Analisis Perbandingan Penggunaan Metode *Tunneling Cloud Virtual Private Network* dan WireGuard *Virtual Private Network* pada Implementasi Infrastruktur *Hybrid Cloud*

Rusdi Hermawan[1], Yuris Mulya Saputra[1,*]
[1]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
rusdi.h@mail.ugm.ac.id
[*]Korespondensi: ym.saputra@ugm.ac.id;

Abstract – The continuous advancement of server infrastructure technology drives organizations to migrate from on-premise to public cloud systems, despite data security and cost challenges. Hybrid cloud infrastructure becomes essential for seamless integration, minimizing downtime, and maintaining operational consistency. This implementation often uses tunneling methods, leveraging VPN technology for secure data transfer between on-premise servers and public cloud services. This research compares two VPN tools: WireGuard (open-source) and Google Cloud Platform's (GCP) Cloud VPN (enterprise solution) in hybrid-cloud contexts. The study measures performance parameters such as latency, throughput, connection stability, ease of configuration, and service costs to evaluate each tool's suitability. Findings show that WireGuard VPN has a throughput advantage of 676.67% and reduces average total latency by 97.66% in initial tests, indicating superior performance compared to* Classic Cloud *VPN.

Keywords – Hybrid Cloud, Tunneling, VPN, WireGuard,* Classic Cloud *VPN

Intisari – Perkembangan teknologi dalam infrastruktur *server* terus maju signifikan setiap tahun. Organisasi dan perusahaan cenderung beralih ke sistem yang lebih kompleks dan terpusat, mendorong migrasi infrastruktur *server* dari *on-premise* ke *public cloud*. Banyak migrasi dilakukan bertahap karena keterbatasan biaya dan keamanan data. Dalam konteks ini, infrastruktur *hybrid cloud* penting untuk integrasi yang mulus, meminimalkan *downtime*, dan menjaga konsistensi operasional. *Hybrid cloud* sering menggunakan *tunneling* untuk menghubungkan *server on-premise* dengan *public cloud*, menggunakan teknologi *Virtual Private Network* (VPN) untuk jalur terenkripsi yang aman. Penelitian ini membandingkan dua alat VPN: WireGuard sebagai solusi *open-source* dan Cloud VPN dari Google Cloud Platform (GCP) sebagai solusi *enterprise*, dalam konteks *hybrid-cloud*. Penelitian fokus pada pengukuran kinerja parameter seperti *latency*, *throughput*, kestabilan koneksi, kemudahan konfigurasi, dan biaya layanan untuk memahami kecocokan masing-masing alat dalam skenario implementasi tertentu. Hasilnya menunjukkan bahwa *throughput* WireGuard VPN unggul 676,67%, dan rata-rata *latency* WireGuard VPN berkurang hingga 97,66% pada uji pertama, menunjukkan kinerja lebih baik dibandingkan Classic Cloud VPN.

Kata kunci – *Hybrid Cloud*, *Tunneling*, VPN, WireGuard, Classic Cloud VPN

I. PENDAHULUAN

Ketatnya persaingan antarperusahaan dan tuntutan akan perkembangan teknologi yang begitu pesat menjadi sebuah aspek yang mendasari pentingnya pengembangan infrastruktur Teknologi Informasi (TI) dalam ranah korporasi. Infrastruktur IT dalam suatu organisasi atau perusahaan mencakup *software*, *hardware,* dan *service* [1]. Aspek yang sangat berpengaruh dalam kelangsungan operasional sebuah perusahaan saat ini sangat bergantung pada layanan yang disediakan, dalam konteks ini adalah layanan digital yang dimiliki perusahaan. Layanan ini biasanya berjalan pada sebuah perangkat *server* yang dimiliki oleh perusahaan. Setiap perusahaan memiliki pendekatan yang berbeda dalam pengelolaan *server* yang mereka miliki.

Perkembangan teknologi *server* yang semakin pesat telah memunculkan paradigma baru dalam pengelolaan sumber daya komputasi yang saat ini dikenal sebagai teknologi *cloud computing*. Hadirnya teknologi *cloud computing* sudah menjadi salah satu paradigma komputasi yang membarui lanskap teknologi isu dalam beberapa tahun terakhir. *Cloud computing* telah menciptakan peluang baru bagi organisasi dan pengguna individu dalam hal skalabilitas, fleksibilitas, dan efisiensi operasional dengan menyediakan akses ke sumber daya komputasi dan penyimpanan data melalui internet [2]. Dengan adanya infrastruktur *cloud* ini, perusahaan menjadi lebih mudah dalam memanajemen layanan digital yang mereka miliki karena layanan *cloud* memberikan fleksibilitas kepada pengelola untuk memanajemen layanan yang berjalan di dalamnya.

Dalam konteks penelitian ini, jenis *cloud computing* dibagi menjadi tiga, yaitu *public cloud*, *private cloud*, dan *hybrid cloud*. *Public cloud* mengacu pada sebuah layanan *cloud* yang diberikan oleh *provider* besar seperti Google Cloud Platform, AWS, dan Microsoft Azure. Kemudian, aksesibilitas *public cloud* dapat terhubung secara *public* melalui internet, sehingga *user* dapat mengakses melalui internet yang bersifat *public*. Di sisi lain, *private cloud* merupakan layanan *cloud* yang memiliki aksesibilitas terbatas, sehingga tidak semua *user* internet dapat mengaksesnya dengan jaringan internet *public*. Hanya *user* yang terhubung dengan jaringan perusahaan atau organisasi saja yang dapat mengaksesnya. Namun, dalam beberapa skenario, terdapat infrastruktur *cloud* yang menggabungkan penggunaan *public cloud* dan *private cloud*, yang dikenal

sebagai *hybrid cloud*. *Hybrid cloud* memungkinkan *user* untuk memanfaatkan masing-masing kelebihan dari *public* dan *private cloud* seperti keamanan dan kontrol dari *private cloud* serta skalabilitas dan elastisitas dari *public cloud* [3].

Pada umumnya, dalam membangun sebuah infrastruktur *hybrid cloud* memerlukan sebuah interkoneksi untuk menghubungkan antara *public cloud* dengan *private cloud* secara aman. Secara umum untuk membuat sebuah interkoneksi digunakan *tools* VPN atau menggunakan protokol *Internet Protocol Security* (IPsec) agar komunikasi jaringan antarinfrastruktur *cloud* berjalan secara aman (*secure*). Namun, untuk membuat sebuah interkoneksi pada sebuah infrastruktur *hybrid cloud* juga perlu memperhatikan aspek kinerja koneksi dari jalannya komunikasi antara *public cloud* dengan *private cloud* melalui VPN. Kinerja ini sangat memengaruhi skalabilitas layanan yang digunakan, sehingga pemilihan *tools* yang sesuai untuk menghubungkan *public cloud* dan *private cloud* melalui interkoneksi menjadi sangat vital.

Berdasarkan latar belakang tersebut, penelitian ini mengangkat dan menganalisis perbandingan *tools* VPN yang bersifat berbayar (*enterprise*) dan gratis (*open source*), sebagai pertimbangan seorang *cloud architect* dalam mengimplementasikan infrastruktur *hybrid cloud* dengan menyesuaikan kondisi infrastruktur *cloud* perusahaan yang sudah ada. Perbandingan ini difokuskan pada pengukuran kinerja parameter uji seperti *latency*, *throughput*, kestabilan koneksi, kemudahan konfigurasi, dan biaya layanan, untuk memberikan pemahaman yang mendalam tentang kecocokan masing-masing *tools* untuk skenario implementasi tertentu.

II. DASAR TEORI

Penelitian ini mengambil beberapa referensi yang memiliki kesamaan dalam pembahasan infrastruktur *hybrid cloud* dan analisis penggunaan VPN. Penulis mengambil referensi pertama dari penelitian dengan judul "*A Performance Comparison of* WireGuard *and* OpenVPN". Penelitian tersebut berisi pembahasan mengenai perbandingan kinerja antara *tools* VPN dengan menggunakan OpenVPN dengan WireGuard VPN. Dalam melakukan perancangan *deployment* tiap *tools* VPN, menggunakan bantuan *tools* otomasi menggunakan Ansible. Penggunaan Ansible pada penelitian ini digunakan untuk melakukan efisiensi waktu saat melakukan *deployment* ulang pada infrastruktur *cloud* yang berbeda. Dalam penelitian ini, infrastruktur yang digunakan mencakup pengguanaan *public cloud* dari AWS dengan menggunakan empat *Virtual Machine* (VM) dengan empat *region* yang berbeda, serta diimplementasikan pada VM lokal dengan OS Ubuntu dan Centos. Keseluruhan penelitian ini membahas secara jelas analisis perbandingan kinerja *tools* antara Open VPN dan WireGuard VPN dengan parameter uji mencakup *throughput* dan sumber daya komputasi yang berfokus pada parameter CPU *usage*. Konklusi dari penelitian ini menjelaskan bahwa hasil perbandingannya memberikan analisis bahwa *tools* WireGuard VPN lebih unggul dibandingkan dengan OpenVPN [4].

Penelitian selanjutnya terdapat bahasan yang hampir sama seperti dengan penelitian sebelumnya yang membahas perbandingan terkait analaisis kinerja *tools* VPN. Pada yang berjudul "*Performance Analysis of VPN Gateways*" yang lebih difokuskan pada analisis kinerja dari *gateway* VPN dengan membandingkan perangkat lunak dari OpenVPN, Linux IPsec, dan WireGuard VPN. Penelitian ini dilakukan untuk mengevaluasi arsitektur perangkat lunak VPN yang paling efektif dalam mencapai kinerja yang optimal. Hasil penelitian ini menunjukkan bahwa WireGuard merupakan implementasi perangkat lunak VPN yang paling efektif dari segi arsitektur dan memiliki hasil pengujian *throughput* tertinggi dari ketiga implementasi VPN yang uji kinerjanya. Penelitian ini juga mengidentifikasi bahwa *bottleneck* utama dalam penskalaan perangkat lunak VPN terletak pada struktur data dan sinkronisasi *multi-core*, tetapi masalah tersebut dapat diatasi dengan mengadopsi arsitektur berbasis *pipelining* dan *message passing* [5].

Dalam penelitian yang masih membahas di lingkungan VPN, terdapat penelitian dengan judul "*Performance Evaluation of Software Routers with VPN Features*". Penelitian tersebut membahas terkait analisis perangkat lunak *router* dengan fitur VPN berbasis *open source* menggunakan Quagga dan StrongSwan. Penelitian ini bertujuan memvalidasi fungsionalitas dalam lingkungan *real case* dan mengukur kinerja algoritma enkripsi dan *hash* yang didukung oleh StrongSwan untuk memberikan konfigurasi VPN optimal. Hasil dari penelitian ini menunjukkan bahwa integrasi Quagga dan StrongSwan dapat meningkatkan ketahanan terhadap kegagalan koneksi dengan AES GCM sebagai pilihan terbaik untuk kinerja enkripsi yang optimal [6].

Selanjutnya terdapat referensi yang memberikan perspektif langsung pada sebuah implementasi VPN dalam infrastruktur *hybrid cloud*. Penelitian ini berjudul "*CloudJoin: Experimenting at scale with Hybrid Cloud Computing*". Penelitian ini membahas tentang CloudJoin, yaitu sebuah pendekatan transformatif untuk memperluas infrastruktur penelitian komputasi dengan menggabungkan CloudLab dan Google Cloud Platform. CloudJoin memungkinkan eksperimen berskala besar tanpa perlu melakukan perubahan pada infrastruktur yang ada. Penelitian ini menunjukkan bagaimana mengintegrasikan kedua infrastruktur tersebut melalui VPN dan *cloud monitoring tools* untuk mendukung skala eksperimen yang lebih besar. Hasil analisis menunjukkan bahwa CloudJoin memberikan akses mudah ke perangkat keras khusus dan layanan *cloud*, serta memungkinkan eksperimen berjalan dengan lancar tanpa perlu infrastruktur tambahan. Dengan demikian, dapat disimpulkan bahwa CloudJoin adalah pendekatan yang efektif untuk mendukung eksperimen berskala besar dalam penelitian sistem komputer [7].

Referensi berikutnya mengambil penelitian dengan judul "Integrasi *Server On-Premise* dengan *Server Cloud* Menggunakan Cloud VPN dan Mikrotik IPSEC untuk Peningkatan Keamanan Koneksi". Pada penelitian ini dilakukan konfigurasi dengan menggunakan metode MikroTik IPsec di *server on-premise* dan pengaturan VPN Tunnel IKEv2 pada layanan Google Cloud, penelitian ini berhasil mengamankan koneksi antara kedua infrastruktur. Uji coba praktis menunjukkan efektivitas solusi keamanan yang diimplementasikan dalam lingkungan infrastruktur *hybrid*, dengan mengurangi potensi risiko seperti *sniffing*, serangan *port scanning*, *brute force*, DDOS, dan ancaman siber lainnya. Hasil penelitian ini memberikan bukti konkret bahwa penggunaan MikroTik IPsec dan VPN Tunnel IKEv2 pada layanan Google Cloud dapat menjadi langkah yang efektif dalam mengamankan komunikasi antara *server on-premise* dan *server cloud* [8].

Pada referensi berikutnya ditinjau penelitian dengan judul "Desain dan Implementasi *Hybrid Cloud Computing* Sebagai Infrastruktur untuk Analisis *Big Data* Menggunakan *Analytic Hierarchy Process* (AHP)". Penelitian ini bertujuan untuk mengimplementasikan *hybrid cloud computing* sebagai infrastruktur untuk analisis *big data* dengan menggunakan metode *Analytic Hierarchy Process* (AHP) untuk minimalisasi biaya dalam pemilihan *public cloud*. Tahapan dalam penelitian ini meliputi pengumpulan data biaya dari berbagai *public cloud provider*, praproses data untuk konsistensi, pengembangan sistem dengan fitur aplikasi AHP, desain arsitektur sistem, dan implementasi AHP dalam bahasa pemrograman Java. Hasil analisis AHP menunjukkan rekomendasi *public cloud* terbaik untuk digunakan dalam infrastruktur *hybrid cloud*, dengan Digital Ocean sebagai pilihan terpilih [9].

Referensi berikutnya mengambil jurnal penelitian dengan judul "*A Comparative Research on VPN Technologies on Operating System for Routers*". Pada penelitian ini mengulas tentang perbandingan dari tiga *tools* VPN meliputi WireGuard VPN, IPsec, dan SSL-VPN, dengan parameter yang digunakan adalah kompleksitas rancangan infrastruktur VPN dan *throughput*. Penelitian ini dilakukan dengan merancang *router* khusus dan menguji *throughput* masing-masing teknologi VPN menggunakan iPerf3 dalam model jaringan Client-to-Site. Hasil penelitian menunjukkan bahwa WireGuard memiliki *throughput* tertinggi dan paling efisien dalam penggunaan sumber daya CPU dibandingkan dengan SSL-VPN yang memiliki kinerja paling rendah, dan IPsec yang meskipun lebih baik dari SSL-VPN, tetapi masih lebih kompleks dan kurang efisien dibandingkan WireGuard [10].

Referensi selanjutnya mengambil dari jurnal berjudul "*IPSec: Performance Analysis in IPv4 and IPv6*". Penelitian tersebut bertujuan untuk menganalisis kinerja *throughput* protokol IPSec pada jaringan IPv4 dan IPv6 menggunakan berbagai algoritma kriptografi yang direkomendasikan. Penelitian dilakukan dengan mengukur kinerja *throughput* untuk algoritma enkripsi terautentikasi seperti AES-GCM dan AES-CCM, algoritma enkripsi seperti AES-CBC, AES-CTR, dan 3DES, serta algoritma autentikasi seperti SHA1, SHA2, dan XCBC. Hasil penelitian menunjukkan bahwa algoritma AES-GCM memberikan kinerja *throughput* yang lebih baik dibandingkan dengan algoritma kriptografi lainnya yang digunakan dalam implementasi protokol IPSec pada jaringan IPv4 dan IPv6 [11].

Referensi selanjutnya mengambil tinjauan dari jurnal dengan judul "*Impact of IPSec on Real Time Applications in IPv6 and 6to4 Tunneled Migration Network*". Penelitian ini bertujuan untuk menyelidiki dampak implementasi IPSec pada aplikasi *real-time* dalam jaringan IPv6 dan jaringan migrasi *tunneling* 6to4. Penelitian ini menggunakan metode simulasi dengan OPNET Simulator versi 14.5, melalui tiga skenario berbeda yaitu jaringan IPv6 tanpa IPSec, IPv6 dengan IPSec, dan migrasi *tunneling* 6to4 dengan IPSec. Hasilnya menunjukkan bahwa meskipun IPSec berhasil memberikan keamanan yang diperlukan, terdapat peningkatan *delay*, *jitter*, dan *packet drop rate* yang signifikan, yang mempengaruhi kinerja aplikasi *real-time* [12].

Referensi berikutnya penulis meninjau jurnal judul penelitian "*Security vs Bandwidth: Performance Analysis Between IPsec and OpenVPN in Smart Grid*", memaparkan penelitian terkait analisis pengaruh enkripsi terhadap *bandwidth* antara IPsec dengan OpenVPN. Skenario pengujian dilakukan dengan mengatur koneksi IPsec dan OpenVPN melalui jaringan *public* LTE menggunakan router industri dan menganalisis *overhead* yang ditambahkan oleh masing-masing metode enkripsi terhadap paket data. Hasil penelitian menunjukkan bahwa IPsec menambahkan rata-rata 64 *byte* per paket dan OpenVPN menambahkan rata-rata 42 *byte* per paket, dengan IPsec membutuhkan sekitar 23% lebih banyak *bandwidth* dibandingkan OpenVPN. Penelitian ini memberikan wawasan penting tentang kebutuhan *bandwidth* dalam implementasi metode enkripsi pada jaringan komunikasi *smart grid* [13].

Referensi selanjutnya merujuk pada penelitian berjudul *"CloudJoin: Experimenting at Scale with Hybrid Cloud Computing"*. Penelitian ini menekankan pentingnya pengembangan *testbed* eksperimen yang terintegrasi dan skalabel antara CloudLab dan Google Cloud Platform (GCP), tanpa memerlukan perubahan mendasar pada infrastruktur yang sudah ada. Penelitian ini menghasilkan pendekatan integrasi infrastruktur eksperimen melalui mekanisme *peer-to-peer Virtual Private Network* (VPN) menggunakan StrongSwan, serta pemanfaatan alat *observability* (alat *monitoring*) seperti Google Stackdriver yang diintegrasikan dengan agen BindPlane dan Collectd. Tujuan utama dari penelitian ini adalah untuk menawarkan solusi *hybrid cloud* yang mampu menjawab tantangan skalabilitas dan keterbatasan sumber daya dalam eksperimen sistem komputasi skala besar. Manfaat yang dihadirkan mencakup penyediaan pendekatan riset komputasi yang lebih inklusif, adaptif, serta berorientasi masa depan (*future-oriented*),

terutama dalam menjawab dinamika kebutuhan sumber daya yang terus meningkat di lingkungan penelitian akademik [14].

Pada referensi terakhir yang ditinjau penulis, yaitu penelitian berjudul *"Comparative Analysis of Optimizing Hybrid Cloud Environments Using AWS, Azure, and GCP"*, dipaparkan bahwa pemilihan penyedia layanan *public cloud* yang tepat merupakan aspek krusial dalam merancang infrastruktur *hybrid cloud* yang optimal. Permasalahan utama yang diangkat dalam penelitian ini berkaitan dengan meningkatnya kebutuhan organisasi untuk mengadopsi arsitektur *hybrid cloud* demi mencapai fleksibilitas, efisiensi biaya, dan peningkatan skalabilitas. Namun, proses adopsi ini tidak dapat dilakukan secara langsung tanpa adanya kajian komparatif terhadap masing-masing *platform cloud*, guna memastikan kesesuaian dengan karakteristik dan kebutuhan spesifik setiap organisasi. Hasil penelitian ini memberikan solusi aplikatif bagi seorang *solution architect* dalam menyelaraskan strategi penerapan layanan cloud dengan kebutuhan organisasi. Dengan demikian, studi ini menjadi kontribusi signifikan bagi akademisi di bidang teknologi informasi maupun praktisi industri, khususnya *solution architect*, dalam memahami serta mengoptimalkan penerapan infrastruktur *hybrid cloud* sebagai fondasi transformasi digital yang berkelanjutan [15].

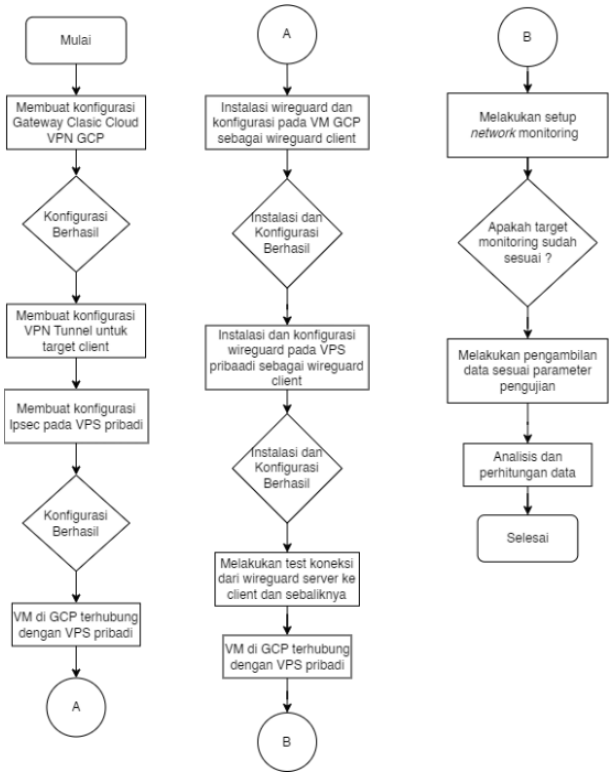
Dari sepuluh referensi yang diambil, penelitian ini memiliki keterbaruan dalam konteks jenis *tools* yang dibandingkan, di mana dalam penelitian ini penulis membandingkan Classic Cloud VPN dari produk GCP yang belum pernah dibandingkan dengan WireGuard VPN. Selanjutnya, terdapat perbedaan dalam implementasi yang diterapkan. Dalam konteks penelitian ini, penulis menerapkan implementasi pada infrastruktur *hybrid cloud*. Terakhir, terdapat keterbaruan dalam skenario pengujian yang dilakukan. Dalam penelitian ini, penulis melakukan skenario dengan menggunakan metode *rsync* untuk menguji performa interkoneksi gateway VPN dari kedua *tools* VPN yang diimplementasikan pada infrastruktur *hybrid cloud*. Detail skenario yang dilakukan adalah perintah *rsync* digunakan untuk mengirim *file dataset* dengan berbagai ukuran, mulai dari kecil hingga besar. Pengiriman dilakukan dengan skenario pengiriman dari *private cloud* ke *public cloud*, *public cloud* ke *private cloud*, dan terakhir adalah pengiriman *file* antar lingkungan *public cloud*.

III. METODOLOGI

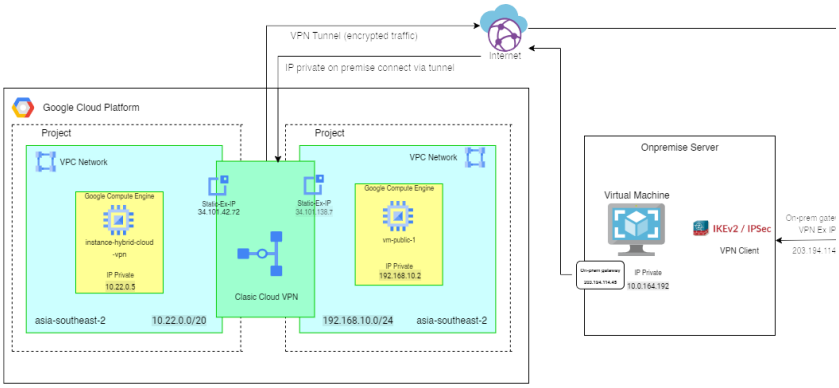
Dalam melaksanakan penelitian ini terdapat beberapa tahapan penelitian yang dilakukan oleh penulis. Detail tahapan penelitian dapat dilihat dalam diagram alir pada Gambar 1.

Gambar 1. Diagram alir penelitian

Setelah menentukan tahapan penelitian yang dikerjakan, penulis melanjutkan untuk melakukan tahapan perancangan sistem. Detail tahapan perancangan sistem yang dilakukan dapat dilihat pada Gambar 2.



Gambar 2. Diagram alir perancangan sistem



Gambar 3. Topologi *Hybrid Cloud Classic Cloud* VPN

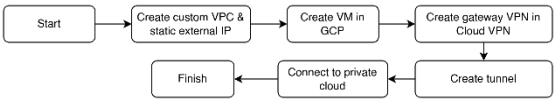
A. Desain Infrastruktur *Hybrid Cloud*

Penelitian ini memiliki dua rancangan infrastruktur *hybrid cloud* dengan perbedaan dalam metode *tunneling* yang digunakan. Pada infrastruktur yang pertama diimplementasikan metode *tunneling* menggunakan Classic Cloud VPN. Classic Cloud VPN merupakan fitur VPN yang memang diberikan Google Cloud untuk kebutuhan interkoneksi yang salah satu tujuanya dibuat untuk diimplementasikan pada arsitektur *hybrid cloud*. Protokol yang dipakai pada Classic Cloud VPN ini menggunakan protokol IPSec/IkeV2. Berikut adalah detail desain topologi pada rancangan infrastruktur *hybrid cloud* menggunakan Classic Cloud VPN tertera pada Gambar 3.

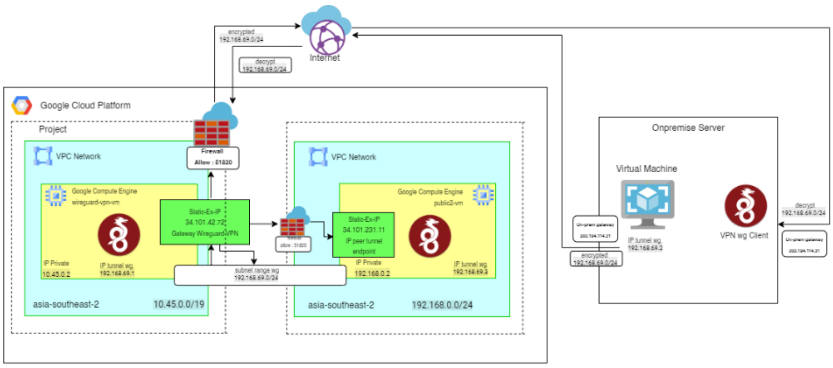
Berikutnya pada infrastruktur kedua diimplementasikan metode *tunneling* menggunakan WireGuard VPN. WireGuard VPN memiliki protokol independen yang juga dinamai sebagai protokol WireGuard. Protokol ini memiliki keunggulan dalam hal kecepatan koneksi, lalu menurut beberapa sumber protokol ini juga ringan jika diimplementasikan di dalam sebuah *server*. Berikut merupakan detail desain arsitektur *hybrid cloud* menggunakan WireGuard VPN tertera pada Gambar 4.

B. *Setup* Infrastruktur *Hybrid Cloud* VPN GCP

Tahap awal dalam membangun infrastruktur *hybrid* menggunakan Cloud VPN di Google Cloud Platform (GCP) dimulai dengan pembuatan *Virtual Private Cloud* (VPC) sebagai dasar pemisahan jaringan untuk layanan Google Compute Engine. Pada tahap ini juga dilakukan pembuatan *static external public IP address* yang akan digunakan sebagai *endpoint* atau *gateway* untuk koneksi Cloud VPN. Setelah konfigurasi *basic networking* selesai, proses dilanjutkan dengan pembuatan *virtual machine* (VM) yang dikaitkan dengan subnet sesuai dengan topologi *custom* VPC yang telah dirancang. Selanjutnya, dilakukan konfigurasi Cloud VPN Gateway menggunakan alamat *static IP* yang telah disiapkan sebelumnya. Tahapan ini diikuti dengan pembuatan VPN *tunnel* yang menghubungkan *gateway* di sisi GCP dengan jaringan di sisi *private cloud*. Konfigurasi terakhir mencakup pengaturan klien VPN pada sisi *private cloud* agar dapat menjalin koneksi terenkripsi secara stabil dengan jaringan di GCP. Detail arsitektur *setup* dapat dilihat pada Gambar 5.



Gambar 5. Diagram alir setup infrastruktur *hybrid cloud* VPN GCP

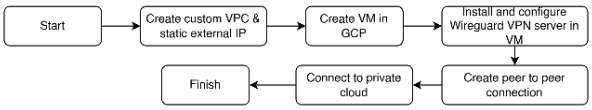


Gambar 4. Topologi *Hybrid Cloud* WireGuard VPN

C. *Setup* Infrastruktur *Hybrid* WireGuard VPN

Perancangan infrastruktur *hybrid cloud* menggunakan metode *tunneling* WireGuard VPN pada tahap awal memiliki alur yang serupa dengan penerapan Classic Cloud VPN, khususnya dalam hal penyiapan *project* di Google Cloud Platform (GCP). Langkah-langkah awal yang dilakukan meliputi aktivasi *billing account*, pembuatan VPC, serta penyusunan *virtual machine* (VM) di lingkungan *public cloud* GCP. Namun, perbedaan utama terletak pada pendekatan implementasi VPN-nya. Pada arsitektur WireGuard, VM di GCP berfungsi langsung sebagai *server* VPN. Ini berbeda dengan Classic Cloud VPN, di mana layanan VPN disediakan secara terpisah dari VM melalui layanan terkelola milik GCP. Dalam skema WireGuard, komunikasi antar jaringan dilakukan melalui alokasi *private* IP *range* yang didefinisikan secara manual, dengan dukungan fitur IP *forwarding*. Dengan demikian, layanan WireGuard berjalan langsung di dalam VM, dan menggunakan *external IP address* dari VM tersebut sebagai *gateway* komunikasi VPN.

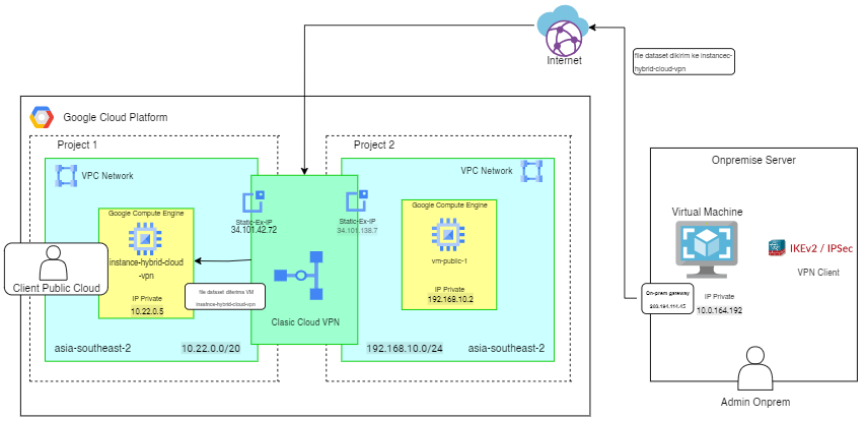
Setelah konfigurasi dasar WireGuard *server* pada VM GCP selesai dilakukan, langkah selanjutnya adalah melakukan konfigurasi pada sisi klien, yaitu di lingkungan *private cloud*. Konfigurasi pada sisi client secara umum mengikuti struktur yang sama seperti pada *server*. Pada kedua sisi, dilakukan pertukaran informasi melalui *peer configuration*, yang memerlukan *public key* dari masing-masing VM serta informasi subnet *private IP range* yang telah ditentukan sebelumnya di sisi *server*. Ilustrasi lengkap mengenai setup infrastruktur *hybrid* dengan WireGuard VPN dapat dilihat pada Gambar 6.



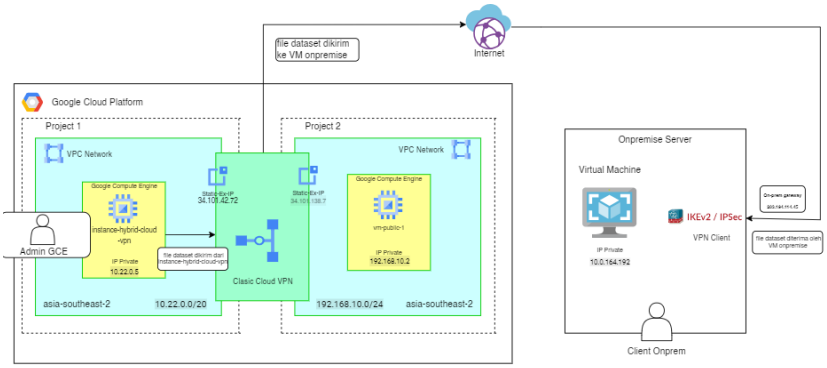
Gambar 6. Diagram alir *setup* infrastruktur *hybrid* WireGuard VPN

D. Skenario Pengujian

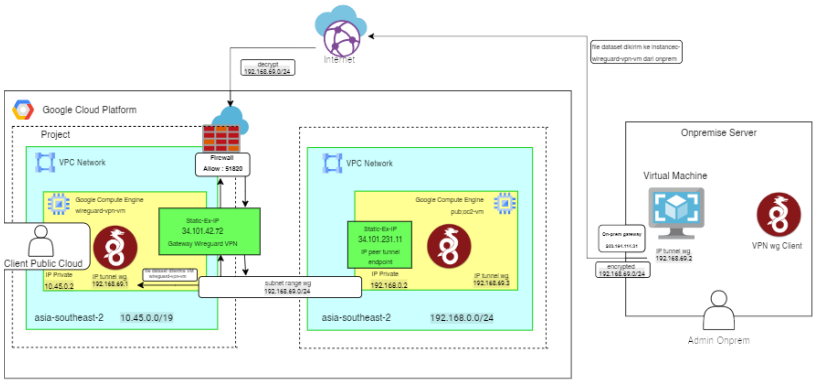
Pada pengerjaan penelitian dilakukan pengujian dengan berfokus pada pengukuran parameter QoS dengan target setiap *gateway* VPN dari kedua infrastruktur *hybrid cloud*. Pengujian dilakukan untuk melakukan pengambilan data metrics terkait data *bytes*, data *packet*, dan detail waktu dari pengiriman kedua data tersebut. Pengujian pertama dilakukan dengan melakukan skenario pengiriman *file dataset* dari lingkungan *private cloud* ke *public cloud*. Lalu selanjutnya pengiriman dilakukan dari lingkungan *public cloud* ke *private*



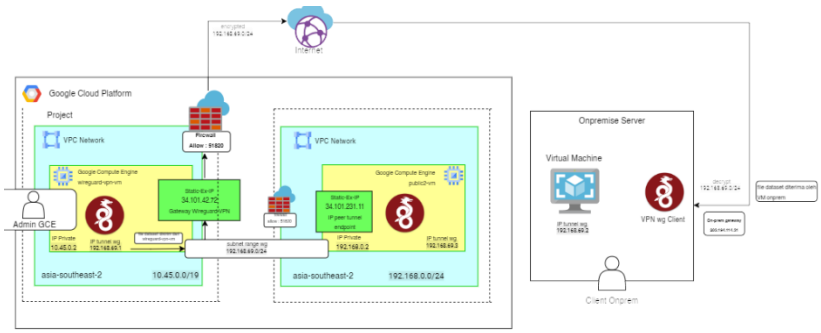
Gambar 7. Skenario pengujian 1 Cloud VPN



Gambar 8. Skenario pengujian 2 Cloud VPN



Gambar 9. Skenario pengujian 1 WireGuard VPN



Gambar 10. Skenario pengujian 2 WireGuard VPN

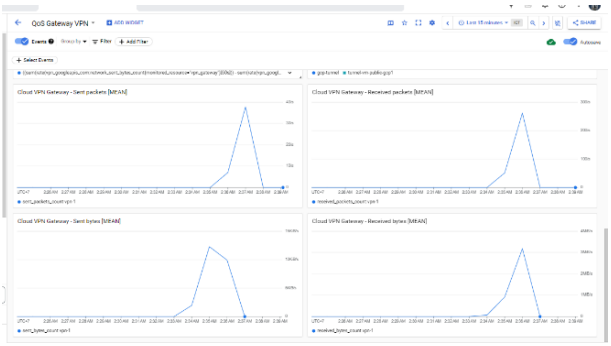
cloud. Secara detail alur skenario pengujian dapat dilihat pada Gambar 7 dan 8.

Pengujian pada lingkungan infrastruktur *hybrid cloud* yang menggunakan WireGuard VPN sebagai alat interkoneksi untuk menghubungkan *private cloud* dengan *public cloud* secara keseluruhan memiliki skenario yang sama dengan pengujian yang dilakukan pada infrastruktur *hybrid cloud* yang menggunakan Classic Cloud VPN dari GCP. Detail skenario pengujian dapat dilihat dari Gambar 9 dan 10.

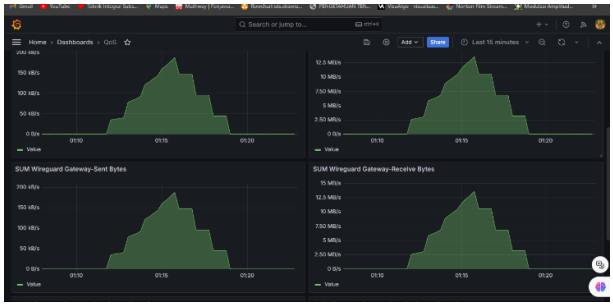
Skenario pengujian yang dilakukan berfokus pada pengukuran *traffic* yang masuk melalui *gateway* Cloud VPN. Tujuan dari pengukuran *traffic* adalah untuk mengetahui seberapa cepat kinerja interkoneksi pada infrastruktur *hybrid cloud*. Selain itu, pengujian ini juga bertujuan untuk mengetahui seberapa besar beban *traffic* yang mampu di tangani oleh VPN, melalui kecepatan data yang masuk pada *gateway* VPN. Skenario pengujian dilakukan dengan mengirim *file* secara berulang menggunakan variasi ukuran *file dataset* yang berbeda, mulai dari ukuran kecil, sedang, hingga ukuran besar. Pengujian juga dilakukan dengan menjalankan skenario pengiriman *file dataset* dari lingkungan cloud yang berbeda. Detail pengujian dapat dilihat pada Gambar 7 hingga 10, yang memberikan informasi terkait alur pengiriman *file dataset*. Pengujian dimulai dengan mengirim *file dataset* dari *private cloud* ke *public cloud*, kemudian dilanjutkan dengan mengirim dari lingkungan *public* ke *private*. Pengiriman *file* yang dilakukan menggunakan metode *remote sync* atau disingkat *rsync*, dengan metode ini memungkinkan *cloud* melakukan pengirirman *file* secara *remote*.

E. Perhitungan *Quality of Service* (QoS)

Setelah mengirim *file* menggunakan metode *rsync*, tahapan terakhir adalah pengambilan data *traffic* interkoneksi VPN pada *dashboard monitoring*. Fungsi dari monitoring ini adalah mencatat data metrik terkait data dan paket yang dikirim dan diterima melalui *gateway* VPN, yang nantinya digunakan untuk analisis hasil perbandingan. Detail gambar *dashboard* visualisasi data dapat dilihat pada Gambar 11 dan 12.



Gambar 11. *Dashboard cloud* monitoring GCP



Gambar 12. *Dashboard* WireGuard VPN

Selanjutnya, proses pengambilan data dilakukan dengan masuk ke *dashboard* monitoring dan mengunduh data CSV dari salah satu visualisasi data yang ditampilkan di dalam *dashboard*. Contoh pada salah satu data mentah dari pengambilan nilai *bytes* dan *packet* saat dilakukan pengujian tertera pada Tabel 1 dan 2.

Tabel 1. *Receive bytes* Cloud VPN

Waktu	*Receive Bytes*
6/24/2024 06:12:00	34,225
6/24/2024 06:13:00	6490566,817
6/24/2024 06:14:00	11712539,69
6/24/2024 06:15:00	149019,4
Total	18352160,13
Rata-Rata	4588040,033

Tabel 2. *Receive bytes* WireGuard VPN

Waktu	*Receive Bytes*
6/24/2024 5:48	7833385,917
6/24/2024 5:49	8307402,961
6/24/2024 5:49	8781420,01
6/24/2024 5:49	9255437,05
6/24/2024 5:49	7833385,92
6/24/2024 5:50	8307402,96
6/24/2024 5:50	8781420,01
6/24/2024 5:50	9255437,05
6/24/2024 5:50	7833385,92
6/24/2024 5:51	8307402,96
6/24/2024 5:51	8781420,01
6/24/2024 5:51	9255437,05
6/24/2024 5:51	7833385,92
6/24/2024 5:52	8307402,96
6/24/2024 5:52	8781420,01
6/24/2024 5:52	9255437,05
Total	4334773,2
Rata-Rata	1444924,4

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengukuran *Throughput*

Pengujian untuk mengukur *throughput* dilakukan untuk mengetahui seberapa cepat data dikirimkan dan diterima oleh *client* yang terhubung melalui jaringan VPN. Parameter uji ini sangat penting untuk kebutuhan operasional perusahaan atau organisasi yang menerapkan infrastruktur *hybrid cloud*. Dari hasil pengujian ini, diambil sembilan nilai *throughput* dari masing-masing skenario pengujian pada penerapan VPN sebagai interkoneksi *hybrid cloud*.

Dari hasil pengujian *throughput* yang dilakukan pada skenario 1 (pengiriman *file dataset* dari *private* ke *public*) yang dapat dilihat pada Tabel 4, perbandingan rata-rata *throughput* menunjukkan bahwa nilai *throughput* WireGuard VPN lebih tinggi dibandingkan dengan Classic Cloud VPN. Namun, jika mengacu pada Tabel 3 hasil uji pengujian pada setiap pengiriman *file dataset*, terlihat bahwa pengiriman *file dataset* yang berukuran kecil dan sedang, Classic Cloud VPN memiliki nilai *throughput* yang sedikit lebih tinggi dibandingkan dengan WireGuard VPN. Pada hasil uji pada data Tabel 3 juga terlihat bahwa waktu yang diperlukan dalam melakukan pengiriman *file* dengan ukuran 500 MB dan 1GB Classic Cloud VPN cenderung lebih cepat dibandingkan dengan WireGuard VPN.

Tabel 3. Hasil uji *throughput* skenario 1

Ukuran *File*	Jenis VPN	Waktu Pengiriman (s)	Total *Throughput (kbps)*
500 MB	Classic Cloud VPN	180	192,66
	WireGuard VPN	300	955,38
1 GB	Classic Cloud VPN	240	295,87
	WireGuard VPN	360	1629,2
2 GB	Classic Cloud VPN	240	611,74
	WireGuard VPN	240	4557,02

Setelah dilakukan percobaan dengan *file dataset* berukuran besar, nilai *throughput* WireGuard VPN mengalami peningkatan signifikan, mencapai dua kali lipat dibandingkan dengan Classic Cloud VPN. Oleh karena itu, saat dilakukan perhitungan total dan rata-rata *throughput* pada skenario pertama ini, nilai *throughput* WireGuard VPN lebih tinggi dibandingkan dengan Classic Cloud VPN.

Pengujian perbandingan uji *throughput* pada skenario ke-2 (pengiriman *file dataset* dari *public* ke *private*), didapatkan sebuah hasil perhitungan total dari jumlah total *throughput* serta rata-rata *throughput* secara keseluruhan nilai *throughput* yang didapatkan dari kedua VPN memiliki nilai *throughput* yang lebih kecil dibandingkan pada rata-rata *throughput* pada pengujian di skenario 1. Pada perbandinganya nilai rata-rata *throughput* yang didapatkan WireGuard VPN memiliki nilai yang jauh lebih tinggi dibandingkan dengan nilai *throughput* Classic Cloud VPN. Untuk detail pengujian *throughput* mengacu pada Tabel 4 dari pengiriman *file* kecil hingga besar WireGuard VPN memiliki kinerja yang sangat jauh dibandingkan dengan Classic Cloud VPN, karena

dibandingkan pada hasil uji pada skenario 1, nilai *throughput* WireGuard VPN selalu unggul dari Classic Cloud VPN dengan selisih nilai yang jauh pada setiap pengiriman *file* dengan ukuran yang berbeda.

Tabel 4. Hasil uji *throughput* skenario 2

Ukuran *File*	Jenis VPN	Waktu Pengiriman (s)	Total *Throughput (kbps)*
500 MB	Classic Cloud VPN	240	0,35
	WireGuard VPN	240	3,09
1 GB	Classic Cloud VPN	300	0,51
	WireGuard VPN	300	4,6
2 GB	Classic Cloud VPN	180	1,54
	WireGuard VPN	300	10,95

Meskipun WireGuard VPN menunjukkan *throughput* yang lebih tinggi pada setiap skenario pengujian transfer *file*, hasil waktu tempuh pengiriman justru menunjukkan bahwa Classic Cloud VPN kerap menyamai atau bahkan mengungguli WireGuard. Anomali ini dipengaruhi oleh beberapa faktor teknis, salah satunya adalah tingginya beban CPU pada VM spesifikasi rendah akibat penggunaan enkripsi ChaCha20-Poly1305 berbasis perangkat lunak pada WireGuard, yang menciptakan *bottleneck* hingga 95% utilisasi CPU. Sebaliknya, Classic Cloud VPN memanfaatkan AES-256 dengan *hardware acceleration*, yang mampu mengurangi beban CPU hingga 70% dan meningkatkan efisiensi proses enkripsi-dekripsi.

Selain itu, *overhead* protokol WireGuard menyebabkan penurunan MTU efektif menjadi 1420 *byte*, memicu fragmentasi paket yang berdampak pada waktu pengiriman. WireGuard juga mengandalkan *routing* statis, yang menambah RTT (*Round-Trip Time*) hingga 40 ms, berbeda dengan Classic Cloud VPN yang memanfaatkan *dynamic* BGP *routing* dan infrastruktur GCP yang dioptimalkan untuk TCP, seperti *auto-tuning window size* dan *congestion control*. Tanpa konfigurasi manual terhadap parameter TCP (misalnya RWIN atau BBR), WireGuard cenderung kurang efisien dalam kondisi jaringan kompleks. Hal ini menunjukkan bahwa *throughput* tinggi tidak selalu merepresentasikan kecepatan transfer *file* yang optimal jika tidak diimbangi dengan efisiensi komputasi dan infrastruktur jaringan yang memadai.

B. Hasil Pengukuran *Packet Loss*

Pengujian *packet loss* dilakukan untuk menghitung jumlah paket yang hilang selama melakukan pengiriman *file dataset* antar lingkungan *cloud*. Jika terdapat nilai presentase *packet loss* yang tinggi perlu ditindak lanjuti apakah terdapat *miss*-konfigurasi pada infrastruktur interkoneksi yang dirancang. Pengujian *packet loss* yang dilakukan pada skenario 1 (pengiriman *file dataset* dari *private* ke *public*) menunjukkan bahwa kedua VPN memiliki persentase *packet loss* sebesar 0%. Hasil ini menunjukkan bahwa kinerja kedua VPN dalam mengirimkan data tidak mengalami kehilangan paket. Oleh karena itu, uji *packet loss* pada skenario 1 memberikan indikator bahwa kedua VPN memiliki kinerja interkoneksi yang sama baiknya, jika ditinjau dari nilai *packet loss* pada skenario pengujian 1.

Tabel 5. Hasil uji *packet loss* skenario 1

Ukuran *File*	Classic Cloud VPN (%)	WireGuard VPN (%)	Selisih
500 MB	0	0	0
1 GB	0	0	0
2 GB	0	0	0

Pengujian perbandingan *packet loss* pada skenario ke-2 (pengiriman *file dataset* dari *public* ke *private*) menunjukkan bahwa kedua VPN mengalami kenaikan persentase *packet loss*. Berdasarkan hasil perhitungan jumlah total dan rata-rata *packet loss* dari kedua VPN, yang ditampilkan pada tabel perhitungan *packet loss* skenario 2, terlihat bahwa Classic Cloud VPN memiliki keunggulan karena persentase *packet loss* yang lebih rendah dibandingkan dengan WireGuard VPN. Detail lengkap persentase *packet loss* pada setiap pengiriman *file* menunjukkan bahwa Classic Cloud VPN memang mengungguli WireGuard VPN dengan memberikan nilai persentase *packet loss* yang lebih kecil. Namun, pada Tabel 6 terlihat bahwa sebenarnya selisih persentase *packet loss* antara kedua VPN tidak terlalu jauh.

Tabel 6. Hasil uji *packet loss* skenario 2

Ukuran *File*	Classic Cloud VPN (%)	WireGuard VPN (%)	Selisih
500 MB	0,94	0,96	0,02
1 GB	0,95	0,97	0,02
2 GB	0,95	0,96	0.01

C. Hasil Pengukuran *Latency*

Pengujian *latency* dilakukan untuk mengukur waktu yang diperlukan bagi sebuah paket atau data untuk dikirimkan dari *host* menuju *destination*. Parameter *latency* menjadi sangat penting sebagai tolak ukur kinerja sebuah infrastruktur jaringan yang berjalan dengan lancar. Jika pengujian *latency* menunjukkan angka yang kecil pada sebuah infrastruktur *hybrid cloud*, maka dapat disimpulkan bahwa infrastruktur interkoneksi pada *hybrid cloud* tersebut sangat baik dalam mendukung operasional perusahaan.

Hasil uji *latency* pada skenario 1 (pengiriman *file dataset* dari *private* ke *public*) menunjukkan perbedaan yang sangat kontras antara WireGuard VPN dan Classic Cloud VPN.

Perhitungan jumlah total dan rata-rata *latency* menunjukkan selisih yang cukup signifikan. Tabel perhitungan menunjukkan bahwa WireGuard VPN memiliki kinerja yang sangat baik dengan total semua *latency* pengiriman *file* sebesar 0,0184 ms. Hasil ini mengindikasikan bahwa WireGuard VPN memiliki kinerja interkoneksi yang sangat baik dalam pengujian skenario 1. Secara detail, semakin besar *file* yang dikirim, semakin kecil *latency* yang dihasilkan pada infrastruktur *hybrid cloud* yang menggunakan WireGuard VPN. Hasil pada pengujian ini menegaskan bahwa WireGuard VPN mampu mempertahankan kinerja *latency* yang optimal bahkan saat mengirim *file* berukuran besar.

Tabel 7. Hasil uji *latency* skenario 1

Ukuran *File*	Classic Cloud VPN (ms)	WireGuard VPN (ms)	Reduksi *Latency* (%)
500 MB	0,38	0,01	97,4
1 GB	0,28	0,006	97,9
2 GB	0,13	0,0024	98,1

Selanjutnya, uji *latency* pada skenario 2 (pengiriman *file dataset* dari *public* ke *private*) memberikan hasil yang kurang memuaskan dari sisi Classic Cloud VPN. Ditinjau dari tabel perhitungan *latency* skenario 2, jumlah total dan rata-rata *latency* WireGuard VPN masih dapat mempertahankan kinerjanya dengan *latency* tidak melebihi 1 ms. Sebaliknya, pada Classic Cloud VPN, rata-rata *latency* meningkat menjadi 6,7 ms. Secara detail, pada setiap pengiriman *file*, WireGuard VPN mampu menjaga konsistensi kinerja interkoneksinya, mirip dengan skenario 1, di mana hasil *latency* akan semakin kecil saat ukuran *file* yang dikirimkan semakin besar. Meskipun Classic Cloud VPN juga menunjukkan kinerja yang relatif mirip dengan memberikan hasil *latency* yang semakin kecil seiring dengan bertambahnya ukuran *file* yang dikirimkan, hasil *latency* yang ditunjukkan masih jauh dibandingkan dengan kinerja WireGuard VPN.

Tabel 8. Hasil uji *latency* skenario 2

Ukuran *File*	Classic Cloud VPN (ms)	WireGuard VPN (ms)	Reduksi *Latency* (%)
500 MB	10,4	0,28	97,3
1 GB	7,55	0,22	97,08
2 GB	2,19	0,08	96,34

D. Hasil Analisis QoS

Dalam hasil pengujian pada setiap skenario untuk masing-masing parameter *Quality of Service* (QoS), ditemukan bahwa WireGuard VPN menunjukkan kinerja interkoneksi yang lebih unggul dibandingkan dengan Classic Cloud VPN. Namun, pada parameter *packet loss*, Classic Cloud VPN sedikit unggul dibandingkan dengan WireGuard VPN, dengan selisih presentase *packet loss* yang sangat kecil. Secara keseluruhan, saat melakukan pengujian menggunakan kedua *tools* VPN, kinerja yang terbilang maksimal terjadi saat dilakukan skenario pengiriman *file* dengan *private cloud* bertindak sebagai pengirim, dibandingkan dengan *public cloud* yang bertindak sebagai pengirim. Fenomena ini terjadi karena bentuk infrastruktur yang dibuat, di mana pemasangan VPN pada lingkungan *public cloud* mengharuskan data keluar dari *public cloud*, melalui internet, masuk ke *server* VPN, dan kemudian masuk ke *private cloud*, yang dapat menyebabkan *latency* dan *bottleneck* tambahan.

WireGuard VPN, menjadi salah satu *tools* VPN terbaik dalam hal kecepatan pengiriman data, menunjukkan kinerja yang lebih baik. Hal ini disebabkan oleh fakta bahwa WireGuard masih merupakan teknologi yang relatif baru, sehingga teknologi yang digunakannya mendukung kinerja WireGuard sebagai *tools* VPN yang memiliki kinerja di atas rata-rata. Di sisi lain, Classic Cloud VPN menggunakan protokol IPSec/IkeV yang telah ada sejak lama. Jadi dari hasil yang diperoleh dapat disimpulkan bahwa kinerja interkoneksi yang ditawarkan oleh WireGuard VPN lebih sesuai untuk diimplementasikan dalam kebutuhan operasional perusahaan yang menggunakan *hybrid cloud* untuk kebutuhan migrasi secara berkala. Hal ini karena WireGuard mampu menjaga stabilitas infrastruktur *hybrid*, terutama saat terjadi lonjakan *traffic*.

E. Perbandingan Harga dan Fleksibiltas Konfigurasi

Pada perbandingan QoS kedua VPN, terlihat bahwa kinerja WireGuard mengungguli Classic Cloud VPN. Selain perbandingan kinerja, penggunaan VPN juga dipengaruhi oleh harga dan fleksibilitas konfigurasi. Perbandingan pertama terkait harga sangat jelas: Classic Cloud VPN adalah alat VPN berbayar dari Google Cloud, sedangkan WireGuard VPN merupakan VPN *open source* yang bersifat gratis jadi perusahaan dan perorangan dapat memakainya tanpa mengeluarkan biaya. Detail biaya Classic Cloud VPN berdasarkan pricing calculator serta VM dari Google Cloud Platform dapat dilihat pada Tabel 9 dan Tabel 10.

Tabel 9. Daftar harga Classic Cloud VPN GCP

Jumlah *Tunnel*	Harga (*region* asia-southeast2)
1	$43,76/ bulan
2	$87,53/ bulan
3	$131,29/ bulan
4	$175,05/ bulan

Tabel 10. Perhitungan harga VM WireGuard *server*

Spesifikasi VM WireGuard *Server*	Harga ($)/bulan	Total Harga ($)/bulan
E2-small (1 shared core vCPU + 2GiB Memory)	16,44	18,39
Balanced Presistend Disk 15 GB	1,95	

Dari segi struktur pembiayaan, tabel perbandingan menunjukkan bahwa terdapat perbedaan mendasar dalam parameter penentuan biaya pada implementasi VPN. Pada layanan Classic Cloud VPN milik GCP, biaya dihitung berdasarkan jumlah *VPN tunnel* yang aktif di masing-masing region, sesuai dengan skema layanan yang telah disediakan oleh GCP. Sementara itu, penggunaan WireGuard VPN bersifat *open source* dan memerlukan *self-hosting*, sehingga komponen biayanya bergantung pada sumber daya *virtual machine (VM)* yang diperkirakan secara mandiri di *platform* GCP. Berdasarkan hasil perhitungan dan analisis parameter biaya pada kedua pendekatan tersebut, dapat disimpulkan bahwa Classic Cloud VPN kurang direkomendasikan bagi perusahaan atau organisasi yang masih dalam tahap berkembang, dengan rancangan infrastruktur yang masih sederhana dan perlu mengalokasikan dana seefektif mungkin. Alternatif seperti WireGuard lebih sesuai karena memberikan fleksibilitas biaya yang lebih tinggi, meskipun membutuhkan pengelolaan infrastruktur secara mandiri.

Pada sisi fleksibilitas, tidak ada kendala signifikan dalam melakukan konfigurasi perancangan kedua VPN. Namun, ada beberapa hal yang perlu digarisbawahi terkait kebijakan masing-masing VPN. Saat melakukan konfigurasi, penulis menemukan bahwa Classic Cloud VPN tidak menerima semua *traffic* masuk dari layanan di luar infrastruktur GCP. Dalam hal ini, *private cloud* yang terhubung dengan *public cloud* GCP tetap memiliki akses terbatas untuk menjalankan layanan yang perlu terintegrasi langsung dengan produk GCP. Analisis ini didasarkan pada pengalaman saat perancangan infrastruktur di mana IP *gateway* Classic Cloud VPN menolak untuk terhubung dengan layanan node_exporter sebagai *agent metrics* untuk kebutuhan monitoring. Kemungkinan, terdapat beberapa konfigurasi keamanan dari sisi GCP yang perlu disesuaikan lebih lanjut. Selain itu, dalam konfigurasi *tunnel*, Classic Cloud VPN hanya menerima IP publik untuk dapat terhubung pada *remote peer address*, sementara untuk koneksi private diperlukan konfigurasi tambahan pada cloud router GCP. Oleh karena itu, dalam hal fleksibilitas koneksi, WireGuard VPN lebih mudah diimplementasikan.

V. SIMPULAN

Setelah menguji konfigurasi VPN, penulis menemukan bahwa Classic Cloud VPN dari GCP jauh lebih rumit dibandingkan dengan WireGuard VPN yang *open source*. Kompleksitas ini muncul karena protokol IPSec/IkeV2 yang digunakan serta kebijakan Google Cloud yang ketat. Meskipun demikian, Classic Cloud VPN punya kelebihan dalam hal pemeliharaan karena otomatis terhubung dengan Cloud Logging GCP yang memudahkan pelacakan masalah *traffic* VPN serta sangat cocok dalam infrastruktur yang lebih kompleks.

Hasil pengujian menunjukkan bahwa WireGuard VPN lebih unggul dalam kinerja. WireGuard mencatat peningkatan *throughput* hingga 676,67% pada uji kedua dan pengurangan *latency* hingga 97,66% pada uji pertama. Ukuran *file* juga berpengaruh signifikan terhadap *throughput*, dengan peningkatan hingga 377,02% saat menggunakan *dataset* besar dibandingkan dengan *dataset* kecil. Namun, lingkungan cloud tempat pengiriman *file* juga mempengaruhi hasil ini.

Penulis merekomendasikan WireGuard VPN untuk perusahaan atau organisasi yang masih berkembang dengan infrastruktur yang masih sederhana karena kinerjanya yang bagus dan sifat *open source*-nya. Namun, untuk perusahaan atau organisasi besar dan sudah mengimplementasikan infrastruktur yang lebih kompleks, perlu diperhatikan bahwa skalabilitas WireGuard masih terbatas karena teknologinya masih baru dan fitur manajemennya belum matang. Untuk penelitian pada masa yang akan datang, penulis menyarankan penambahan parameter pengujian seperti utilitas *node*, perhitungan *jitter*, dan total *bandwidth*. Penggunaan alat pemantauan jaringan yang konsisten serta pengujian QoS dalam skenario *streaming* dan *load balancing* juga disarankan untuk pemahaman lebih mendalam tentang performa VPN.

REFERENSI

[1] N. I. Fitria D, Nurisnaini Putri, and Putri Zahrani, "LITERATURE REVIEW DETERMINASI INFRASTRUKTUR TI: TELEKOMUNIKASI, INTERNET DAN BRAINWARE," *J. Manaj. Pendidik. DAN ILMU Sos.*, vol. 3, no. 2, pp. 561–572, 2022, doi: 10.38035/jmpis.v3i2.1119.

[2] D. Nafis Alfarizi and I. Heidiani Ikasari, "Tinjauan Literatur Terhadap Pemanfaatan Cloud Computing," *JURIHUM J. Inov. dan Hum.*, vol. 01, no. 01, pp. 148–154, 2023, [Online]. Available: https://jurnalmahasiswa.com/index.php/jurihum

[3] A. Budiyanto, "Apa Itu Cloud Computing? Karakteristik dan Jenis Layanannya," Cloud Computing Indonesia. Accessed: May 04, 2024. [Online]. Available: https://www.cloudcomputing.id/pengetahuan-dasar/apa-itu-cloud-computing

[4] S. Mackey, I. Mihov, A. Nosenko, F. Vega, and Y. Cheng, "A Performance Comparison of WireGuard and OpenVPN," *CODASPY 2020 - Proc. 10th ACM Conf. Data Appl. Secur. Priv.*, no. July, pp. 162–164, 2020, doi: 10.1145/3374664.3379532.

[5] M. Pudelko, P. Emmerich, S. Gallenmüller, and G. Carle, "Performance Analysis of VPN Gateways," *IFIP Netw. 2020 Conf. Work. Netw. 2020*, pp. 325–333, 2020.

[6] H. Redzovic, A. Smiljanic, and B. Savic, "Performance evaluation of Software Routers with VPN features," *24th Telecommun. Forum, TELFOR 2016*, pp. 1–4, 2017, doi: 10.1109/TELFOR.2016.7818727.

[7] J. Brassil and I. Kopaliani, "CloudJoin: Experimenting at scale with Hybrid Cloud Computing," *2020 IEEE 3rd 5G World Forum, 5GWF 2020 - Conf. Proc.*, pp. 467–472, 2020, doi: 10.1109/5GWF49715.2020.9221055.

[8] H. Afifi Al-Atsari and I. Suharjo, "Integrasi Server On-Premise dengan Server Cloud Menggunakan Cloud VPN dan Mikrotik Ipsec Untuk Peningkatan Keamanan Koneksi," *J. Syntax Admiration*, vol. 4, no. 11, pp. 1977–1996, 2023, doi: 10.46799/jsa.v4i11.757.

[9] T. A. Cinderatama, Y. Yunhasnawa, and R. Z. Alhamri, "Desain Dan Implementasi Hybrid Cloud Computing Sebagai Infrastruktur Untuk Analisis Big Data Menggunakan Analytic Hierarchy Process(AHP)," *Techno.Com*, vol. 17, no. 4, pp. 404–414, 2018, doi: 10.33633/tc.v17i4.1871.

[10] P. N. P. Hai, H. N. Hong, B. B. Quoc, and T. Hoang, "A

Comparative Research on VPN Technologies on Operating System for Routers," *Int. Conf. Adv. Technol. Commun.*, vol. 2021-Octob, pp. 89–93, 2021, doi: 10.1109/ATC52653.2021.9598334.

[11] P. Thiruvasagam and K. Jijo George, "IPSec: Performance analysis in IPv4 and IPv6," *J. ICT Stand.*, vol. 7, no. 1, pp. 59–76, 2019, doi: 10.13052/jicts2245-800X.714.

[12] J. L. Shah and J. Parvez, "Impact of IPSec on Real Time applications in IPv6 and 6to4 Tunneled Migration Network," *ICIIECS 2015 - 2015 IEEE Int. Conf. Innov. Information, Embed. Commun. Syst.*, pp. 1–6, 2015, doi: 10.1109/ICIIECS.2015.7193114.

[13] 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," *2022 Int. Symp. Networks, Comput. Commun. ISNCC 2022*, pp. 1–5, 2022, doi: 10.1109/ISNCC55209.2022.9851717.

[14] J. Brassil and I. Kopaliani, "CloudJoin: Experimenting at scale with Hybrid Cloud Computing," *2020 IEEE 3rd 5G World Forum, 5GWF 2020 - Conf. Proc.*, pp. 467–472, 2020, doi: 10.1109/5GWF49715.2020.9221055.

[15] S. Shekhar and I. Researcher, "Comparative Analysis Of Optimizing Hybrid Cloud Environments Using AWS , Azure , And GCP," vol. 10, no. 8, pp. 791–806, 2022.