ANALISIS KINERJA NGINX DAN APACHE SEBAGAI *REVERSE* PROXY DENGAN *OPEN*VPN UNTUK AKSES *HOME* SERVER

1st Habibullah Al Qomar
Fakultas Informatika
Telkom University Purwokerto
Purwokerto, Jawa Tengah
habibqr@student.telkomuniversity.ac.id

2nd Alon Jala Tirta Segara, S.Kom., M.Kom.
Fakultas Informatika
Telkom University Purwokerto
Purwokerto, Jawa Tengah
@telkomuniversity.ac.id

***Abstrak* — Kebutuhan akan akses jarak jauh yang aman terhadap layanan dan data pribadi mendorong penggunaan solusi seperti *reverse* proxy dan *Virtual Private Network* (VPN) dalam pengelolaan *home* server. Penelitian ini menganalisis kinerja dua web server populer, NGINX dan Apache, sebagai *reverse* proxy yang diintegrasikan dengan *Open*VPN untuk akses *home* server, dengan fokus pada performa di lingkungan terbatas seperti Virtual Private Server (VPS) dan Raspberry Pi. Empat skenario pengujian dirancang untuk mengukur dan membandingkan *response time*, *latency*, serta konsumsi CPU dan RAM: NGINX dan Apache sebagai *reverse* proxy dengan *Open*VPN, serta *direct access* tanpa *reverse* proxy pada masing-masing platform. Pengukuran dilakukan menggunakan curl, ping, top, dan free untuk mendapatkan data empiris yang valid. Hasil evaluasi menunjukkan bahwa Apache *reverse* proxy dengan *Open*VPN menghasilkan *response time* tercepat (0,1062 detik) namun dengan konsumsi CPU lebih tinggi (6,1%), sedangkan NGINX *reverse* proxy menawarkan efisiensi penggunaan *resource* (CPU 3,1%, RAM 8,3%) dengan *response time* yang konsisten. Skenario *direct access* mengindikasikan trade-off antara efisiensi dan stabilitas akses. Temuan ini memberikan rekomendasi empiris terkait pemilihan *reverse* proxy optimal pada lingkungan *home* server berbasis VPN, serta menjadi referensi praktis bagi administrator jaringan dalam mengonfigurasi dan mengoptimalkan infrastruktur akses jarak jauh yang efisien dan aman.**

***Kata kunci*— Apache, *home* server, *latency*, Nginx, *Open*VPN, *response time, resource usage*.**

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. Fenomena ini mendorong peningkatan adopsi solusi yang memungkinkan pengguna untuk terhubung ke jaringan pribadi dari jarak jauh, 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 [1].

Akses jarak jauh yang aman ke jaringan pribadi menjadikan *Virtual Private Network* (VPN) sebagai solusi yang mapan untuk menciptakan kanal komunikasi yang aman melalui infrastruktur publik seperti internet. Dengan teknik enkripsi dan tunneling, VPN menjamin kerahasiaan, integritas, dan otentikasi data yang ditransmisikan. Selain VPN, *reverse* proxy juga semakin populer sebagai lapisan di depan *home* server untuk mengelola koneksi masuk, memberikan fitur keamanan tambahan seperti terminasi SSL/TLS, otentikasi, dan load balancing. Penggunaan *reverse* proxy juga menyederhanakan akses ke berbagai layanan internal melalui satu titik masuk publik [2].

Solusi untuk mengatasi masalah kinerja potensial melibatkan pemilihan jenis *reverse* proxy yang efisien dan konfigurasi sistem yang tepat. Dua jenis *reverse* proxy yang saat ini banyak digunakan dan diakui karena efisiensi dan fleksibilitas berbeda adalah Nginx dan Apache. Nginx menggunakan arsitektur *event-driven* yang dirancang untuk menangani ribuan koneksi simultan dengan konsumsi *resource* minimal, sementara Apache menerapkan model multi-process yang memberikan stabilitas tinggi namun dengan *overhead* memori yang lebih besar. Penggunaan *reverse* proxy ganda dengan karakteristik berbeda bersamaan dengan protokol *Open*VPN merupakan salah satu pendekatan arsitektur yang layak diimplementasikan untuk meningkatkan keamanan dan fungsionalitas akses *home* server dari jarak jauh [3].

Penelitian ini mengusulkan pendekatan analisis empiris terhadap kinerja konfigurasi akses *home* server menggunakan kombinasi *Open*VPN dengan membandingkan implementasi *reverse* proxy Nginx dan Apache dalam lingkungan yang terkontrol. Fokus analisis diarahkan pada parameter utama (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 empiris yang telah dikumpulkan [4].

Penelitian ini menyediakan data kuantitatif perbandingan kinerja antara implementasi *reverse* proxy Nginx dan Apache ketika digunakan dalam lingkungan *Open*VPN pada skenario akses *home* server. Dengan menampilkan hasil pengujian secara terstruktur dan menampilkan metrik kinerja dalam bentuk tabel, penelitian ini memberikan gambaran jelas mengenai dampak *overhead* dan interaksi antar komponen dalam arsitektur multi-tier *reverse* proxy. Hasil penelitian ini akan memberikan panduan yang jelas bagi pengguna dalam memilih jenis *reverse* proxy yang paling sesuai untuk kebutuhan akses *home* server mencapai keseimbangan. 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

Bagian ini menguraikan berbagai teori yang berhubungan dengan variabel penelitian, yang menjadi landasan utama dalam proses pengembangan sistem. Berikut adalah teori-teori yang relevan:

A. *Reverse* Proxy

Reverse proxy berfungsi menyeimbangkan beban trafik dengan meneruskan permintaan klien ke beberapa server *backend* serta menyediakan *caching* untuk meningkatkan performa web server. Kelebihannya mencakup peningkatan performa, stabilitas, fitur lengkap, dan efisiensi sumber daya, seperti pada Nginx. *Reverse* proxy juga mengurangi beban server dengan menyimpan data halaman dalam memori. Namun, kekurangannya bisa muncul dari ketergantungan pada konfigurasi yang tepat dan risiko jika koneksi internet tidak stabil [5].

Nginx adalah server yang mengalihkan permintaan pengguna ke server lain, mendistribusikan lalu lintas, dan mengoptimalkan beban server untuk meningkatkan kecepatan dan pengalaman pengguna. Apache adalah server web yang menyajikan halaman web kepada pengguna, memungkinkan mereka mengakses konten seperti teks, gambar, dan video. Selain itu, Apache juga dapat berfungsi sebagai perantara (*reverse* proxy), memungkinkan akses ke banyak layanan melalui satu titik masuk, yang meningkatkan kecepatan dan keamanan [6].

B. Nginx

Nginx adalah web server efisien buatan Igor Sysoev (2004) dengan arsitektur *event-driven*, mampu menangani ribuan koneksi dengan memori minimal. Dalam *reverse* proxy berbasis *Open*VPN, Nginx di Pi01 bertugas menerima permintaan HTTP/HTTPS lewat tunnel terenkripsi, lalu meneruskannya ke *backend* via proxy_pass sambil mempertahankan info client dengan X-Real-IP dan X-*Forwarded*-For. Arsitekturnya yang ringan cocok untuk perangkat terbatas seperti Raspberry Pi, tidak seperti Apache yang memakai model multi-process [7].

Konfigurasi Nginx meliputi upstream, SSL termination, dan load balancing, yang membantu menekan *latency* dan *response time* dalam tunnel VPN. Keunggulannya ada pada efisiensi CPU/RAM dan dukungan HTTP/2 serta WebSocket untuk performa web modern yang optimal [8][9].

C. *Apache*

Apache HTTP Server adalah web server *open source* yang dikembangkan sejak 1995 oleh Apache Software *Foundation*, terkenal dengan arsitektur modular yang fleksibel. Dalam *reverse* proxy berbasis *Open*VPN, Apache di Pi02 bertindak sebagai perantara permintaan HTTP/HTTPS dari client melalui tunnel terenkripsi, lalu meneruskannya ke *backend* menggunakan modul mod_proxy dengan direktif ProxyPass dan ProxyPassReverse. Arsitektur multi-process (MPM) memungkinkan setiap proses bekerja mandiri, memberi stabilitas tinggi meski konsumsi *resource* lebih besar dibanding Nginx [10].

Konfigurasi Apache melibatkan aktivasi modul seperti mod_proxy, mod_ssl, dan mod_proxy_balancer untuk mendukung SSL termination, load balancing, dan routing lanjutan. Keunggulannya terletak pada fleksibilitas konfigurasi, dukungan berbagai protokol dan metode otentikasi, serta kemampuan fine-tuning melalui direktif yang detail. Namun, konsumsi CPU dan RAM yang tinggi menjadi pertimbangan pada perangkat terbatas seperti Raspberry Pi [11].

D. *Virtual Private Network*

Virtual Private Network (VPN) adalah teknologi yang mengamankan pengiriman data antar jaringan melalui jalur terenkripsi, dengan enkapsulasi dan autentikasi penerima untuk menjaga keamanan, terutama saat menghubungkan antar divisi dalam perusahaan [12].

*Open*VPN adalah software *open-source* untuk membangun koneksi VPN yang aman melalui internet. Berdasarkan jurnal "Implementasi VPN pada VPS Server menggunakan *Open*VPN dan Raspberry Pi", *Open*VPN meningkatkan keamanan dan kecepatan transfer data. Pengguna dapat mengakses jaringan seolah berada di lokasi yang sama dengan IP yang berbeda dari IP fisik, meningkatkan privasi. Kekurangannya ada pada konfigurasi yang kompleks dan butuh pengetahuan teknis, namun tetap efektif untuk keamanan dan komunikasi jarak jauh [2][13].

E. *OpenVPN*

*Open*VPN adalah solusi VPN populer yang memungkinkan koneksi jaringan privat yang aman melalui internet, menggunakan *tunneling* dan enkripsi untuk melindungi data. Sistem ini membuat "terowongan virtual" yang memungkinkan pengguna mengakses jaringan internal seolah-olah berada di lokasi yang sama [14].

*Open*VPN menggunakan SSL/TLS untuk otentikasi dan enkripsi. Koneksi dimulai dengan handshake antara server dan client untuk pertukaran kunci dan verifikasi sertifikat digital. Data kemudian dienkripsi (misalnya dengan AES) dan hanya dapat dibaca oleh pihak yang berwenang. Metode otentikasi dapat berupa username/password, sertifikat digital, atau pre-shared key [15].

Arsitekturnya berbasis *client-server*, di mana server mengelola koneksi dari banyak client. Enkripsi bekerja di level network layer, sehingga semua paket data dari interface virtual dienkripsi sebelum dikirim dan didekripsi saat tiba. *Open*VPN mendukung berbagai topologi seperti point-to-point dan site-to-site VPN [16].

*Open*VPN efektif untuk akses remote yang aman. Untuk *home* server, seperti dengan Raspberry Pi, *Open*VPN memungkinkan akses dari mana saja dengan tingkat keamanan tinggi. Dalam penelitian ini, *Open*VPN menjadi

teknologi inti yang menghubungkan VPS dan *home* server melalui *reverse* proxy [14].

F. *Virtual Pivate Server*

Virtual Private Server (VPS) adalah hasil virtualisasi server fisik menjadi beberapa lingkungan virtual yang terisolasi dan independen, menggabungkan fleksibilitas *dedicated* server dengan efisiensi biaya. Dalam penelitian ini, VPS digunakan sebagai infrastruktur cloud untuk hosting *Open*VPN server dan layanan *reverse* proxy yang menghubungkan *home* server Pi01 (Nginx) dan Pi02 (Apache) melalui tunnel *Open*VPN terenkripsi [17].

VPS menggunakan teknologi virtualisasi berbasis hypervisor atau container yang menyediakan sistem operasi terisolasi untuk tiap instance sesuai kebutuhan. Dengan fitur isolasi sumber daya, kemampuan restart mandiri, dan kendali konfigurasi penuh, VPS01 dioptimalkan sebagai server OpenVPN yang melayani dua reverse proxy berbeda. Arsitekturnya meliputi hypervisor, mesin virtual/container, antarmuka jaringan virtual, dan penyimpanan virtual. VPS01 mengelola IP, firewall, dan routing secara mandiri, termasuk konfigurasi subnet OpenVPN 10.10.20.0/24.

Keamanan dijamin melalui lapisan virtualisasi dan kontrol sistem operasi untuk melindungi data dalam tunnel. Dalam konteks enterprise dan home server, VPS cocok untuk web deployment, akses jarak jauh, dan solusi berperforma tinggi. Dibanding server dedicated, VPS lebih efisien, fleksibel, dan kompatibel dengan arsitektur Nginx (event-driven) dan Apache (multi-process), sekaligus menambah lapisan keamanan dan konfigurasi reverse proxy, sehingga menjadi solusi optimal dengan ketersediaan tinggi dan performa stabil pada OpenVPN [18].

G. *Raspberry Pi*

Raspberry Pi adalah Raspberry Pi adalah komputer papan tunggal berbasis ARM yang hemat energi dan fleksibel, cocok sebagai home server. Model 4 B dengan prosesor quad-core Cortex-A72 1,8 GHz dan konsumsi daya 2,5–7,6W dapat menjalankan reverse proxy Nginx (Pi01) dan Apache (Pi02) melalui tunnel OpenVPN dengan beban minimal. Sistem ini menggunakan Raspberry Pi OS 64-bit berbasis Debian dan mendukung layanan jaringan dengan enkripsi AES-256-CBC.

Meskipun antarmuka Gigabit Ethernet tersedia, throughput OpenVPN terbatas antara 20–80 Mbps karena enkripsi berbasis CPU tanpa akselerasi perangkat keras. Dengan SoC Broadcom BCM2711, RAM hingga 8GB, dan bandwidth memori 12,8 GB/s, Pi01 dan Pi02 mengoperasikan reverse proxy berbeda pada IP 10.10.20.2 dan 10.10.20.3 dalam jaringan OpenVPN. Bottleneck muncul pada CPU saat enkripsi/dekripsi koneksi paralel. Raspberry Pi merupakan solusi ekonomis dan ramah lingkungan untuk akses jarak jauh melalui OpenVPN [19].

H. *Home Server*

Home server adalah sistem penyimpanan data terpusat dalam jaringan rumah, berfungsi untuk layanan multimedia, file sharing, dan backup data secara mandiri. Sistem ini menggunakan teknologi *Network Attached Storage* (NAS) yang memungkinkan akses file melalui jaringan, lebih sederhana dibanding SAN atau DAS. Dengan arsitektur *client-server*, *home* server mendukung akses remote dan streaming *real-time* [20].

I. Ubuntu

Ubuntu adalah sistem Ubuntu adalah sistem operasi open source berbasis Linux yang dikembangkan oleh Canonical Ltd., dikenal karena stabilitas, keamanan, dan fleksibilitas dalam pengelolaan jaringan. Sistem ini mendukung virtualisasi (KVM) dan kontainerisasi (Docker), serta menerima pembaruan keamanan secara rutin untuk melindungi data dari ancaman siber.

Ubuntu menggunakan kernel Linux, repositori paket, dan beragam tools keamanan yang mendukung protokol kriptografi modern serta deteksi ancaman secara proaktif. Karena kemudahan instalasi dan dokumentasi lengkap, Ubuntu banyak digunakan pada server, termasuk *home* server, serta menawarkan stabilitas dan dukungan protokol keamanan pada layanan VPN dan *reverse proxy*, sehingga menjadi standar solusi keamanan jaringan saat ini [21][20].

J. Analisis Kinerja

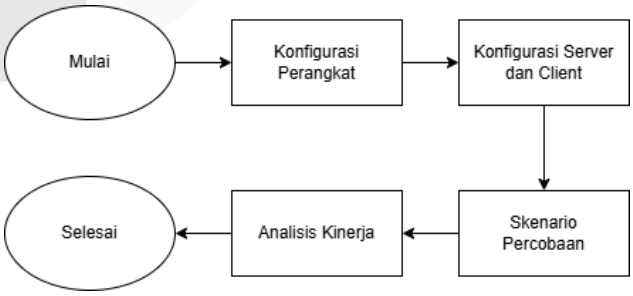
Analisis kinerja jaringan merupakan proses untuk mengevaluasi efektivitas sistem jaringan, khususnya saat menerapkan *reverse* proxy Nginx dan Apache melalui protokol *Open*VPN di infrastruktur *home* server. Evaluasi ini mencakup pengukuran *throughput*, *latency*, *response time*, dan penggunaan CPU/RAM, dengan fokus pada dampak enkripsi *Open*VPN serta perbedaan arsitektur *event-driven* Nginx dan multi-process.

Data dikumpulkan melalui monitoring menggunakan *tools* seperti curl, ping, *top*, *free*, dan iperf, yang disesuaikan untuk melihat performa masing-masing *reverse* proxy dalam skenario VPN terenkripsi. Server VPS01 berfungsi sebagai endpoint *Open*VPN, sedangkan Pi01 dan Pi02 sebagai klien yang menjalankan *reverse* proxy berbeda.

Simulasi dilakukan untuk menguji performa pada berbagai beban trafik, membandingkan efisiensi Nginx yang unggul dalam menangani koneksi paralel dan konsumsi memori rendah, dengan Apache yang lebih fleksibel melalui sistem modulnya. Analisis ini bertujuan menentukan solusi terbaik dalam lingkungan *home* server terbatas, dengan tetap menjaga keamanan melalui enkripsi *Open*VPN.

III. METODE

Diagram blok pada Gambar 1 menggambarkan langkah-langkah untuk menganalisis kinerja Nginx dan apache sebagai *reverse* proxy menggunakan *Open*VPN untuk akses *home* server. Setiap bagian akan di jelaskan di bawah ini.



GAMBAR 1
Diagram Blok Implementasi

A. Konfigurasi Perangkat

Penelitian ini melibatkan empat komponen utama dalam implementasi proksi terbalik ganda berbasis *Open*VPN: VPS01 sebagai peladen *Open*VPN pusat, VPS02 sebagai klien pengujian, serta Pi01 dan Pi02 sebagai peladen rumah yang menjalankan proksi terbalik Nginx dan Apache. VPS01 menggunakan subnet 10.10.20.0/24 di port UDP 1194. Pi01 (IP 10.10.20.2) menjalankan Nginx untuk meneruskan permintaan HTTP melalui terowongan *Open*VPN, sementara Pi02 (IP 10.10.20.3) menggunakan Apache dengan modul proksi aktif. Kedua proksi mengarahkan permintaan dari VPS02 ke layanan web masing-masing, dengan virtual host pi1.habeebqr.my.id untuk Nginx dan pi2.habeebqr.my.id untuk Apache, serta seluruh lalu lintas terenkripsi melalui *Open*VPN.

B. Konfigurasi Server dan Client

Konfigurasi server dan klien dimulai dengan instalasi *Open*VPN di VPS01 menggunakan *easy*-rsa untuk manajemen sertifikat. Konfigurasi utama disimpan di /etc/*open*vpn/server.conf dengan IP 10.10.20.1/24, port UDP 1194, enkripsi AES-256-CBC, dan iptables untuk meneruskan lalu lintas ke Pi01 dan Pi02. Sertifikat klien dibuat via build-key dan disimpan di /etc/*open*vpn/keys/.

Di Pi01, *reverse* proxy Nginx dikonfigurasi lewat /etc/nginx/sites-available/default untuk meneruskan HTTP ke port 808056. Di Pi02, Apache dikonfigurasi via /etc/apache2/sites-available/000-default.conf dengan modul proxy aktif. Keduanya memakai header HTTP seperti X-Real-IP dan X-Forwarded-For untuk pelacakan IP asli klien.

VPS02 sebagai klien pengujian menggunakan curl, iperf3, dan ping via SSH ke VPS01 dengan ssh user@vps01-ip "./test_script.sh" untuk menjalankan pengukuran HTTP, latensi, throughput, serta pemantauan CPU dan RAM.

Pi01 dan Pi02 bertindak sebagai web server sederhana di /var/www/html, masing-masing terkoneksi ke *Open*VPN dengan IP 10.10.20.2 dan 10.10.20.3. Validasi akhir dilakukan dengan ping, traceroute, nginx -t, apache2ctl configtest, dan *open*vpn --config client.ovpn guna memastikan sistem siap menjalankan skenario *reverse* proxy ganda secara terstruktur.

C. Skenario Percobaan

TABEL 2
Skenario Percobaan

Skenario	Reserve Proxy	VPN	Target	Metrik
1	Nginx *Pi01 + VPS02*	Ya	Pi01 *via VPS01*	*Response time, latency, resource usage*
2	Apache Pi02 + *VPS02*	Ya	Pi02 *via VPS01*	*Response time, latency, resource usage*
3	Nginx *Pi01 + VPS02* tanpa *domain*	Ya	Pi01	*Response time, latency, resource usage*
4	Apache Pi02 + *VPS02* tanpa *domain*	Ya	Pi02	*Response time, latency, resource usage*

Penelitian ini merancang eksperimen untuk membandingkan performa Nginx dan Apache sebagai *reverse* proxy dalam akses *home* server melalui *Open*VPN, VPS, dan Raspberry Pi. Empat skenario diuji: (1) Apache dengan VPN (sebagai *reverse* proxy), (2) Nginx dengan VPN (sebagai *reverse* proxy), (3) Apache direct (akses langsung tanpa proxy), dan (4) Nginx direct.

Pada skenario pertama, *reverse* proxy Nginx di VPS01 mengarahkan permintaan HTTP ke Pi01 (10.10.20.2) melalui domain pi1.habeebqr.my.id dengan tunnel *Open*VPN terenkripsi. Ini memungkinkan analisis efisiensi arsitektur *event-driven* Nginx dalam menangani request simultan. Skenario kedua menguji Apache sebagai *reverse* proxy di VPS01 melalui pi2.habeebqr.my.id ke Pi02 (10.10.20.3), menyoroti arsitektur multi-process Apache dalam mengelola HTTP traffic dan konsumsi *resource*.

Skenario ketiga dan keempat mengakses langsung Pi01 dan Pi02 lewat IP-nya dalam jaringan VPN tanpa *reverse* proxy, untuk memperoleh data baseline. Pendekatan ini memungkinkan pembandingan langsung dampak penambahan *reverse* proxy terhadap *response time*, latensi, dan efisiensi *resource*.

Keempat skenario dirancang sistematis guna memperoleh dataset komprehensif yang mendukung analisis trade-off antara aksesibilitas, keamanan, dan performa, serta mengidentifikasi konfigurasi optimal untuk implementasi *home* server dalam konteks produksi.

D. Analisis Kerja

Analisis kinerja dalam penelitian ini berfokus pada empat metrik utama: *response time*, *latency*, penggunaan CPU, dan RAM. Setiap metrik diukur dengan alat dan parameter terstandar guna memastikan validitas data dalam evaluasi performa *reverse* proxy Nginx dan Apache melalui protokol *Open*VPN.

TABEL 2
Metrik Analisis Kerja

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}"*
RAM	*free*	Penggunaan memori	*free | awk "/^Mem:/*

		dalam persentase	*{printf \"%.1f\", $3/$2*100}"*

Analisis kinerja dalam penelitian ini mencakup empat metrik utama, yaitu *response time*, *latency*, penggunaan CPU, dan RAM. *Response time* diukur menggunakan perintah curl, yang mencatat durasi dari pengiriman permintaan HTTP hingga seluruh respons diterima. Hasilnya menunjukkan bahwa Nginx memberikan *response time* yang lebih konsisten dibandingkan Apache karena arsitektur *event-driven* yang efisien dalam menangani koneksi simultan. *Latency* diukur dengan ping terhadap 20 paket ICMP, dan *reverse* proxy Nginx menunjukkan nilai *latency* yang lebih rendah dibandingkan Apache karena *overhead* pemrosesan yang lebih ringan. Untuk penggunaan sumber daya, pemantauan dilakukan menggunakan *top* dan *free*, di mana Apache terlihat mengonsumsi CPU dan RAM lebih tinggi akibat model multi-*process* yang membutuhkan memori terpisah per proses, sementara Nginx lebih efisien dengan satu master *process* dan *worker threads*. Analisis korelasi antar metrik menunjukkan bahwa Nginx unggul dalam performa saat beban tinggi, sedangkan Apache cenderung lebih stabil pada konfigurasi kompleks seperti SSL/TLS. Keduanya mengalami peningkatan *latency* ketika ditambahkan lapisan *reverse* proxy di VPS01 akibat tambahan hop jaringan melalui *Open*VPN. Secara keseluruhan, pemilihan antara Nginx dan Apache bergantung pada prioritas sistem, di mana Nginx lebih cocok untuk kebutuhan throughput tinggi dengan respons cepat, dan Apache lebih sesuai untuk skenario yang menuntut fleksibilitas konfigurasi dan dukungan modul, meskipun dengan konsumsi sumber daya yang lebih besar.

IV. HASIL DAN PEMBAHASAN

A. Skenario Percobaan

Pengujian kinerja sistem dilakukan dalam environment terkontrol menggunakan koneksi internet fiber optic 100 Mbps dan router rumahan standar. Pengujian dijadwalkan dini hari (pukul 03.00–05.00 WIB) untuk meminimalkan gangguan eksternal dan menjaga konsistensi data. Infrastruktur pengujian terdiri dari empat perangkat utama: satu VPS (IP 20.255.49.168) sebagai server *Open*VPN sekaligus *reverse* proxy (Nginx dan Apache), dua Raspberry Pi (10.10.20.2 dan 10.10.20.3) sebagai *backend* dan client *Open*VPN, serta satu workstation lokal sebagai terminal pengujian. Tiap perangkat dikonfigurasi sesuai fungsinya dengan kombinasi interface eth0/wlan0 untuk akses internet dan tun0 untuk koneksi VPN. Pengujian membandingkan empat skenario: akses ke *backend* via Nginx (pi1.habeebqr.my.id), via Apache (pi2.habeebqr.my.id), serta *direct access* ke masing-masing Raspberry Pi menggunakan IP VPN tanpa *reverse* proxy. Metrik yang diuji meliputi *response time*, *latency*, penggunaan CPU, dan memori. Pengumpulan data dilakukan menggunakan skrip bash otomatis yang mengintegrasikan curl, ping, dan SSH untuk monitoring performa secara *real-time* di server.

1. Nginx sebagai Reverse Proxy dengan OpenVPN

Skenario awal menguji performa aksesibilitas *backend* Pi1 (IP 10.10.20.2) melalui *reverse* proxy Nginx yang dioperasikan pada VPS1. Proses benchmarking dilakukan dari VPS2 sebagai workstation evaluasi, menggunakan endpoint domain *pi1.habeebqr.my.id* yang mengarahkan seluruh traffic HTTP melewati Nginx sebelum diteruskan ke Pi1 melalui tunnel *Open*VPN. Hasil pengujian menunjukkan rata-rata waktu respons sebesar 0,1546 detik dan latensi jaringan 60,097 milidetik tanpa terjadi packet loss. Pemantauan terhadap VPS1 menunjukkan utilisasi CPU sebesar 3,1% dan penggunaan RAM sebesar 8,3% selama pengujian. Hasil ini menunjukkan bahwa Nginx mampu menjalankan fungsi *reverse* proxy dengan performa stabil, efisiensi penggunaan *resource* yang baik, dan tingkat responsivitas akses yang tinggi.

2. Apache sebagai Reverse Proxy dengan OpenVPN

Evaluasi tahap selanjutnya difokuskan pada pengujian *backend* Pi2 (IP 10.10.20.3) yang diakses melalui *reverse* proxy Apache pada VPS1. Proses benchmarking dilakukan dengan pendekatan serupa menggunakan domain pi2.habeebqr.my.id, di mana seluruh permintaan HTTP dialihkan melalui Apache dan diteruskan ke Pi2 melalui infrastruktur tunnel *Open*VPN yang telah dikonfigurasi sebelumnya. Hasil pengujian menunjukkan waktu respons rata-rata sebesar 0,1062 detik, disertai latensi rata-rata 61,444 milidetik dan tanpa terjadi packet loss. Pemantauan terhadap utilisasi sistem pada VPS1 mencatat konsumsi CPU yang meningkat hingga 6,1 persen, sementara penggunaan RAM berada pada kisaran 7,9 persen. Temuan ini mengindikasikan bahwa Apache mampu menjalankan peran *reverse* proxy dengan kinerja yang solid dalam skenario koneksi VPN menuju *home* server, meskipun menunjukkan pola konsumsi CPU yang lebih tinggi dibandingkan Nginx.

3. Nginx sebagai Reverse Proxy dengan OpenVPN tanpa Domain

Skenario evaluasi ketiga mengadopsi pendekatan akses langsung terhadap *backend* Pi1 melalui alamat IP VPN tanpa keterlibatan *reverse* proxy sebagai perantara. Berdasarkan hasil benchmarking, waktu respons rata-rata tercatat sebesar 0,1214 detik dengan latensi jaringan 60,369 milidetik serta konsistensi *zero packet loss* yang tetap terjaga. Namun, observasi terhadap utilisasi *resource* menunjukkan adanya lonjakan signifikan pada beban CPU VPS1 yang mencapai 15,2 persen, sementara konsumsi RAM relatif stabil pada angka 7,8 persen. Temuan yang bersifat paradoksal ini mengindikasikan bahwa meskipun akses langsung cenderung menghasilkan waktu respons yang sedikit lebih optimal dibandingkan dengan skenario menggunakan *reverse* proxy Nginx, beban pemrosesan CPU justru meningkat tajam akibat seluruh traffic VPN diproses langsung oleh server tanpa adanya mekanisme distribusi atau pengelolaan beban tambahan dari *reverse* proxy.

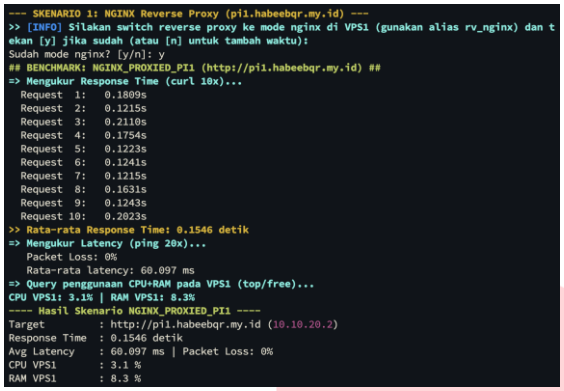
4. Apache sebagai Reverse Proxy dengan OpenVPN tanpa Domain

Implementasi skenario terakhir melibatkan akses langsung menuju *backend* Pi2 melalui alamat IP VPN tanpa melalui mekanisme *reverse* proxy sebagai gateway. Pengujian komprehensif menunjukkan waktu respons rata-rata sebesar 0,1291 detik dengan latensi jaringan mencapai 62,353 milidetik, serta mempertahankan konsistensi *zero packet loss* sepanjang proses evaluasi. Hasil monitoring *resource* mencatat utilisasi CPU VPS1 pada level 3,2 persen dan konsumsi RAM sebesar 7,8 persen selama periode pengujian berlangsung. Interpretasi terhadap temuan ini mengindikasikan bahwa pendekatan akses langsung terhadap Pi2 mampu menghasilkan performa yang stabil dari sisi responsivitas maupun efisiensi penggunaan sumber daya sistem, serta menawarkan karakteristik performa yang

distinctive apabila dibandingkan dengan metode akses berbasis *reverse* proxy.

B. Hasil Percobaan Skenario

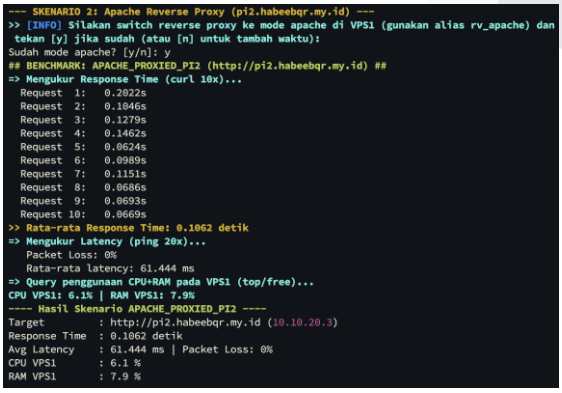
1. Hasil Pengujian Nginx sebagai *Reverse* Proxy dengan *Open*VPN



GAMBAR 2
Hasil Skenario Pertama

Evaluasi tahap awal berfokus pada analisis kinerja aksesibilitas *backend* Pi1 yang diakses melalui alamat IP 10.10.20.2 dan diproses melalui lapisan *reverse* proxy NGINX pada infrastruktur VPS1. Metodologi benchmarking dilakukan dari VPS2 yang berperan sebagai workstation penguji, dengan menggunakan endpoint domain pi1.habeebqr.my.id untuk mengarahkan seluruh traffic HTTP melalui NGINX di VPS1 sebelum diteruskan ke *backend* Pi1 melalui jalur tunnel *Open*VPN terenkripsi. Hasil evaluasi menunjukkan performa yang stabil dengan waktu respons rata-rata 0,1546 detik dan latensi jaringan rata-rata sebesar 60,097 milidetik, serta mempertahankan *zero packet loss* secara konsisten. Observasi terhadap utilisasi sumber daya sistem pada VPS1 menunjukkan beban CPU yang terkendali pada level 3,1 persen, serta konsumsi memori RAM sebesar 8,3 persen selama periode pengujian. Temuan ini memvalidasi kapabilitas NGINX dalam menyajikan layanan *reverse* proxy yang andal, dengan stabilitas performa yang unggul, efisiensi penggunaan *resource* yang optimal, serta tingkat responsivitas yang mampu memenuhi ekspektasi pengguna akhir.

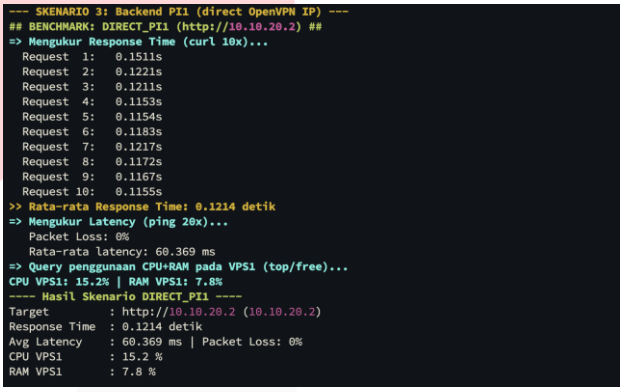
2. Hasil Pengujian Apache sebagai *Reverse* Proxy dengan *Open*VPN



GAMBAR 3
Hasil Skenario Kedua

Skenario selanjutnya menguji akses *backend* Pi2 (IP 10.10.20.3) melalui *reverse* proxy Apache di VPS1, menggunakan domain pi2.habeebqr.my.id yang mengarahkan trafik HTTP ke Pi2 via tunnel *Open*VPN. Hasil benchmarking menunjukkan waktu respons rata-rata 0,1062 detik dengan latensi 61,444 milidetik dan *zero packet loss* yang konsisten. Selama pengujian, VPS1 mencatatkan konsumsi CPU sebesar 6,1 persen dan RAM 7,9 persen. Temuan ini menegaskan bahwa Apache mampu memberikan kinerja *reverse* proxy yang andal, meskipun dengan konsumsi CPU yang lebih tinggi dibanding NGINX pada skenario sebelumnya.

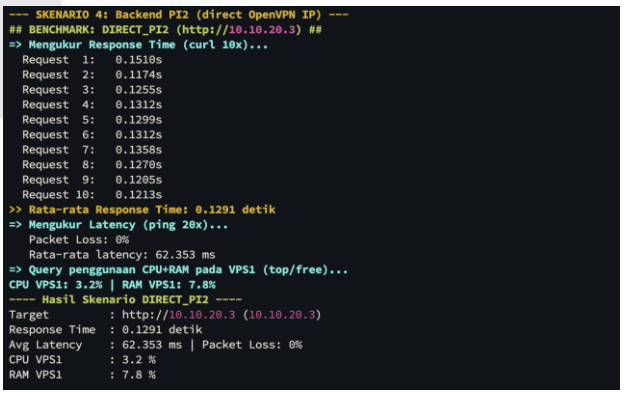
3. Hasil Pengujian Nginx sebagai *Reverse* Proxy dengan *Open*VPN tanpa Domain



GAMBAR 4
Hasil Skenario Ketiga

Skenario ketiga menguji akses langsung ke *backend* Pi1 via IP VPN tanpa *reverse* proxy. Benchmark mencatat waktu respons rata-rata 0,1214 detik, latensi 60,369 milidetik, dan *zero packet loss*. Namun, beban CPU VPS1 meningkat signifikan hingga 15,2 persen, meski RAM tetap stabil di 7,8 persen. Hal ini menunjukkan bahwa meski respons cukup cepat, akses langsung menimbulkan CPU yang tinggi akibat penanganan traffic VPN tanpa bantuan lapisan proxy.

4. Hasil Pengujian Apache sebagai *Reverse* Proxy dengan *Open*VPN tanpa Domain



GAMBAR 5
Hasil Skenario Keempat

Skenario terakhir menguji akses langsung ke *backend* Pi2 via IP VPN tanpa *reverse* proxy. Hasil menunjukkan respons rata-rata 0,1291 detik, latensi 62,353 milidetik, dan *zero packet loss*. Selama pengujian, CPU VPS1 terpantau stabil di 3,2 persen dan RAM di 7,8 persen. Temuan ini menegaskan bahwa akses direct ke Pi2 menawarkan performa stabil dan efisien, serta memperlihatkan karakteristik berbeda dibanding pendekatan *reverse* proxy sebelumnya.

C. Analisis Kerja

Dataset hasil pengujian keempat skenario mengungkapkan adanya divergensi performa yang substansial antara metodologi *reverse* proxy berbasiskan Nginx dan Apache versus pendekatan *direct access*. Secara komprehensif, implementasi *reverse* proxy Apache menghasilkan waktu respons paling optimal dengan nilai 0,1062 detik, diikuti oleh akses direct menuju Pi1 pada 0,1214 detik, kemudian akses direct ke Pi2 dengan 0,1291 detik, dan terakhir *reverse* proxy NGINX dengan 0,1546 detik. Namun demikian, dalam konteks efisiensi utilisasi CPU pada VPS1, akses direct menuju Pi1 memicu eskalasi konsumsi CPU yang signifikan hingga mencapai 15,2 persen, sementara implementasi *reverse* proxy NGINX dan Apache menunjukkan efisiensi superior dengan konsumsi masing-masing 3,1 persen dan 6,1 persen.

TABEL 3
Analisis Kerja

Skenario	***Response time* (*ms*)**	***Latency* (*ms*)**	***CPU* (%)**	***RAM* (%)**
NGINX + *Open*VPN (pi1.habeebqr)	0,1546	60,097	3,1	8,3
Apache + *Open*VPN (pi2.habeebqr)	0,1062	61,444	6,1	7,9
NGINX + *Open*VPN tanpa *domain* (10.10.20.2)	0,1214	60,369	15,2	7,8
Apache + *Open*VPN tanpa *domain* (10.10.20.3)	0,1291	62,353	3,2	7,8
NGINX + *Open*VPN (pi1.habeebqr)	0,1546	60,097	3,1	8,3

Profil *latency* keseluruhan skenario memperlihatkan konsistensi relatif dalam rentang 60 hingga 62 milidetik, dengan *zero packet loss* yang terjaga konsisten, mengindikasikan stabilitas infrastruktur link VPN dan jaringan lokal yang reliable.

V. KESIMPULAN

Berdasarkan dataset hasil evaluasi benchmark terhadap empat metodologi akses *home* server melalui protokol *Open*VPN dengan implementasi *reverse* proxy Nginx dan Apache, diperoleh temuan empiris sebagai berikut:

Evaluasi skenario ini melibatkan aksesibilitas *backend* Pi1 melalui endpoint domain pi1.habeebqr.my.id yang dikonstruksikan sebagai *reverse* proxy Nginx pada infrastruktur VPS1. Dataset pengujian memperlihatkan waktu respons rata-rata 0,1546 detik, latensi rata-rata 60,097 milidetik dengan konsistensi *zero packet loss*, serta utilisasi CPU VPS1 pada level 3,1 persen dan konsumsi RAM 8,3 persen selama periode benchmarking berlangsung. Hasil ini mendemonstrasikan stabilitas operasional dan efisiensi pengelolaan sumber daya sistem pada akses melalui implementasi Nginx *reverse* proxy.

Pada evaluasi kedua, akses menuju *backend* Pi2 dijalankan melalui endpoint domain pi2.habeebqr.my.id yang beroperasi sebagai *reverse* proxy Apache pada VPS1. Hasil benchmarking menghasilkan waktu respons rata-rata 0,1062 detik, latensi rata-rata 61,444 milidetik dengan *maintenance zero packet loss*, disertai penggunaan CPU VPS1 sebesar 6,1 persen dan RAM 7,9 persen. Implementasi *reverse* proxy Apache menghadirkan waktu respons paling optimal di antara keseluruhan skenario, namun mensyaratkan konsumsi CPU yang hampir berlipat ganda dibandingkan dengan implementasi Nginx *reverse* proxy.

Metodologi akses langsung ke Pi1 via alamat IP VPN tanpa melibatkan lapisan *reverse* proxy menghasilkan waktu respons rata-rata 0,1214 detik, latensi 60,369 milidetik, serta utilisasi CPU VPS1 yang meningkat signifikan yaitu 15,2 persen dan RAM 7,8 persen. Implementasi *direct access* tanpa *reverse* proxy memicu eskalasi beban CPU pada VPS1 meskipun waktu respons tetap mempertahankan karakteristik kompetitif.

Pendekatan akses langsung ke Pi2 menggunakan alamat IP VPN menunjukkan waktu respons rata-rata 0,1291 detik, latensi 62,353 milidetik, dengan utilisasi CPU 3,2 persen dan RAM 7,8 persen. Profil kinerja secara holistik memperlihatkan similaritas dengan akses direct menuju Pi1, namun dengan waktu respons yang sedikit lebih tinggi dan utilisasi CPU yang lebih rendah dibandingkan dengan implementasi *reverse* proxy Apache.

REFERENSI

[1] S. S.-J. Moon dan H, "Agent for *Home* Server Management in Intelligent Smart *Home* Network," *Int. J. Internet, Broadcast. Commun.*, vol. 14, no. 2, pp. 225–230, 2022, doi: 10.7236/IJIBC.2022.14.2.225.

[2] M.Affandi, "Implementasi *Virtual Private Network* (Vpn) *Open* vpn Dengan Keamanan Sertifikat SSL pada Network Attached Storage (Nas) Freenas," *J. Impresi Indones.*, vol. 1, no. 12, pp. 1329–1341, 2022, doi: 10.58344/jii.v1i12.748.

[3] S. I, K, S, Satwika dan K. N, "PERBANDINGAN PERFORMANSI WEB SERVER APACHE DAN NGINX DENGAN MENGGUNAKAN IPV6," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 15, no. 1, 2020, doi: 10.33005/scan.v15i1.1847.

[4] W. Y, "Implementasi Keamanan Jalur Internet Menggunakan IP Tunneling pada *Open*VPN Access Server dengan Protokol *Open*VPN dan Protokol DNS Over HTTPS," *J. Syntax Admiration*, vol. 2, no. 4, pp. 712–730, 2021, doi: 10.46799/jsa.v2i4.207.

[5] D. K. M. K and A. Rengarajan, "*Reverse* Proxy Technology," *Int. J. Innov. Res. Comput. Commun.*

Eng., vol. 12, no. 02, pp. 1067–1071, 2024, doi: 10.15680/IJIRCCE.2024.1202057.

[6] dan S. S. Lady Agustine, "Penerapan Metode SAW dalam Analisa Perbandingan Performa Web server (Apache, Nginx, Lighttpd, Iis) pada Bahasa Pemrograman PHP," *remik*, vol. 7, no. 1, pp. 409–420, 2023, doi: 10.33395/remik.v7i1.12075.

[7] N. R. Proxy, "Nginx *Reverse* Proxy."

[8] A. M, "Perbandingan Kinerja Nginx dan Caddy sebagai Web Server untuk Aplikasi PHP," *Insect (Informatics Secur. J. Tek. Inform.*, vol. 11, no. 1, pp. 88–96, 2025, doi: 10.33506/insect.v11i1.4223.

[9] D. K. F. H. Z. Bustomi, M. Syahiruddin, M. I. Afandi, "Load Balancing Web Server Menggunakan Nginx pada Lingkungan Virtual," *J. Inform. J. Pengemb. IT*, vol. 5, no. 1, pp. 32–36, 2020, doi: 10.30591/jpit.v5i1.1745.

[10] APACHE, "Apache HTTP Server Version 2.4 *Reverse* Proxy Guide." Accessed: Jun. 12, 2024. [Online]. Available: https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html

[11] C. A. Y, "Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request," *J. Sist. dan Inform.*, vol. 14, no. 1, pp. 48–56, 2019, doi: 10.30864/jsi.v14i1.248.

[12] T. A. E. Suhadi, "RANCANGAN *VIRTUAL PRIVATE NETWORK* PADA KANTOR PROLOV MENGGUNAKAN ZEROTIER," *JIKA (Jurnal Inform.*, vol. 8, no. 1, p. 66, 2024, doi: 10.31000/jika.v8i1.9979.

[13] Jul, "Performance Evaluation of Secured *Virtual Private Network* based on Dynamic Multipoint *Virtual Private Network*," ResearchGate.

[14] *Open*VPN, "What is *Open*VPN," *Open*VPN. Accessed: Jun. 12, 2024. [Online]. Available: https://*open*vpn.net/faq/what-is-*open*vpn/#:~:text=The%2520*Open*VPN%2520Community%2520Edition%25

[15] I. E. Papadogiannaki dan S, "A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures," *ACM Comput Surv*, vol. 54, no. 6, pp. 1–35, 2022, doi: 10.1145/3457904.

[16] S. S. B. W. Aulia, M. Rizki, P. Prindiyana, "Peran Krusial Jaringan Komputer dan Basis Data dalam Era Digital," *JUSTINFO | J. Sist. Inf. dan Teknol. Inf.*, vol. 1, no. 1, pp. 9–20, 2023, doi: 10.33197/justinfo.vol1.iss1.2023.1253.

[17] A. K. S. A. Kumar, G. Sharma, P. Jain, A. Upadhyay, S. Sharma, "Virtual environments testing in cloud service enviorment: a framework to optimize the performance of virtual applications," *Int. J. Syst. Assur. Eng. Manag.*, vol. 13, 2022, doi: 10.1007/s13198-021-01105-y.

[18] H. K. T. Rahman, G. M. V. T. Mariatmojo, H. Nurdin, "Implementasi VPN Pada VPS Server Menggunakan *Open*VPN dan Raspberry Pi," *Teknika*, vol. 11, no. 2, pp. 138–147, 2022, doi: 10.34148/teknika.v11i2.482.

[19] Z.-D. Z. et Al, "TopADDPi: An Affordable and Sustainable Raspberry Pi Cluster for Parallel-Computing Topology Optimization," *Processes*, vol. 13, no. 3, p. 633, 2025, doi: 10.3390/pr13030633.

[20] A. A. R. Rakhmadi Rahman, Awal Ramadhan Nasrun, "Desain dan Implementasi Sistem Operasi Linux Ubuntu Versi 22.04 untuk Perlindungan Data dari Serangan Komputasi Kuantum," *Bridg. J. Publ. Sist. Inf. dan Telekomun.*, vol. 2, no. 3, pp. 207–213, 2024, doi: 10.62951/bridge.v2i3.159.

[21] L. T. H. Tang, S. S. Kolahi, "Evaluation of HTTP Flood DDoS Cyber Attack on Apache2 Web Server with Linux Ubuntu 22.04," *2023 IEEE Int. Conf. Comput. (ICOCO), IEEE*, pp. 53–58, 2023, doi: 10.1109/ICOCO59262.2023.10398152.