi akses *home server* menggunakan kombinasi VPN, dengan membandingkan protokol WireGuard dan OpenVPN serta penerapan *reverse proxy*. Fokus analisis diarahkan pada parameter utama *Quality of Service* (*QoS*), yaitu *response time*, *latency*, dan penggunaan sumber daya seperti *CPU* dan *RAM*. Hasil pengujian akan diperoleh melalui serangkaian skrip pengujian terotomatisasi dan divisualisasikan dalam bentuk grafik metrik, sehingga memungkinkan analisis kinerja secara komprehensif dan akurat berdasarkan data yang telah dikumpulkan.

Penelitian ini menyediakan data kuantitatif perbandingan kinerja antara implementasi *WireGuard* dan *OpenVPN* ketika digunakan bersama *reverse proxy* pada skenario akses *home server*. Dengan menampilkan hasil pengujian secara terstruktur dan menampilkan metrik kinerja dalam bentuk grafik, penelitian ini memberikan gambaran jelas mengenai dampak *overhead* dan interaksi antar komponen. Hasil penelitian ini akan memberikan panduan yang jelas bagi pengguna dalam memilih arsitektur VPN dan protokol yang paling sesuai untuk kebutuhan akses *home server* mereka, mencapai keseimbangan keamanan dan efisiensi. Analisis ini diharapkan dapat membantu mengoptimalkan konfigurasi akses jarak jauh, memastikan pengalaman pengguna yang lebih baik dan pemanfaatan sumber daya sistem yang lebih efisien dibandingkan mengandalkan asumsi teoretis.

II. KAJIAN TEORI

I. *Virtual Private Network (VPN)*

VPN adalah teknologi jaringan yang memungkinkan koneksi aman melalui internet publik, dengan membentuk *tunnel* terenkripsi antara perangkat pengguna dan server. Teknologi ini menjamin kerahasiaan data melalui enkripsi dan otentikasi yang kuat, serta memungkinkan akses ke jaringan internal secara aman [14].

B. WIREGUARD

WireGuard adalah protokol VPN modern yang ringan dan efisien. Dirancang untuk berjalan di level *kernel*, WireGuard menggunakan kriptografi modern seperti ChaCha20 dan BLAKE2s. Dengan basis kode yang minimal dan pendekatan *peer-to-peer*, WireGuard menawarkan kinerja tinggi dan *overhead* rendah, serta cocok untuk arsitektur *multi-core* [14].

C. OPENVPN

OpenVPN adalah protokol VPN berbasis SSL/TLS yang telah lama digunakan secara luas. Dengan dukungan enkripsi kuat seperti AES dan fleksibilitas konfigurasi tinggi, OpenVPN cocok untuk berbagai topologi jaringan. Namun, karena berjalan di *user space,* OpenVPN cenderung memiliki *overhead* lebih tinggi dibanding WireGuard [15].

D. PROTOKOL TCP

Protokol TCP *(Transmission Control Protocol)* adalah protokol komunikasi di layer *transport* yang memfasilitasi transfer data yang andal antara perangkat dalam jaringan Industrial *Control Systems (ICS)*. TCP berfungsi sebagai protokol *connection oriented* yang memastikan setiap komunikasi diawali dengan pembentukan koneksi yang stabil antara pengirim dan penerima .

E. PROTOKOL UDP

User Datagram Protocol (UDP) adalah protokol di lapisan *transport* dalam arsitektur TCP/IP yang memfasilitasi komunikasi menggunakan segmen data berukuran kecil. UDP sering disebut sebagai protokol ringan karena tidak memerlukan proses *handshake* untuk mengenali perangkat lain, sehingga dapat mengurangi *delay* dan memiliki ukuran data yang lebih kecil akibat tidak [16]

F. REVERSE PROXY

Reverse proxy merupakan server perantara yang menerima permintaan dari klien dan meneruskannya ke server *backend*. Teknologi ini meningkatkan keamanan, menyederhanakan manajemen SSL/TLS, serta mendukung *load balancing* dan *caching*. Dalam konteks *home server*, reverse proxy (misalnya Nginx) berperan sebagai titik masuk utama dari internet publik ke jaringan internal [17].

G. *VIRTUAL PRIVATE SERVER*

Virtual Private Network (VPN) merupakan teknologi jaringan yang memungkinkan terciptanya koneksi aman dan terenkripsi melalui server VPN sebagai perantara antara perangkat pengguna dengan internet. Teknologi VPN memanfaatkan infrastruktur jaringan publik yang telah tersedia untuk membentuk sebuah jaringan privat virtual, di mana koneksi privat dapat terbentuk melalui internet publik atau jaringan internal yang dioperasikan oleh penyedia layanan internet. [18]

H. *RASPBERRY PI*

Raspberry Pi adalah komputer papan tunggal berbasis arsitektur ARM yang hemat biaya, dirancang untuk efisiensi energi, kesederhanaan, dan fleksibilitas guna mendukung beragam aplikasi komputasi [19].

I. *HOME SERVER*

Home Server merupakan sistem penyimpanan data terpusat yang beroperasi dalam lingkungan jaringan rumah untuk menyediakan layanan multimedia dan manajemen data secara mandiri. Konsep ini melibatkan

penggunaan perangkat khusus yang terhubung ke jaringan lokal untuk menyimpan, mengorganisir, dan mendistribusikan konten digital kepada berbagai perangkat klien seperti komputer, *smartphone*, dan *smart TV* [20].

J. UBUNTU

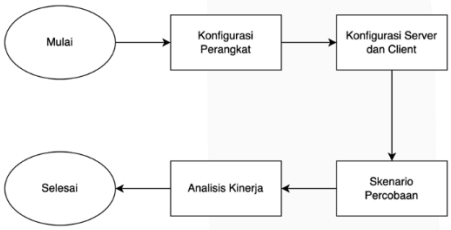
Ubuntu merupakan distribusi sistem operasi Linux yang dikembangkan oleh *Canonical Ltd.* dan bersifat *open source*. Sistem operasi ini dibangun berdasarkan prinsip fundamental Linux yang menawarkan stabilitas, keamanan, dan fleksibilitas dalam pengelolaan infrastruktur jaringan [21].

K. ANALISIS KINERJA

Analisis kinerja jaringan merupakan proses evaluasi sistematis untuk mengukur dan menilai efektivitas serta efisiensi operasional sistem jaringan dalam kondisi tertentu. Konsep ini melibatkan pengukuran berbagai parameter teknis yang mencerminkan kemampuan jaringan dalam menangani beban kerja dan mempertahankan kualitas layanan.

III. METODE

Diagram blok pada Gambar 1 menggambarkan langkah-langkah untuk menganalisis kinerja Wireguard dan OpenVPN pada reverse proxy untuk akses *home server*. Setiap bagian akan di jelaskan di bawah ini.



GAMBAR 1
DIAGRAM BLOK IMPLEMENTASI

A. KONFIGURASI PERANGKAT

Langkah awal dalam penelitian ini adalah menyiapkan infrastruktur fisik dan virtual. Sistem terdiri dari dua unit *Virtual Private Server* (VPS01 dan VPS02) dan dua *Raspberry Pi* (Pi01 dan Pi02) sebagai *home server*. VPS01 dikonfigurasi sebagai server WireGuard dan OpenVPN sekaligus *reverse proxy* menggunakan *Nginx*. VPS02 berperan sebagai klien pengujian. *Raspberry Pi* dikonfigurasi sebagai web server yang menjalankan sistem operasi melalui USB *flash drive*. Kedua unit menggunakan sambungan nirkabel WiFi dan dilengkapi modul *active cooling* untuk menjaga kestabilan suhu saat pengujian berlangsung.

B. Konfigurasi Server dan Client

Selanjutnya dilakukan instalasi dan konfigurasi perangkat lunak pada setiap *node*. WireGuard dan OpenVPN di pasang pada VPS01, termasuk pengaturan jaringan dan otentikasi. *Nginx* dikonfigurasi sebagai reverse proxy untuk meneruskan permintaan HTTP berdasarkan domain. Pi01 menggunakan WireGuard *client*, sementara Pi02 menggunakan OpenVPN client. VPS02 dipasangi berbagai *tool* pengujian seperti curl, ping, iperf3, top, dan *free*.

C. Skenario Percobaan

TABEL 1
SKENARIO PERCOBAAN

Skenario	*VPN*	*Reverse Proxy*	Target	Metrik
1	WireGuard	Tidak	Pi01 *direct*	*Response time, latency, resource usage*
2	WireGuard	Ya	Pi01 via VPS01	*Response time, latency, resource usage*
3	OpenVPN	Tidak	Pi02 *direct*	*Response time, latency, resource usage*
4	OpenVPN	Ya	Pi02 via VPS01	*Response time, latency, resource usage*

Penelitian ini menguji performa WireGuard dan OpenVPN dalam empat skenario: dengan dan tanpa reverse proxy. Pengujian dilakukan dari VPS02 ke dua *Raspberry Pi* dengan mengukur *response time, latency*, serta penggunaan CPU dan RAM.

D. Analisis Kinerja

Analisis kinerja dalam penelitian ini mengacu pada empat metrik utama yang tercantum dalam Tabel 2 Metrik Analisis Kinerja, yaitu *response time, latency,* penggunaan CPU, dan penggunaan RAM.

TABEL 2
METRIK ANALISIS KINERJA

Metrik	Tools Pengujian	Deskripsi	Command Pengujian
Response Time	curl	Waktu yang diperlukan server untuk menanggapi permintaan HTTP	curl -s -o /dev/null -w '%{time_total}' "$target_url"
Latency	ping	Waktu tempuh data dari sumber ke tujuan	ping -c 20 -i 0.2 "$target_ip"
CPU	top	Penggunaan prosesor dalam persentase	top -bn1 | grep "^%Cpu" | awk "{print 100-$8}"

IV. HASIL DAN PEMBAHASAN

A. Skenario Percobaan

Pengujian kinerja sistem dilaksanakan pada lingkungan terkendali menggunakan layanan internet *service provider Biznet* dengan perute Huawei EG8145V5 pada mode 802.11a/n/ac berkapasitas 100Mbps. Pengambilan data dilakukan pada rentang

waktu pukul 3 pagi hingga 5 pagi untuk mereduksi interferensi lalu lintas jaringan. Arsitektur pengujian melibatkan empat unit utama VPS01 berperan sebagai peladen VPN dan reverse proxy, VPS02 sebagai titik asal pengujian, Pi01 sebagai klien WireGuard, dan Pi02 sebagai klien OpenVPN. Penyiapan arsitektur ini mencakup konfigurasi antarmuka pada setiap unit VPS01 menggunakan eth0 dengan IP publik 20.187.146.108, wg0 dengan IP 10.10.10.1, dan tun0 dengan IP 10.10.20.1. VPS02 menggunakan eth0 dengan IP publik 104.214.172.3 dan tun0 dengan IP 10.10.20.2. Pi01 terhubung melalui antarmuka wlan0 dan memiliki antarmuka wg0 dengan IP 10.10.10.2. Pi02 terhubung melalui antarmuka wlan0 dan memiliki antarmuka tun0 dengan IP 10.10.20.3. Langkah perangkaian diawali dengan pengaturan WireGuard pada VPS01 dan Pi01, dilanjutkan konfigurasi *ProxyPass* pada VPS01, serta penyiapan OpenVPN pada VPS01, VPS02, dan Pi02. Verifikasi koneksi dilakukan melalui perintah *ping* dan *traceroute*.

1. *WireGuard tanpa Revers Proxy*

Pada skenario pertama, dilakukan evaluasi kinerja konektivitas langsung menggunakan protokol WireGuard tanpa implementasi reverse proxy. Pengujian kinerja ini dieksekusi dari sisi vps2, dengan menargetkan *endpoint* spesifik berupa alamat IP WireGuard dari klien pi1, yakni 10.10.10.2. Serangkaian metrik kinerja fundamental diukur secara sistematis menggunakan *tools* yang telah ditentukan secara terbatas. Pengukuran *Response Time* dilaksanakan memanfaatkan utilitas *curl*, bertujuan untuk mengidentifikasi durasi yang dibutuhkan server dalam merespons permintaan HTTP. Evaluasi *Latency* dilakukan melalui eksekusi perintah *ping*, guna mengukur waktu tempuh paket data dari sumber ke tujuan serta persentase kehilangan paket. Selain itu, pemantauan penggunaan sumber daya sistem pada sisi server juga dicakup, meliputi penggunaan CPU yang diukur menggunakan top dan penggunaan RAM yang diukur menggunakan *free*. Seluruh data hasil pengukuran dari skenario ini didokumentasikan dan berfungsi sebagai *baseline* untuk perbandingan pada skenario-skenario pengujian berikutnya.

2. WireGuard dengan Revers Proxy

Pada skenario kedua, evaluasi kinerja konektivitas dilaksanakan menggunakan protokol WireGuard yang diintegrasikan dengan implementasi reverse proxy. Pengujian kinerja ini dilakukan dari sisi vps2, dengan menargetkan *endpoint proxy_pass* yang spesifik, yaitu nama domain pi1.hipsterweeds.my.id. Serangkaian metrik kinerja diukur secara sistematis menggunakan perangkat yang telah ditentukan secara terbatas. Pengukuran *Response Time* dilakukan menggunakan utilitas *curl*, yang bertujuan untuk menentukan waktu yang dibutuhkan server dalam menanggapi permintaan HTTP melalui jalur reverse proxy. Evaluasi *Latency* dilakukan melalui eksekusi perintah *ping*, guna mengukur waktu tempuh paket data dari sumber ke tujuan serta persentase kehilangan paket pada jalur yang sama. Selain itu, pemantauan penggunaan sumber daya sistem pada sisi server juga dicakup, meliputi penggunaan CPU yang diukur menggunakan utilitas top dan penggunaan RAM yang diukur menggunakan utilitas free. Seluruh data hasil pengukuran dari skenario ini didokumentasikan dan berfungsi sebagai data untuk perbandingan pada skenario pengujian berikutnya.

3. OpenVPN tanpa Reverse Proxy

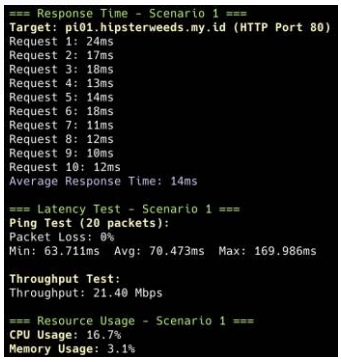
Skenario ketiga berfokus pada pengukuran performa koneksi OpenVPN tanpa menggunakan reverse proxy. Pengujian ini dilaksanakan dengan menargetkan secara langsung alamat IP klien PI2, yaitu 10.10.20.3, dari vps2. Metodologi pengukuran yang diterapkan mencakup serangkaian pengujian standar untuk mengukur aspek kinerja jaringan dan utilitas sumber daya: pengukuran *response time* menggunakan alat curl, pengukuran *latency* melalui pengiriman 20 paket ping, serta pemantauan utilitas CPU dan RAM menggunakan perintah top dan *free*. Data performa yang diperoleh dari skenario koneksi OpenVPN *direct* ini berperan sebagai basis komparasi terhadap hasil pengujian skenario WireGuard *direct* dan OpenVPN yang diakses melalui reverse proxy.

4. OpenVPN dengan Reverse Proxy

Pada skenario terakhir, pengujian performa dilakukan pada konfigurasi OpenVPN yang diintegrasikan dengan reverse proxy. Alur pengujian dimulai dari VPS2, dilanjutkan melalui reverse proxy pada VPS01, terhubung melalui protokol OpenVPN ke PI02, dengan target *endpoint* akses pada http://pi2.hipsterweeds.my.id. Setelah dipastikan bahwa jalur komunikasi dan fungsi reverse proxy telah beroperasi dengan optimal, serangkaian pengujian kinerja dijalankan secara berurutan. Pengujian meliputi pengukuran *response time* menggunakan curl, *latency* menggunakan ping, serta pemantauan penggunaan sumber daya sistem yaitu CPU menggunakan top dan RAM menggunakan free. Hasil yang diperoleh dari skenario ini bertujuan untuk mengevaluasi pengaruh kumulatif implementasi reverse proxy terhadap performa koneksi OpenVPN, serta untuk menyediakan basis komparasi dengan hasil pengujian pada skenario sebelumnya yang tidak melibatkan reverse proxy.

B. Hasil Percobaan Skenario

1. Hasil Percobaan WireGuard tanpa Reverse Proxy



GAMBAR 2
HASIL SKENARIO PERTAMA

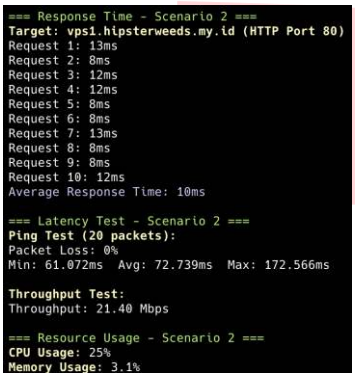
Berdasarkan Gambar 4.1 implementasi WireGuard tanpa reverse proxy menghasilkan *response time* sebesar 14*ms* dengan *latency* rata-rata 70,473*ms*. Penggunaan sumber daya sistem mencatat konsumsi *CPU* sebesar 16,7% dan *RAM* 3,1% pada server Pi01. Nilai *throughput*

yang terukur menunjukkan performa jaringan stabil dengan karakteristik koneksi langsung yang minim *overhead*. Hasil ini merepresentasikan *baseline* performa WireGuard dalam konfigurasi sederhana tanpa lapisan tambahan, memvalidasi efisiensi protokol tersebut dalam menangani komunikasi *point-to-point*.

TABEL 3
HASIL SKENARIO PERTAMA

Metrik	Nilai
Response Time	14*ms*
Latency	70,473*ms*
CPU Usage	16,7%
RAM Usage	3,1%

2. Hasil Percobaan WireGuard dengan Reverse Proxy



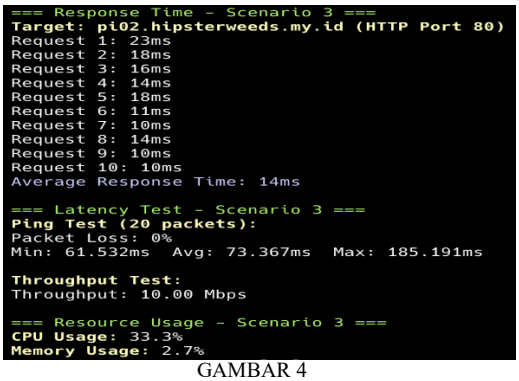
GAMBAR 3
HASIL SKENARIO KEDUA

Berdasarkan Gambar 4.2 penerapan reverse proxy pada WireGuard menurunkan *response time* menjadi 10*ms* meskipun *latency* meningkat menjadi 72,739*ms*. Konsumsi *CPU* meningkat signifikan ke 25% sementara penggunaan *RAM* tetap stabil di 3,1%. Penurunan *response time* mengindikasikan optimasi proses *routing* melalui VPS01, sedangkan peningkatan *latency* disebabkan oleh penambahan hop jaringan melalui reverse proxy. Data ini membuktikan bahwa reverse proxy mampu meningkatkan kecepatan respon aplikasi web meskipun berdampak pada latensi jaringan dan beban komputasi server.

TABEL 4
HASIL SKENARIO KEDUA

Metrik	Nilai
Response Time	10*ms*
Latency	72,739*ms*
CPU Usage	25%
RAM Usage	3,1%

3. Hasil Percobaan OpenVPN tanpa Reverse Proxy



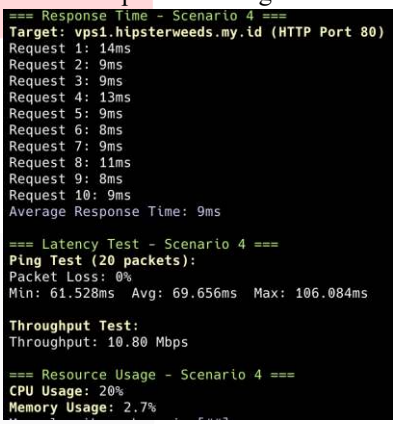
GAMBAR 4
HASIL SKENARIO KETIGA

Berdasarkan Gambar 4.3 konfigurasi OpenVPN langsung mencatat *response time* 14ms dengan *latency* tertinggi di antara semua skenario (73,367*ms*). Konsumsi sumber daya sistem lebih tinggi dengan penggunaan *CPU* 33,3% dan *RAM* 2,7%, mencerminkan kompleksitas enkripsi *SSL/TLS* yang melekat pada protokol ini. Hasil ini mengonfirmasi karakteristik OpenVPN yang lebih intensif dalam penggunaan *CPU* dibandingkan WireGuard, meskipun tetap mampu mempertahankan *response time* setara dengan WireGuard *direct*.

TABEL 5
HASIL SKENARIO KETIGA

Metrik	Nilai
Response Time	14*ms*
Latency	73,367*ms*
CPU Usage	33,3%
RAM Usage	2,7%

4. Hasil Percobaan OpenVPN dengan Reverse Proxy



GAMBAR 5
HASIL SKENARIO KEEMPAT

Berdasarkan Gambar 4.4 kombinasi OpenVPN dan reverse proxy menghasilkan *response time* terbaik (9*ms*) dengan *latency* 69,656*ms* yang lebih rendah dibandingkan skenario OpenVPN *direct*. Konsumsi *CPU* turun signifikan menjadi 20% sementara penggunaan *RAM* tetap 2,7%. Penurunan *CPU usage* mengindikasikan optimalisasi proses enkripsi melalui distribusi beban antara server VPN dan reverse proxy. Hasil ini menunjukkan sinergi positif antara OpenVPN dan reverse proxy dalam meningkatkan kecepatan respons sekaligus mengurangi beban komputasi.

TABEL 6
HASIL SKENARIO KEEMPAT

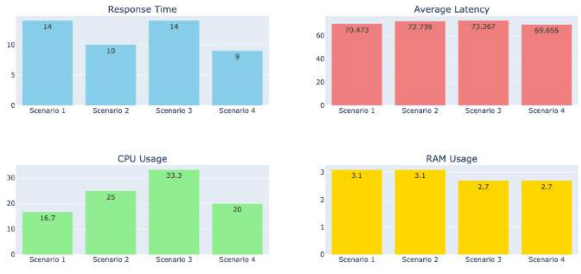
Metrik	Nilai
Response Time	9*ms*
Latency	69,656*ms*
CPU Usage	20%
RAM Usage	2,7%

C. Analisis Kinerja

Berdasarkan hasil eksperimen keempat skenario, teridentifikasi variasi kinerja yang signifikan akibat interaksi antara protokol VPN dan keberadaan reverse proxy. Konfigurasi OpenVPN dengan reverse proxy mencatat respons time tercepat sebesar 9*ms* disertai penurunan *latency* sebesar 3,711ms dibandingkan

konfigurasi langsung, menjadikannya pilihan optimal untuk aplikasi web yang mengutamakan kecepatan respon. Namun, keunggulan ini diimbangi dengan peningkatan kompleksitas arsitektur dan ketergantungan pada kinerja VPS sebagai perantara, yang berpotensi menimbulkan *single point of failure*. Pada skenario WireGuard tanpa reverse proxy, efisiensi sumber daya menjadi keunggulan utama dengan penggunaan CPU hanya 16,7% dan *latency* 70,473*ms*, menjadikannya ideal untuk lingkungan komputasi terbatas seperti perangkat *embedded*. Meskipun demikian, *response time* 14*ms* yang lebih tinggi dibandingkan konfigurasi dengan reverse proxy serta ketiadaan fitur keamanan tambahan menjadi kelemahan utameanya. Di sisi lain, implementasi WireGuard dengan reverse proxy berhasil mengurangi *response time* sebesar 28,57% menjadi 10*ms* melalui optimasi *routing*, meskipun diiringi peningkatan *latency* 3,26% (72,739*ms*) dan lonjakan penggunaan CPU hingga 25% akibat beban pemrosesan tambahan pada VPS.

Skenario OpenVPN tanpa Reverse Proxy menunjukkan stabilitas *response time* 14*ms* yang setara dengan WireGuard *direct*, meskipun menggunakan enkripsi lebih kompleks. Namun, konsumsi CPU tertinggi (33,3%) dan *latency* maksimal 73,367*ms* membatasi kelayakannya untuk *deployment* jangka panjang pada perangkat rendah daya. Adapun OpenVPN dengan reverse proxy tidak hanya mencapai *response time* terbaik 9*ms* tetapi juga mengurangi beban CPU sebesar 39,9% melalui distribusi tugas enkripsi antara server VPN dan reverse proxy, meskipun memerlukan infrastruktur dua lapis yang meningkatkan biaya pemeliharaan.



GAMBAR 6
GRAFIK ANALISIS KINERJA

Pemilihan konfigurasi optimal bergantung pada prioritas penggunaan: OpenVPN dengan reverse proxy unggul untuk aplikasi kritikal yang membutuhkan kecepatan respons tinggi, sedangkan WireGuard *direct* lebih sesuai untuk sistem dengan sumber daya terbatas. WireGuard dengan reverse proxy menawarkan keseimbangan antara respons time 10ms dan konsumsi CPU moderat 25%, sementara OpenVPN *direct* cocok untuk skenario yang memerlukan stabilitas *response time* dengan toleransi terhadap penggunaan sumber daya tinggi. Temuan ini mempertegas bahwa integrasi reverse proxy tidak bersifat universal, melainkan memberikan dampak diferensial tergantung karakteristik protokol VPN yang digunakan, dengan *trade off* antara optimasi performa dan kompleksitas infrastruktur sebagai faktor penentu utamanya. Berdasarkan data rata-rata dan standar deviasi dari keempat skenario, kami menetapkan target kinerja sebagai berikut.

TABEL 7
RESEPONSE TIMES

Skenario	*Device*	CPU (%)	RAM (%)	*Response Time* (*ms*)	*Latency* (*ms*)
WireGuard *Direct*	Pi 01	16.7	3.1	14	70.473
WireGuard + Reverse Proxy	Pi 01	25.0	3.1	10	72.739
OpenVPN *Direct*	Pi 02	33.3	2.7	14	73.367
OpenVPN + Reverse Proxy	PI 02	20.0	2.7	9	69.656

Response Time ≤ 14*ms* memastikan waktu respon HTTP untuk domain *web server* tetap cepat baik *direct* maupun dengan reverse proxy. *Latency* ≤ 73*ms* bertujuan menghindari penambahan hop jaringan melebihi batas toleransi *delay*. *CPU Usage Peak* ≤ 30% menjamin *overhead* enkripsi dan proxy tidak membebani prosesor lebih dari ambang wajar. *RAM Usage* Maksimum ≤ 3,1% menjaga penggunaan memori tetap minimal agar *web service* server tidak terganggu.

TABEL 8
AVARAGE DAN STANDAR DEVIASI

Metrik	*Average*	Standar Deviasi	Target Kinerja
Response Time	11,75*ms*	± 2,28*ms*	≤ μ + σ = 14*ms*
Latency	71,56*ms*	±1,54*ms*	≤ μ + σ = 73*ms*
CPU Usage	23,75%	± 6,26%	*Peak* ≤ μ + σ = 30%
RAM Usage	2,9%	± 0,20%	*Baseline* ≥ μ − σ = 17%

Tabel ini menyajikan data kuantitatif yang mencakup beberapa metrik kinerja, disertai dengan nilai rata-rata dan standar deviasi yang diukur. Simbol `±` (plus-minus) digunakan untuk menunjukkan rentang variabilitas atau standar deviasi dari nilai rata-rata yang diamati. Pada tabel, angka desimal menggunakan koma (,) sebagai pemisah untuk merepresentasikan bagian pecahan dari nilai numerik. Untuk menetapkan kriteria atau batasan yang harus dipenuhi dalam target kinerja, simbol pertidaksamaan `≤` (kurang dari atau sama dengan) dan `≥` (lebih dari atau sama dengan) digunakan. Dalam formula target kinerja, simbol `μ` melambangkan nilai rata-rata (*mean*) dan `σ` melambangkan nilai standar deviasi, yang kemudian dihitung dan disamakan dengan nilai numerik target yang ditetapkan, sebagaimana ditunjukkan oleh penggunaan tanda sama dengan (=).

V. KESIMPULAN

Berdasarkan pengujian terhadap empat skenario akses *home server* implementasi reverse proxy Nginx secara konsisten menurunkan *response time* kedua protokol VPN. WireGuard mengalami penurunan dari 14*ms* menjadi 10*ms* atau 28.6%. OpenVPN menurun dari 14*ms* menjadi 9*ms* atau 35.7%. Hasil ini bersamaan

dengan pengaruh pada *latency*. *Latency* WireGuard meningkat dari 70.473*ms* menjadi 72.739*ms* atau 3.26%. Sementara itu *latency* OpenVPN menurun dari 73.367ms menjadi 69.656*ms* atau 5.05%. Dari sisi penggunaan sumber daya metrik menunjukkan perbedaan signifikan. Konfigurasi OpenVPN dengan reverse proxy mencatat beban CPU 20% dan penggunaan RAM 2.6%. Sedangkan konfigurasi WireGuard dengan reverse proxy menunjukkan beban CPU tertinggi 25% dengan penggunaan RAM stabil 3.0%. Perbedaan tersebut mengindikasikan bahwa meskipun reverse proxy menambah lapisan *forwarding* juga berkontribusi pada optimalisasi *routing* dan distribusi beban komputasi.

Secara keseluruhan implementasi OpenVPN dengan *reverse proxy* menunjukkan kinerja optimal ditinjau dari *response time* tercepat *latency* terendah serta efisiensi CPU dan RAM terbaik. Sementara itu implementasi WireGuard tanpa proxy unggul dalam efisiensi CPU pada skenario koneksi *point to point*. Temuan ini menunjukkan bahwa optimalisasi kinerja VPN melalui reverse proxy bergantung pada karakteristik protokol yang digunakan yakni arsitektur berlapis lebih menguntungkan OpenVPN sedangkan WireGuard lebih sesuai untuk implementasi sederhana tanpa reverse proxy.

1. Pengaruh Reverse Proxy terhadap Kinerja Akses *Home Server* melalui Protokol VPN

Implementasi reverse proxy berkontribusi signifikan pada parameter *response time* kedua protokol VPN yang dievaluasi. Penggunaan reverse proxy Nginx menunjukkan penurunan *response time* sebesar 28,6% pada protokol WireGuard dari 14*ms* menjadi 10*ms* serta 35,7% pada protokol OpenVPN dari 14*ms* menjadi 9*ms*. Meskipun penambahan lapisan reverse proxy mengakibatkan peningkatan *latency* sebesar 3,26% pada WireGuard dari 70,473*ms* menjadi 72,739*ms* sementara OpenVPN mengalami penurunan *latency* sebesar 5,1% dari 73,367ms menjadi 69,656*ms* hasil pengujian mengindikasikan bahwa reverse proxy tidak hanya berfungsi sebagai penerus koneksi melainkan turut berperan dalam optimalisasi *routing* dan pengelolaan koneksi jaringan. Penggunaan sumber daya CPU bervariasi WireGuard menunjukkan kenaikan dari 16,7% menjadi 25% sedangkan OpenVPN mengalami penurunan dari 33,3% menjadi 20% hal ini mengindikasikan distribusi beban komputasi yang lebih efisien melalui arsitektur berlapis.

2. . Protokol VPN Optimal untuk Implementasi *Home Server* dengan Arsitektur Reverse Proxy

Berdasarkan evaluasi komprehensif terhadap parameter *Quality of Service* OpenVPN dengan implementasi reverse proxy menunjukkan kinerja optimal untuk skenario *home server*. Dari konfigurasi ini tercatat *response time* tercepat 9*ms latency* terendah 69,656*ms* serta efisiensi penggunaan CPU 20% dan RAM 2,7%. Kinerja optimal OpenVPN dalam arsitektur reverse proxy didukung oleh kemampuannya mendistribusikan beban enkripsi *SSL/TLS* antara server VPN dan reverse proxy yang menghasilkan penurunan *overhead* CPU signifikan dibandingkan konfigurasi tanpa reverse proxy. Sebaliknya WireGuard menunjukkan performa optimal pada konfigurasi koneksi langsung tanpa reverse proxy dengan efisiensi CPU tertinggi 16,7% namun mengalami degradasi kinerja saat diintegrasikan dengan reverse proxy. Hal ini mengindikasikan bahwa karakteristik protokol WireGuard lebih sesuai untuk implementasi *point-to-point* yang lebih sederhana.

3. Perbandingan Efisiensi Penggunaan Sumber Daya CPU dan RAM antara WireGuard dan OpenVPN dengan Reverse Proxy

Analisis efisiensi sumber daya sistem mengungkap perbedaan karakteristik signifikan antara kedua protokol VPN saat diintegrasikan dengan reverse proxy. WireGuard dalam kombinasi dengan reverse proxy menunjukkan konsumsi CPU 25% dan RAM 3.1%. Tingkat penggunaan sumber daya relatif tinggi ini mencerminkan *overhead* substansial akibat kompleksitas integrasi dengan reverse proxy. Kontras dengan OpenVPN pada konfigurasi serupa dengan reverse proxy menunjukkan efisiensi superior dengan konsumsi CPU 20% dan RAM 2.7%. Temuan ini mengindikasikan optimalisasi yang lebih baik dalam pengelolaan sumber daya komputasi pada OpenVPN. Stabilitas penggunaan RAM pada kedua protokol menunjukkan konsistensi dengan variasi minimal sebesar 0.4% yang secara operasional dianggap tidak signifikan. Secara keseluruhan temuan ini mengonfirmasi bahwa meskipun WireGuard dikenal sebagai protokol *lightweight* implementasinya dengan reverse proxy menghasilkan *overhead* CPU yang lebih tinggi dibandingkan OpenVPN. Implikasi temuan ini adalah efisiensi dasar suatu protokol tidak selalu berkorelasi linear dengan efisiensi dalam arsitektur implementasi yang kompleks.

REFERENSI

[1] B. W. Aulia, M. Rizki, P. Prindiyana, and S. Surgana, “Peran Krusial Jaringan Komputer dan Basis Data dalam Era Digital,” *JUSTINFO | Jurnal Sistem Informasi dan Teknologi Informasi*, vol. 1, no. 1, pp. 9–20, 2023, doi: 10.33197/justinfo.vol1.iss1.2023.1253.

[2] B. W. Aulia, M. Rizki, P. Prindiyana, and S. Surgana, “Peran Krusial Jaringan Komputer dan Basis Data dalam Era Digital,” *JUSTINFO | Jurnal Sistem Informasi dan Teknologi Informasi*, vol. 1, no. 1, pp. 9–20, Dec. 2023, doi: 10.33197/justinfo.vol1.iss1.2023.1253.

[3] A. K. M. B. Haque, B. Bhushan, and G. Dhiman, “Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends,” *Expert Syst*, vol. 39, no. 5, Jun. 2022, doi: 10.1111/exsy.12753.

[4] I. Meijers, “Two-Way Quality of Service Policy Enforcement Methods in Dynamically Formed Overlay Virtual Private Networks,” in *2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, IEEE, Oct. 2023, pp. 1–4. doi: 10.1109/ITMS59786.2023.10317738.

[5] S. Budiyanto and D. Gunawan, “Comparative Analysis of VPN Protocols at Layer 2 Focusing on Voice Over Internet Protocol,” *IEEE Access*, vol. 11, pp. 60853–60865, 2023, doi: 10.1109/ACCESS.2023.3286032.

[6] A. Esseghir, F. Kamoun, and O. Hraiech, “AKER: An open-source security platform integrating IDS and SIEM functions with encrypted traffic analytic capability,” *Journal of Cyber Security Technology*, vol. 6, no. 1–2, pp. 27–64, Apr. 2022, doi: 10.1080/23742917.2022.2058836.

[7] P. Centobelli, R. Cerchione, P. Del Vecchio, E. Oropallo, and G. Secundo, "Blockchain technology for bridging trust, traceability and transparency in circular supply chain," *Information & Management*, vol. 59, no. 7, p. 103508, Nov. 2022, doi: 10.1016/j.im.2021.103508.

[8] H. Abbas *et al.*, "Security Assessment and Evaluation of VPNs: A Comprehensive Survey," *ACM Comput Surv*, vol. 55, no. 13s, pp. 1–47, Dec. 2023, doi: 10.1145/3579162.

[9] R. Ibrahim, I. Khider, S. Edam, and T. Mukhtar, "Comprehensive Strategies for Enhancing SD-WAN: Integrating Security, Dynamic Routing and Quality of Service Management," *IET Networks*, vol. 14, no. 1, p. 1, Jan. 2025, doi: 10.1049/ntw2.70007.

[10] M. Shehab and L. R. Alzabin, "Evaluating the Effectiveness of Stealth Protocols and Proxying in Hiding VPN Usage," *Journal of Computational and Cognitive Engineering*, Sep. 2024, doi: 10.47852/bonviewJCCE42023642.

[11] A. V. Ostroukh, C. B. Pronin, A. A. Podberezkin, J. V. Podberezkina, and A. M. Volkov, "Enhancing Corporate Network Security and Performance: A Comprehensive Evaluation of WireGuard as a Next-Generation VPN Solution," in *2024 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, IEEE, Jul. 2024, pp. 1–5. doi: 10.1109/SYNCHROINFO61835.2024.10617501.

[12] K. Ghanem, S. Ugwuanyi, J. Hansawangkit, R. McPherson, R. Khan, and J. Irvine, "Security vs Bandwidth: Performance Analysis Between IPsec and OpenVPN in Smart Grid," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, Jul. 2022, pp. 1–5. doi: 10.1109/ISNCC55209.2022.9851717.

[13] L. Alevizos, V. T. Ta, and M. Hashem Eiza, "Augmenting zero trust architecture to endpoints using blockchain: A <scp>state-of-the-art</scp> review," *SECURITY AND PRIVACY*, vol. 5, no. 1, pp. 1–27, Jan. 2022, doi: 10.1002/spy2.191.

[14] R. Hermawan and Y. M. Saputra, "Analisis Perbandingan Penggunaan Metode Tunneling Cloud Virtual Private Network dan WireGuard Virtual Private Network pada Implementasi Infrastruktur Hybrid Cloud," *Journal of Internet and Software Engineering*, vol. 6, no. 1, pp. 1–12, 2025.

[15] "Торайғыров университетінің хабаршысы," vol. 1, 2023.

[16] M. H. H. Ichsan, R. Maulana, and O. M. W. Wardhana, "UDP Pervasive Protocol Design and Implementation on Multi Devices using MyRIO," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 2, p. 307, Jul. 2022, doi: 10.26555/jiteki.v8i2.23835.

[17] L. de S. Oliveira, J. P. C. de Sousa, and J. V. A. Ribeiro, "Bypassing Cloudflare's reverse proxy: a case study / Contornando o proxy reverso do Cloudflare: um estudo de caso," *Brazilian Journal of Development*, vol. 8, no. 4, pp. 27250–27259, Apr. 2022, doi: 10.34117/bjdv8n4-298.

[18] E. Suhadi and T. Arifin, "RANCANGAN VIRTUAL PRIVATE NETWORK PADA KANTOR PROLOV MENGGUNAKAN ZEROTIER," *JIKA (Jurnal Informatika)*, vol. 8, no. 1, p. 66, Jan. 2024, doi: 10.31000/jika.v8i1.9979.

[19] Z. D. Zhang *et al.*, "TopADDPi: An Affordable and Sustainable Raspberry Pi Cluster for Parallel-Computing Topology Optimization," *Processes*, vol. 13, no. 3, Mar. 2025, doi: 10.3390/pr13030633.

[20] M. Faiz Khan, B. Hazela, D. Pandey, K. K. Singh, and S. Singh, "International Journal of Telecommunications & Emerging Technologies Design of a Home Server Employing PCIe," 2024, doi: 10.37628/IJTET.

[21] Rakhmadi Rahman, Awal Ramadhan Nasrun, and Adinda Aulia Rahmi, "Desain dan Implementasi Sistem Operasi Linux Ubuntu Versi 22.04 untuk Perlindungan Data dari Serangan Komputasi Kuantum," *Bridge : Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 3, pp. 207–213, Jul. 2024, doi: 10.62951/bridge.v2i3.159.