IMPLEMENTASI WIREGUARD SEBAGAI VPN *SITE-TO-SITE* PADA MIKROTIK ROUTEROS DI LINGKUNGAN LABORATORIUM TEKNOLOGI INFORMASI

TUGAS AKHIR

Diajukan oleh:

RAHMATDI
220705075

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2026 M / 1448 H

LEMBAR PERSETUJUAN

IMPLEMENTASI WIREGUARD SEBAGAI VPN *SITE-TO-SITE* PADA MIKROTIK ROUTEROS DI LINGKUNGAN LABORATORIUM TEKNOLOGI INFORMASI

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
dalam Ilmu Teknologi Informasi

Oleh:
RAHMATDI
NIM. 220705075

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Ghufran Ibnu Yasa, M.T.	**Aulia Syarif Aziz, S.Kom., M.Sc.**
NIP. 198409262014031005	**NIP. 199305212022031001**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Malahayati, M.T.
NIP. 198301272015032003

LEMBAR PENGESAHAN

IMPLEMENTASI WIREGUARD SEBAGAI VPN *SITE-TO-SITE* PADA MIKROTIK ROUTEROS DI LINGKUNGAN LABORATORIUM TEKNOLOGI INFORMASI

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Jum'at, 21 Agustus 2026 M
8 Rabi'ul-Awal 1448 H
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,	Sekretaris,
Ghufran Ibnu Yasa, M.T.	**Aulia Syarif Aziz, S.Kom., M.Sc.**
NIP. 198409262014031005	**NIP. 199305212022031001**
Penguji I,	Penguji II,
Mulkan Fadhli, S.T., M.T.	**Khairan AR, M.Kom.**
NIP. 198811282020121006	**NIP. 198607042014031001**

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rahmatdi

NIM : 220705075

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul : IMPLEMENTASI WIREGUARD SEBAGAI VPN *SITE-TO-SITE* PADA MIKROTIK ROUTEROS DI LINGKUNGAN LABORATORIUM TEKNOLOGI INFORMASI

Dengan ini menyatakan bahwa dalam penulisan tugas akhir/skripsi ini, saya:

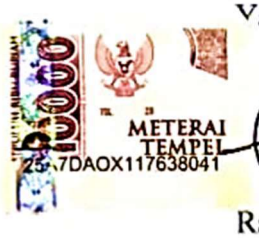
1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 21 Agustus 2026

Yang Menyatakan



Rahmatdi

ABSTRAK

Nama	:	Rahmatdi
NIM	:	220705075
Program Studi	:	Teknlologi Informasi
Judul	:	IMPLEMENTASI WIREGUARD SEBAGAI VPN *SITE-TO-SITE* PADA MIKROTIK ROUTEROS DI LINGKUNGAN LABORATORIUM TEKNOLOGI INFORMASI
Tanggal Sidang	:	21 Agustus 2026
Jumlah Halaman	:	103
Pembimbing I	:	Ghufran Ibnu Yasa, M.T.
Pembimbing II	:	Aulia Syarif Aziz, S.Kom., M.Sc.

Perkembangan teknologi jaringan meningkatkan kebutuhan komunikasi antarjaringan yang aman dan efisien, termasuk pada lingkungan *Small Office/Home Office* (SOHO) dengan perangkat *router* yang memiliki keterbatasan sumber daya komputasi. Penelitian ini mengimplementasikan dan menganalisis *WireGuard* sebagai VPN *site-to-site* serta membandingkan karakteristik performanya dengan L2TP/IPsec, *OpenVPN*, dan SSTP pada *router* MikroTik berarsitektur MIPSBE *single-core*. Metodologi penelitian menggunakan *Network Development Life Cycle* (NDLC) yang disesuaikan dengan tahapan analisis, desain, implementasi, dan monitoring. Evaluasi dilakukan menggunakan pengujian beban TCP dan UDP dengan *iPerf3*, pengukuran *latency* menggunakan ICMP *ping*, serta pemantauan utilisasi CPU dan RAM pada kondisi jaringan ISP asimetris pada siang dan malam hari. Hasil pengujian menunjukkan bahwa karakteristik performa setiap protokol berbeda bergantung pada parameter, arah trafik, dan kondisi waktu pengujian. *WireGuard* menunjukkan *throughput* TCP yang lebih tinggi dibandingkan protokol pembanding pada sebagian besar skenario serta kebutuhan sumber daya yang relatif rendah. Pada kondisi siang, *WireGuard* menghasilkan *throughput* TCP arah *Reverse* sebesar 17,86 Mbps dengan utilisasi CPU sebesar 64,75%, sedangkan L2TP/IPsec mencapai 15,54 Mbps dengan utilisasi CPU 94,85%. Penggunaan RAM *WireGuard* stabil sebesar 37 MB. Pada pengukuran *latency*, median

WireGuard berada pada 5,00–7,00 ms bergantung pada kondisi waktu pengujian. Namun, keunggulan tersebut tidak berlaku pada seluruh parameter. Pada UDP 10 Mbps arah *Reverse* siang hari, *OpenVPN* menghasilkan *packet loss* terendah sebesar 0,093%, dibandingkan L2TP/IPsec sebesar 1,053% dan *WireGuard* sebesar 2,182%, sedangkan pada kondisi malam L2TP/IPsec menghasilkan *packet loss* terendah sebesar 0,037%, diikuti *WireGuard* sebesar 0,056% dan *OpenVPN* sebesar 0,074%. Berdasarkan hasil tersebut, tidak terdapat satu protokol yang unggul secara mutlak pada seluruh parameter pengujian. *WireGuard* menunjukkan karakteristik yang menguntungkan pada *throughput* TCP dan efisiensi penggunaan sumber daya, meskipun terdapat *trade-off* pada *packet loss* UDP dalam kondisi dan arah trafik tertentu. Dengan karakteristik tersebut, *WireGuard* dinilai layak diterapkan sebagai VPN *site-to-site* pada perangkat MikroTik kelas SOHO yang menjadi objek penelitian, khususnya pada kondisi yang memprioritaskan *throughput* TCP dan efisiensi sumber daya perangkat, dengan tetap mempertimbangkan *trade-off* pada keandalan trafik UDP.

Kata Kunci : *WireGuard, L2TP/IPsec, OpenVPN, SSTP, VPN Site-to-Site, QoS, MIPSBE.*

ABSTRACT

Name	:	Rahmatdi
Student Number	:	220705075
Depertement	:	Information Technology
Title	:	IMPLEMENTATION OF WIREGUARD AS A SITE-TO-SITE VPN ON MIKROTIK ROUTEROS IN THE INFORMATION TECHNOLOGY LABORATORY
Date	:	21 August 2026
Number of Pages	:	103
Supervisor I	:	Ghufran Ibnu Yasa, M.T.
Supervisor II	:	Aulia Syarif Aziz, S.Kom., M.Sc.

The development of network technology has increased the need for secure and efficient inter-network communication, including in Small Office/Home Office (SOHO) environments that use routers with limited computational resources. This study implements and analyzes WireGuard as a site-to-site VPN and compares its performance characteristics with L2TP/IPsec, OpenVPN, and SSTP on a MikroTik router with a single-core MIPSBE architecture. The research methodology uses the Network Development Life Cycle (NDLC), consisting of analysis, design, implementation, and monitoring stages. The evaluation was conducted using TCP and UDP load testing with iPerf3, latency measurement using ICMP ping, and monitoring of CPU and RAM utilization under asymmetric ISP network conditions during daytime and nighttime. The results show that the performance characteristics of each protocol vary depending on the parameter, traffic direction, and testing period. WireGuard achieved higher TCP throughput than the comparison protocols in most test scenarios while requiring relatively low system resources. During daytime testing, WireGuard achieved a TCP throughput of 17.86 Mbps in the Reverse direction with CPU utilization of 64.75%, whereas L2TP/IPsec achieved 15.54 Mbps with CPU utilization of 94.85%. WireGuard also maintained a stable RAM usage of 37 MB. In latency measurements, the median latency of WireGuard ranged from 5.00 to 7.00 ms depending on the testing period. However, this

advantage was not consistent across all parameters. In the 10 Mbps UDP Reverse scenario during daytime, OpenVPN achieved the lowest packet loss at 0.093%, compared with 1.053% for L2TP/IPsec and 2.182% for WireGuard, while during nighttime L2TP/IPsec achieved the lowest packet loss at 0.037%, followed by WireGuard at 0.056% and OpenVPN at 0.074%. These results indicate that no single protocol demonstrated absolute superiority across all tested parameters. WireGuard exhibited favorable characteristics in terms of TCP throughput and resource efficiency, although it presented a trade-off in UDP packet loss under certain traffic conditions and directions. Based on these characteristics, WireGuard is considered suitable for deployment as a site-to-site VPN on the MikroTik SOHO-class devices examined in this study, particularly when TCP throughput and resource efficiency are prioritized, while the trade-off in UDP traffic reliability must also be considered.

Keywords : *WireGuard, L2TP/IPsec, OpenVPN, SSTP, Site-to-Site VPN, QoS, MIPSBE.*

KATA PENGANTAR

Puji syukur senantiasa penulis panjatkan ke hadirat Allah SWT, Tuhan Yang Maha Esa, atas segala rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul "Implementasi *WireGuard* sebagai VPN *Site-to-Site* pada MikroTik RouterOS di Lingkungan Laboratorium Teknologi Informasi". Salawat serta salam tak lupa senantiasa tercurahkan kepada junjungan Nabi Besar Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya.

Penyusunan skripsi ini merupakan salah satu syarat akademik untuk menyelesaikan pendidikan program sarjana (S1) pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry Banda Aceh.

Penulis menyadari sepenuhnya bahwa penyelesaian skripsi ini tidak lepas dari dukungan, bimbingan, arahan, serta doa dari berbagai pihak sejak awal masa perkuliahan hingga tahap penyusunan akhir. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi.
2. Bapak Ghufran Ibnu Yasa, M.T. selaku Dosen Pembimbing I dan Bapak Aulia Syarif Aziz, S.Kom., M.Sc. selaku Dosen Pembimbing II, yang telah meluangkan banyak waktu, tenaga, dan pikiran untuk memberikan bimbingan, masukan berharga, serta motivasi yang tak ternilai harganya kepada penulis selama proses penyusunan skripsi ini.
3. Seluruh Bapak/Ibu Dosen serta Staf Akademik Program Studi Teknologi Informasi yang telah membagikan ilmu yang sangat bermanfaat selama masa perkuliahan dan banyak membantu urusan administrasi penulis.
4. Wanita terhebat bergelar ibu, dan pria yang jejak langkahnya takkan pernah bisa kulampaui.
5. Keluarga besar yang tidak pernah putus memberikan doa, kasih sayang, dukungan moral, maupun material sehingga penulis dapat menyelesaikan pendidikan dengan baik.
6. Para manusia lucu di sekitarku baik di dalam maupun di luar lingkungan kampus serta rekan-rekan mahasiswa Teknologi Informasi angkatan 2022

yang telah bersama-sama berjuang, belajar, dan saling memberikan semangat selama masa perkuliahan.

7. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang secara langsung maupun tidak langsung telah memberikan bantuan, dukungan, dan motivasi selama proses penyusunan skripsi ini.

Penulis menyadari bahwa di dalam penulisan dan penyusunan skripsi ini masih terdapat banyak kekurangan karena keterbatasan ilmu dan pengalaman. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari semua pihak.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat yang nyata bagi pengembangan ilmu pengetahuan dan teknologi, khususnya di bidang keamanan dan rekayasa jaringan komputer, serta dapat menjadi referensi yang berguna bagi penelitian-penelitian selanjutnya di masa yang akan datang.

Sehubungan dengan proses penyusunan tugas akhir ini, pemanfaatan kecerdasan buatan (*Artificial Intelligence*/AI) dibatasi sebagai alat bantu dalam eksplorasi informasi, diskusi teknis, penulisan, dan pemeriksaan konsistensi, sedangkan seluruh data, analisis, keputusan, dan hasil penelitian telah diverifikasi serta menjadi tanggung jawab penulis sepenuhnya.

Banda Aceh, 21 Agustus 2026
Penulis

Rahmatdi

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Transformasi digital mendorong organisasi, institusi pendidikan, perusahaan, maupun pengelola sistem berbasis *Internet of Things* (IoT) untuk menghubungkan jaringan yang tersebar secara geografis melalui internet. Penggunaan internet sebagai media komunikasi menimbulkan tantangan keamanan karena data pada jaringan publik berpotensi mengalami penyadapan, modifikasi, maupun akses oleh pihak yang tidak berwenang (Stallings, 2020). Oleh karena itu, diperlukan mekanisme komunikasi yang mampu menjaga keamanan pertukaran data antarjaringan. Salah satu teknologi yang digunakan untuk memenuhi kebutuhan tersebut adalah *Virtual Private Network* (VPN) (Kurose & Ross, 2022; Gleeson dkk., 2000).

Salah satu bentuk implementasi VPN adalah VPN *Site-to-Site*, yaitu mekanisme yang menghubungkan dua atau lebih jaringan lokal melalui *tunnel* terenkripsi sehingga perangkat pada jaringan yang berbeda dapat saling berkomunikasi melalui internet. Implementasi tersebut banyak digunakan untuk menghubungkan kantor pusat dan cabang, laboratorium, maupun infrastruktur IoT yang tersebar secara geografis. Penelitian Santoso dkk. (2021) menunjukkan bahwa VPN *Site-to-Site* berbasis L2TP/IPsec dapat menyediakan komunikasi antarjaringan melalui internet secara aman. Namun, proses enkripsi, dekripsi, dan enkapsulasi paket pada *gateway* menimbulkan *overhead* pemrosesan yang dapat memengaruhi kualitas layanan jaringan (*Quality of Service/QoS*) serta penggunaan sumber daya perangkat (Hussein & Rehman, 2025).

Permasalahan tersebut menjadi lebih relevan pada lingkungan *Small Office/Home Office* (SOHO) yang menggunakan *router* dengan sumber daya komputasi terbatas sebagai *gateway*. Selain menjalankan fungsi *routing* dan layanan jaringan lainnya, *router* harus memproses enkripsi dan dekripsi paket secara *real-time*. Kondisi tersebut semakin kompleks pada jaringan ISP asimetris,

ketika kapasitas *upload* dan *download* berbeda sehingga beban pemrosesan pada arah trafik dapat memiliki karakteristik yang berbeda. Oleh karena itu, evaluasi protokol VPN pada perangkat dengan keterbatasan sumber daya perlu mempertimbangkan tidak hanya kualitas transmisi, tetapi juga efisiensi penggunaan CPU dan memori perangkat.

Berbagai protokol VPN telah dikembangkan dengan karakteristik implementasi yang berbeda, antara lain L2TP/IPsec, *OpenVPN*, SSTP, dan *WireGuard*. *WireGuard* dikembangkan dengan pendekatan desain yang sederhana, basis kode yang relatif ringkas, operasi pada *kernel space*, serta penggunaan algoritma kriptografi modern (Donenfeld, 2017). Karakteristik tersebut menjadikan *WireGuard* relevan untuk dievaluasi pada perangkat jaringan dengan sumber daya komputasi terbatas, khususnya dalam implementasi VPN *Site-to-Site* pada perangkat kelas SOHO.

Sejumlah penelitian telah mengevaluasi *WireGuard* pada berbagai lingkungan implementasi. Rahman dan Harnaningrum (2024) mengevaluasi *WireGuard* pada layanan *Voice over Internet Protocol* (VoIP), Hermawan dan Saputra (2025) mengimplementasikannya pada infrastruktur *hybrid cloud*, sedangkan Muhajir dan Utami (2026) membandingkan *WireGuard* dengan *OpenVPN* pada lingkungan perbankan berbasis MikroTik. Hasil penelitian terdahulu menunjukkan bahwa karakteristik performa *WireGuard* dapat berbeda bergantung pada perangkat keras, topologi, dan lingkungan jaringan yang digunakan. Kondisi tersebut menunjukkan bahwa hasil pengujian pada lingkungan tertentu belum dapat secara langsung merepresentasikan performa *WireGuard* pada perangkat SOHO dengan keterbatasan sumber daya.

Berdasarkan kondisi tersebut, masih terbatas penelitian yang secara khusus menghubungkan parameter QoS dengan penggunaan sumber daya CPU dan RAM pada *router* MikroTik berarsitektur MIPSBE *single-core* tanpa dukungan *hardware cryptographic acceleration*, khususnya dalam jaringan ISP asimetris. Padahal, hubungan antara kualitas transmisi dan beban komputasi merupakan aspek penting untuk menilai karakteristik operasional VPN pada perangkat dengan sumber daya terbatas. Kesenjangan tersebut menjadi dasar dilakukannya

evaluasi komparatif yang tidak hanya mempertimbangkan *throughput*, *latency*, *jitter*, dan *packet loss*, tetapi juga utilisasi CPU dan RAM. Penelitian ini menggunakan lingkungan Laboratorium Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh sebagai lokasi *Headquarters* dan jaringan *Branch* pada lokasi residensial yang dihubungkan melalui internet publik. Lingkungan tersebut dipilih karena merepresentasikan implementasi VPN *Site-to-Site* pada router MikroTik kelas SOHO dengan koneksi ISP asimetris. Pengujian dilakukan menggunakan kapasitas koneksi ISP yang tersedia dengan konfigurasi, topologi, dan skenario yang dikendalikan agar perbandingan antarprotokol dapat dilakukan secara konsisten.

Berdasarkan *research gap* tersebut, penelitian ini mengimplementasikan *WireGuard* sebagai VPN *Site-to-Site* pada dua *router* MikroTik RB951G-2HnD dan RB951Ui-2nD yang berarsitektur MIPSBE *single-core*, kemudian membandingkan karakteristik kinerjanya dengan L2TP/IPsec, *OpenVPN*, dan SSTP menggunakan konfigurasi, topologi, serta skenario pengujian yang identik. Evaluasi dilakukan berdasarkan parameter *throughput*, *latency*, *jitter*, *packet loss*, utilisasi CPU, dan utilisasi RAM pada kondisi jaringan ISP asimetris. Penelitian ini tidak dimaksudkan untuk membuktikan keunggulan mutlak *WireGuard*, tetapi untuk mengidentifikasi karakteristik performa, efisiensi sumber daya, serta *trade-off* masing-masing protokol dalam lingkungan pengujian yang digunakan.

1.2 Rumusan Masalah

1. Bagaimana implementasi dan performa *WireGuard* sebagai VPN *Site-to-Site* pada *router* MikroTik berarsitektur MIPSBE *single-core* dalam kondisi jaringan ISP asimetris dengan keterbatasan *bandwidth*?
2. Sejauh mana efisiensi utilisasi sumber daya CPU dan RAM serta performa parameter Quality of Service (QoS) protokol *WireGuard* dibandingkan dengan protokol VPN pembanding saat diuji pada skenario *load testing* di jaringan ISP asimetris?

1.3 Tujuan Penelitian

1. Menganalisis implementasi dan performa *WireGuard* sebagai VPN *Site-to-Site* secara komparatif dengan protokol VPN pembanding pada router MikroTik berarsitektur MIPSBE *single-core* dalam kondisi jaringan ISP asimetris dengan keterbatasan *bandwidth*.
2. Mengevaluasi efisiensi utilisasi sumber daya CPU dan RAM serta karakteristik parameter Quality of Service (QoS) *WireGuard* dibandingkan dengan protokol VPN pembanding pada skenario *load testing* di jaringan ISP asimetris untuk mengidentifikasi kelebihan dan *trade-off* performa pada kondisi pengujian yang digunakan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi secara teoretis maupun praktis, yang mencakup:

1. **Bagi Institusi Akademis:** Menambah khazanah literatur rekayasa jaringan komputer, khususnya dalam menyediakan rujukan empiris terkait perilaku adaptasi dan efisiensi protokol berbasis *kernel-space* (*WireGuard*) ketika dieksekusi pada topologi dengan kelangkaan sumber daya komputasi (*resource-constrained environment*).
2. **Bagi Praktisi dan Administrator Jaringan:** Menyediakan referensi teknis mengenai penerapan dan kinerja *WireGuard* pada perangkat MikroTik dengan sumber daya terbatas dan jaringan ISP asimetris.

1.5 Batasan Masalah

Agar penelitian lebih terarah, ditetapkan batasan masalah sebagai berikut:

1. Fokus penelitian adalah implementasi *WireGuard*, sedangkan L2TP/IPsec, *OpenVPN* (UDP), dan SSTP digunakan sebagai protokol pembanding.
2. Perangkat perutean dibatasi pada MikroTik berarsitektur MIPSBE *single-core* (RB951G-2HnD dan RB951Ui-2nD hAP) tanpa akselerasi kriptografi perangkat keras.
3. Pengujian dilakukan pada koneksi ISP asimetris sisi *Branch* dengan *bandwidth* 19,35 Mbps (*download*) dan 9,67 Mbps (*upload*).
4. Penyesuaian *Maximum Transmission Unit* (MTU) ke 1420 *bytes* dan *TCP MSS Clamping* diterapkan sebagai prasyarat konfigurasi (SOP) untuk mencegah fragmentasi, bukan sebagai metrik evaluasi independen.
5. Pemantauan sumber daya hanya dilakukan pada *Router Branch* (RB951Ui-2nD) sebagai perangkat berspesifikasi terendah (*worst-case scenario*) untuk menganalisis beban enkripsi dan dekripsi.
6. Evaluasi dibatasi pada uji beban sesaat (*load testing*) dan tidak mencakup pengujian stabilitas jangka panjang seperti *memory leak* atau *thermal throttling*.
7. Evaluasi difokuskan pada aspek rekayasa kinerja (*performance engineering*) dan tidak mencakup pengujian keamanan atau kerentanan algoritma kriptografi.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian mengenai implementasi dan evaluasi kinerja *Virtual Private Network* (VPN), khususnya adopsi protokol modern seperti *WireGuard*, telah banyak dikaji melalui berbagai topologi jaringan. Tinjauan literatur dilakukan guna memetakan posisi teknis, mengidentifikasi kelemahan dari studi terdahulu, dan menegaskan kebaruan (*novelty*) dari penelitian ini.

Komparasi performa antara *WireGuard* dan *OpenVPN* telah dibuktikan oleh Muhajir dan Utami (2026). Meskipun *WireGuard* secara umum memberikan performa yang lebih baik pada stabilitas *delay* dan efisiensi *resource gateway*, penelitian tersebut mengungkap bahwa keunggulan *throughput WireGuard* bersifat kontekstual. Secara spesifik, pada skenario TCP *Upload*, *OpenVPN* justru terbukti secara signifikan mengalahkan *WireGuard* (Median *OpenVPN* 5,38 Mbps vs *WireGuard* 1,50 Mbps). Mengingat adanya anomali performa pada protokol *transport* tertentu ini, penelitian ini akan melakukan pembebanan jaringan menggunakan *iPerf3* pada skenario TCP maupun UDP *Stream* untuk memvalidasi apakah degradasi *WireGuard* pada TCP *Upload* juga terjadi secara konsisten pada arsitektur MIPSBE di lingkungan ISP publik.

Dari segi kemudahan implementasi, studi oleh Ikhwandi, dkk. (2025) menerapkan metode *Software Development Life Cycle* (SDLC) untuk mengintegrasikan *WireGuard* pada *router* MikroTik guna mengamankan akses situs web. Dalam implementasi praktisnya, mereka menemukan bahwa *routing* lintas ISP dapat mendegradasi keunggulan *WireGuard*. Penggunaan *WireGuard* dilaporkan melipatgandakan latensi secara drastis dari 29 ms menjadi 67 ms akibat kompleksitas rute jaringan yang harus melintasi *border* ISP. Karena penelitian ini merancang topologi *Site-to-Site* yang menghubungkan jaringan laboratorium kampus dengan ISP asimetris perumahan, analisis fluktuasi QoS

(khususnya *delay* dan *jitter*) akibat kondisi rute jaringan publik akan menjadi salah satu fokus pengujian utama untuk mengevaluasi ketahanan protokol.

Terkait evaluasi *Quality of Service* (QoS), Rahman dan Harnaningrum (2024) membandingkan ketahanan L2TP/IPsec dan *WireGuard* khusus untuk mengamankan lalu lintas *Voice over IP* (VoIP). Meski *WireGuard* unggul dalam meminimalisir *jitter*, beban uji yang dilakukan tergolong ringan (paket suara), sehingga belum merepresentasikan kerentanan router terhadap kondisi keterbatasan pemrosesan (*CPU-bound bottleneck*) saat mentransmisikan data TCP berskala maksimal. Sementara itu, efisiensi arsitektural *WireGuard* pada ekosistem tervirtualisasi dieksplorasi oleh Hermawan dan Saputra (2025) yang membandingkannya dengan *Classic Cloud VPN* di lingkungan *Hybrid Cloud*. Meskipun mencatat keunggulan *throughput WireGuard* secara keseluruhan, studi tersebut juga menemukan anomali kritis: pada perangkat *Virtual Machine* (VM) berspesifikasi rendah, ketiadaan akselerasi perangkat keras membuat enkripsi ChaCha20-Poly1305 berbasis perangkat lunak pada *WireGuard* menguras utilisasi CPU hingga 95% dan memicu *bottleneck*. Fakta ini mendasari urgensi penelitian ini untuk menguji *WireGuard* pada *router* komoditas MikroTik berarsitektur MIPSBE *single-core* yang juga tidak memiliki dukungan *hardware crypto acceleration*, guna membuktikan apakah limitasi serupa terjadi pada perangkat keras fisik. Untuk memperjelas posisi penelitian ini terhadap studi-studi sebelumnya, komparasi parameter dan kesenjangan riset dirangkum pada Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Penulis & Tahun	**Judul**	**Parameter Pengujian**	**Kesimpulan**
(Muhajir & Utami, 2026)	*Analisis Performa WireGuard dan*	*Throughput*, Latensi	*WireGuard* secara umum unggul pada

	OpenVPN pada VPN Perbankan Berbasis MikroTik menggunakan ICMP, iPerf3, dan Mann-Whitney	(ICMP/*iPerf3*), Analisis Mann-Whitney.	latensi dan efisiensi CPU, namun *OpenVPN* terbukti lebih unggul secara signifikan pada *throughput* skenario TCP Upload.
(Ikhwandi dkk., 2025)	Implementasi *WireGuard* Sebagai Koneksi Menggunakan Routing MikroTik	Kemudahan implementasi, performa tinggi, dan keamanan data menggunakan metode *Software Development Life Cycle* (SDLC).	*WireGuard* meningkatkan *throughput* secara stabil, namun kompleksitas rute lintas ISP melipatgandakan latensi (dari 29 ms menjadi 67 ms). Perencanaan IP *Gateway* yang unik sangat krusial.
(Santoso dkk., 2021)	VPN Site to Site *Implementation* Using Protocol L2TP and IPSec	Stabilitas koneksi, fungsionalitas pertukaran data antar-situs, Throughput, Delay, dan Packet Loss.	L2TP/IPsec sukses mengamankan transmisi data antar-lokasi (*file sharing*).

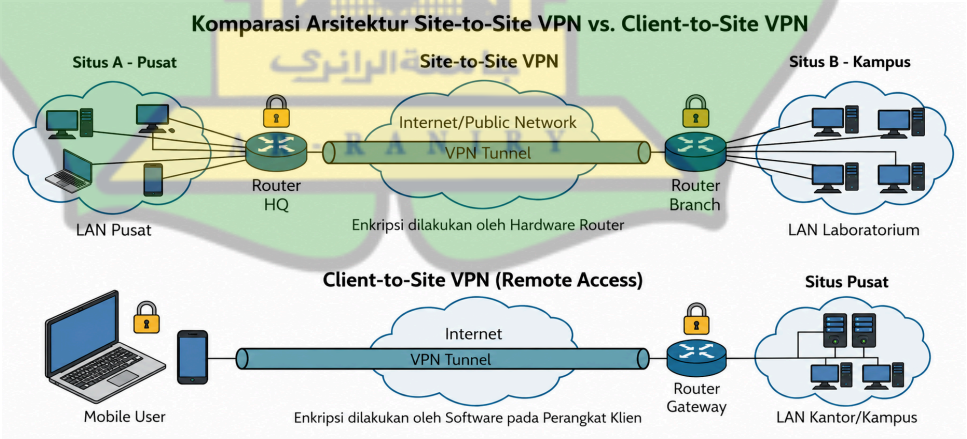
(Rahman & Harnaningrum, 2024)	Analisa Quality of Service (QoS) Pada Jaringan L2TP IPSec Dan *WireGuard* VPN untuk mengamankan VoIP	*Delay, Jitter, Packet Loss* pada transmisi suara. .	Penggunaan L2TP/IPsec dan *WireGuard* sukses mengamankan komunikasi data VoIP. *WireGuard* mencatatkan *delay* yang lebih baik(19.9994ms) dibandingkan L2TP/IPsec (19,9997 ms).
(Hermawan & Saputra, 2025)	Analisis Perbandingan Penggunaan Metode Tunneling Cloud *Virtual Private Network* dan *WireGuard Virtual Private Network* pada Implementasi Infrastruktur Hybrid Cloud	*Throughput*, Latensi, Efisiensi Biaya (*Cost*).	*WireGuard* unggul secara *throughput* dan latensi di lingkungan *cloud*, namun memicu *bottleneck* CPU hingga 95% pada VM berspesifikasi rendah akibat ketiadaan *hardware crypto acceleration*.

Berdasarkan tinjauan pustaka di atas, terdapat kesenjangan riset (*research gap*) di mana sebagian besar penelitian terdahulu mengevaluasi implementasi *WireGuard* pada lingkungan komputasi yang relatif lebih tinggi atau menggunakan skenario pembebanan jaringan yang berbeda dengan kondisi

operasional perangkat SOHO. Penelitian ini mengisi kesenjangan tersebut melalui implementasi *WireGuard* pada topologi VPN *Site-to-Site* menggunakan router MikroTik berarsitektur MIPSBE *single-core* tanpa dukungan hardware crypto acceleration dalam kondisi *bandwidth* ISP asimetris. Selain memberikan kontribusi empiris mengenai performa *WireGuard* pada perangkat berdaya rendah, penelitian ini juga menerapkan pendekatan pengukuran terpusat pada Router *Branch* untuk mengevaluasi utilisasi CPU, RAM, serta karakteristik beban proses enkripsi dan dekripsi pada berbagai skenario pengujian.

2.2 Konsep *Virtual Private Network* (VPN) *Site-to-Site*

Virtual Private Network (VPN) merupakan teknologi yang digunakan untuk mengamankan komunikasi data melalui jaringan publik seperti internet dengan memanfaatkan mekanisme enkripsi, autentikasi, dan *tunnel*ing (Kurose & Ross, 2022). Salah satu implementasi VPN yang umum digunakan adalah topologi *Site-to-Site*, yaitu metode yang menghubungkan dua atau lebih jaringan lokal (LAN) yang berada di lokasi berbeda agar dapat saling berkomunikasi secara transparan. Berbeda dengan model Client-to-Site (Remote Access) yang mendistribusikan proses enkripsi ke perangkat pengguna, topologi *Site-to-Site* memusatkan seluruh proses *tunnel*ing, enkripsi, dan *routing* pada perangkat gateway di masing-masing lokasi (Stallings, 2020).



Gambar 2.1 Komparasi *Site-to-Site* vs Client-to-Site

Pada arsitektur *Site-to-Site*, seluruh proses negosiasi koneksi, enkapsulasi, dekapsulasi, dan forwarding paket dijalankan langsung oleh router VPN (Kurose & Ross, 2022). Kondisi ini menyebabkan beban pemrosesan terpusat pada CPU router, terutama saat lalu lintas jaringan meningkat. Pada perangkat kelas SOHO dengan arsitektur MIPSBE *single-core* tanpa dukungan *hardware crypto acceleration,* kondisi tersebut dapat memicu *bottleneck* CPU yang berdampak pada peningkatan *delay, packet loss,* dan penurunan *throughput*.

2.3 Protokol *WireGuard*

WireGuard merupakan protokol VPN modern yang dirancang dengan struktur implementasi yang lebih ringkas dibandingkan protokol seperti IPsec dan *OpenVPN*. Salah satu karakteristik utama *WireGuard* adalah ukuran basis kode yang relatif kecil sekitar kurang dari 4.000 baris kode, sehingga lebih sederhana dalam proses pengembangan, audit keamanan, dan pemeliharaan sistem. Operasional *WireGuard* berjalan di dalam ruang kernel (*kernel-space*) sehingga secara teoritis dapat mengurangi latensi pemrosesan dan beban CPU karena meniadakan mekanisme perpindahan konteks (*context switching*) antara *kernel-space* dan *user-space* (Donenfeld, 2017).

Pada ekosistem perangkat keras MikroTik, dukungan *native* terhadap *WireGuard* mulai tersedia secara resmi sejak RouterOS versi 7 yang dibangun di atas kernel Linux 5.6. Sebelum kehadiran RouterOS v7, perangkat MikroTik yang menjalankan protokol seperti *OpenVPN* harus memproses data pada *user-space*. Hal ini memicu terjadinya *context-switching* yakni perpindahan proses yang berulang kali antara *user-space* dan *kernel-space* yang memakan penggunaan utilisasi CPU. Dengan terintegrasinya *WireGuard* di *kernel-space* pada RouterOS v7, proses enkripsi, dekripsi, dan perutean paket dapat dieksekusi secara langsung tanpa interupsi *context-switching*. Arsitektur inilah yang membuat *WireGuard* memiliki *overhead* yang secara teoritis berpotensi lebih minimal, sehingga menjadi alternatif yang efisien untuk diimplementasikan pada perangkat router berspesifikasi rendah, termasuk arsitektur MIPSBE *single-core* tanpa dukungan *hardware crypto acceleration*.

2.3.1 Mekanisme *Cryptokey Routing*

Menurut (Donenfeld, 2017), *WireGuard* menggunakan konsep *Cryptokey Routing* yang mengintegrasikan mekanisme kriptografi dengan tabel *routing* jaringan. Dalam arsitektur ini, setiap *peer* diidentifikasi menggunakan *public key* yang dikaitkan dengan daftar alamat IP tertentu melalui parameter *AllowedIPs*. Pendekatan tersebut memungkinkan *WireGuard* membatasi proses dekripsi dan forwarding paket hanya pada *peer* yang memiliki kecocokan alamat IP dan kunci kriptografi yang valid. Selain berfungsi sebagai mekanisme keamanan, *AllowedIPs* juga berperan sebagai tabel rute internal dalam proses pengiriman paket antar-*peer*. Dibandingkan beberapa protokol VPN lain yang memerlukan proses negosiasi dan pengelolaan koneksi yang lebih kompleks, pendekatan *Cryptokey Routing* pada *WireGuard* memiliki struktur *state machine* yang lebih sederhana. Struktur tersebut dirancang dengan mekanisme state yang relatif sederhana sehingga secara teoritis dapat mengurangi kompleksitas pengelolaan koneksi dan *overhead* pemrosesan, terutama pada perangkat dengan keterbatasan sumber daya komputasi.

2.3.2 Algoritma Kriptografi Modern

Efisiensi *WireGuard* pada perangkat keras berdaya rendah didukung oleh penggunaan algoritma kriptografi modern yang dirancang agar dapat berjalan secara optimal pada prosesor umum tanpa dukungan akselerasi khusus. *WireGuard* menggunakan algoritma ChaCha20 untuk enkripsi simetris dan Poly1305 untuk autentikasi pesan (*Message Authentication Code*) (Donenfeld, 2017).

Berbeda dengan algoritma AES (*Advanced Encryption Standard*) yang dapat dijalankan melalui perangkat lunak maupun akselerasi perangkat keras, ChaCha20 dirancang agar dapat berjalan secara efisien menggunakan instruksi CPU umum tanpa bergantung pada akselerator kriptografi khusus (Donenfeld, 2017). Hal ini menjadikan *WireGuard* secara teoritis lebih sesuai untuk perangkat router berarsitektur MIPSBE, seperti MikroTik seri RB951 yang tidak memiliki modul akselerasi kriptografi perangkat keras. Selain menggunakan

ChaCha20-Poly1305, *WireGuard* juga memanfaatkan Curve25519 dalam proses pertukaran kunci menggunakan mekanisme *Elliptic Curve Diffie-Hellman* (ECDH) yang mendukung implementasi *Perfect Forward Secrecy* (PFS) dengan beban komputasi minimal (Donenfeld, 2017; Jumakhan & Mirzaeinia, 2024).

2.3.3 Mekanisme Overhead dan Penyesuaian Maximum Transmission Unit (MTU)

Secara arsitektural, *WireGuard* memiliki beban *overhead* transmisi tetap sebesar 60 *bytes* untuk paket IPv4 dan 80 *bytes* untuk IPv6. *Overhead* ini terdiri dari *header* IPv4 luar (20 *bytes*), *header* UDP (8 *bytes*), dan struktur *framing WireGuard* (32 *bytes*). Sebagaimana dirinci oleh Donenfeld (2017), *header WireGuard* sebesar 32 *bytes* tersebut terdiri dari empat komponen, 4 *bytes* untuk tipe paket, 4 *bytes* untuk indeks kunci, 8 *bytes* untuk nonce, dan 16 *byte* untuk tag autentikasi pesan (*Message Authentication Code*). Overhead tersebut mengharuskan penyesuaian nilai Maximum Transmission Unit (MTU) pada antarmuka virtual. Apabila standar MTU fisik jaringan Ethernet adalah 1500 *bytes*, penambahan *overhead* enkapsulasi *WireGuard* sebesar 60 *bytes* akan menyebabkan ukuran paket bertambah melampaui batas fisik antarmuka. Oleh karena itu, pada penelitian ini penyesuaian nilai MTU pada antarmuka virtual *WireGuard* diturunkan secara manual menjadi 1420 *bytes*. Penyesuaian ini dilakukan untuk mengurangi risiko fragmentasi paket pada lingkungan jaringan IPv4 maupun IPv6 serta mempertahankan efisiensi transmisi.

2.3.4 Lapisan Isolasi Keamanan *Site-to-Site*

Pada arsitektur *Site-to-Site*, isolasi kriptografis *WireGuard* melalui mekanisme *AllowedIPs* beroperasi pada level *peer* router, bukan pada level perangkat individual di dalam LAN. Setiap perangkat yang berada dalam *subnet* yang telah didefinisikan sebagai bagian dari *AllowedIPs* akan secara otomatis dapat melintasi *tunnel* tanpa memerlukan autentikasi *WireGuard* tersendiri, karena proses enkripsi dan dekripsi sepenuhnya ditangani oleh router gateway. Karakteristik ini merupakan prinsip dasar topologi *Site-to-Site* yang memang dirancang agar transparan terhadap perangkat klien. Konsekuensinya, keamanan

akses bergantung sepenuhnya pada kontrol siapa saja yang diizinkan bergabung ke dalam *subnet* tersebut. Apabila terdapat segmen jaringan publik yang tidak disegmentasi secara terpisah dari *subnet* internal yang terdaftar pada *AllowedIPs*, maka perangkat pada segmen tersebut berpotensi memperoleh akses transparan ke jaringan di sisi lain *tunnel*. Risiko ini bersifat universal pada seluruh implementasi VPN *Site-to-Site* dan bukan merupakan kelemahan spesifik protokol *WireGuard*. Mitigasi terhadap risiko ini berada pada ranah perancangan arsitektur jaringan, khususnya segmentasi VLAN antara akses publik dan jaringan internal.

2.4 Protokol VPN Pembanding

Dalam mengevaluasi efisiensi *WireGuard*, penelitian ini menggunakan tiga protokol VPN sebagai pembanding pengujian yaitu L2TP/IPsec, *OpenVPN* dan SSTP yang merupakan pendahulu *WireGuard* pada perangkat router MikroTik. Ketiga protokol ini mewakili pendekatan kriptografi dan enkapsulasi yang berbeda, sehingga dapat digunakan sebagai pembanding dalam evaluasi utilisasi CPU dan performa jaringan pada perangkat router berspesifikasi rendah.

Penelitian ini tidak bertujuan untuk menentukan performa maksimum masing-masing protokol VPN pada berbagai kelas perangkat keras, melainkan untuk mengevaluasi kelayakan implementasi *WireGuard* pada perangkat MikroTik berarsitektur MIPSBE *single-core* dengan menggunakan protokol VPN yang sudah dijelaskan diatas sebagai protokol pembanding. Oleh karena itu, seluruh interpretasi hasil penelitian dibatasi pada karakteristik perangkat dan kondisi operasional yang digunakan selama eksperimen, dan tidak dimaksudkan sebagai generalisasi lintas kelas perangkat keras VPN secara umum.

2.4.1 L2TP/IPsec (Layer 2 Tunneling Protocol over IPsec)

L2TP/IPsec merupakan protokol VPN yang menggabungkan mekanisme *tunnel*ing L2TP dengan sistem keamanan IPsec. Implementasi ini menghasilkan proses enkapsulasi ganda (*double encapsulation*) di mana paket data dibungkus oleh *header* L2TP dan kemudian diamankan menggunakan *header* IPsec ESP (*Encapsulating Security Payload*) (Stallings, 2020). Pada perangkat MikroTik

berarsitektur MIPSBE yang tidak memiliki modul akselerasi kriptografi perangkat keras (*hardware crypto acceleration*), seluruh proses kalkulasi algoritma enkripsi IPsec (seperti AES) dieksekusi melalui perangkat lunak. Hal ini menyebabkan meningkatnya penggunaan utilisasi CPU dan menambah latensi ketika router menangani lalu lintas jaringan dalam jumlah besar (Demirdelen & Kirmizi, 2025).

2.4.2 *OpenVPN*

OpenVPN merupakan protokol VPN yang arsitekturnya beroperasi pada *user-space*. Implementasi *OpenVPN* yang berjalan pada *user-space* melibatkan interaksi antara proses VPN dan *network stack* kernel sehingga dapat menimbulkan *overhead* perpindahan konteks (Høiland-Jørgensen dkk., 2018). Di sisi lain, sistem juga harus menangani interupsi perangkat keras (*hardware interrupt* atau IRQ) dari antarmuka jaringan, yang turut meningkatkan beban pemrosesan pada *network stack* (Cai & Karsten, 2023). Pada perangkat dengan prosesor *single-core*, akumulasi kedua jenis *overhead* tersebut meningkatkan penggunaan CPU sehingga *throughput* yang dapat dicapai menjadi lebih rendah.

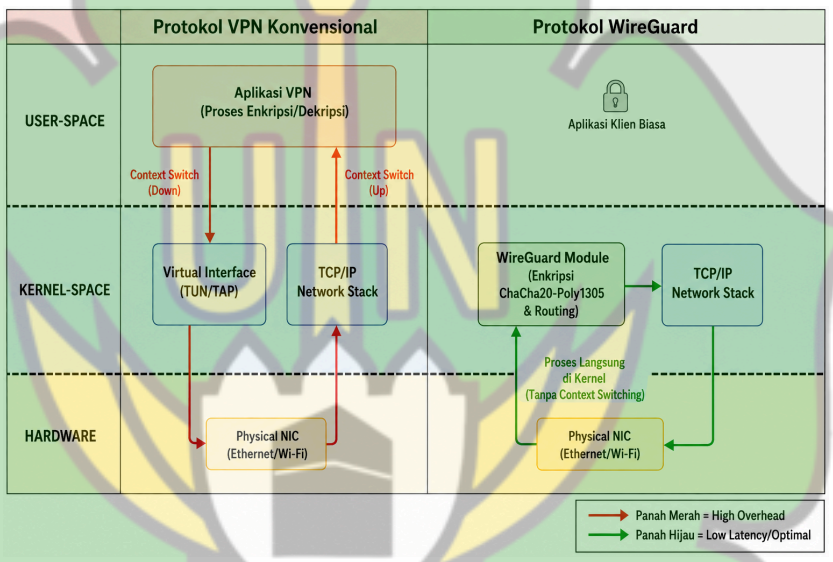
2.4.3 SSTP (*Secure Socket Tunneling Protocol*)

SSTP mengenkapsulasi paket melalui saluran SSL/TLS yang berjalan di atas protokol *transport* TCP. Penggunaan TCP sebagai media pembungkus rentan memicu fenomena *TCP Meltdown*. Sebagaimana dijelaskan secara fundamental oleh (Honda dkk., 2005) mengenai arsitektur *tunneling TCP-over-TCP*, saat terjadi kongesti atau gangguan pada jaringan publik, mekanisme kontrol kemacetan (*congestion control*) pada protokol TCP di dalam *tunnel* akan bertabrakan dengan kontrol kemacetan pada TCP pembungkus. Kondisi ini memicu retransmisi paket dan menambah latensi, terutama pada jaringan dengan kapasitas *bandwidth* terbatas atau kualitas koneksi yang tidak stabil.

2.5 MikroTik RouterOS sebagai Platform VPN

MikroTik RouterOS merupakan sistem operasi jaringan berbasis Linux yang digunakan untuk menjalankan fungsi *routing*, firewall, manajemen *bandwidth*, dan berbagai layanan jaringan lainnya. Dalam ekosistem jaringan

komoditas, RouterOS banyak diadopsi oleh institusi pendidikan dan perkantoran skala kecil hingga menengah (SOHO) karena rasio efisiensi biaya dan kelengkapan fiturnya. Seiring dengan rilisnya arsitektur RouterOS versi 7, MikroTik secara resmi mengintegrasikan dukungan *native* untuk protokol *WireGuard* langsung ke dalam *kernel-space* sistem operasi (MikroTik, t.t..). Integrasi ini memungkinkan proses enkripsi, dekripsi, dan forwarding paket dijalankan langsung oleh kernel tanpa bergantung pada aplikasi VPN tambahan di *user-space*. Visualisasi perbandingan arsitektur pemrosesan vpn pada *user-space* dan *kernel-space* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Perbandingan Arsitektur Pemrosesan VPN pada *User-space* dan *Kernel-space*

Pada protokol VPN lama yang berjalan di *user-space*, setiap paket data yang masuk harus melewati proses *context switching* yang berulang antara *user-space* dan *kernel-space*. Kondisi tersebut dapat menambah *overhead* pemrosesan dan penggunaan CPU, terutama pada perangkat dengan arsitektur *single-core*. Dengan implementasi *native* pada RouterOS v7, *WireGuard* memiliki karakteristik pemrosesan yang lebih ringan dibandingkan beberapa protokol VPN sebelumnya yang umum digunakan pada MikroTik. Karakteristik ini menjadi salah satu alasan pemilihan RouterOS sebagai platform penelitian, khususnya pada perangkat MIPSBE dengan keterbatasan sumber daya komputasi (Demirdelen & Kirmizi, 2025).

2.6 Arsitektur Perutean Berdaya Rendah (MIPSBE)

Kinerja pemrosesan protokol kriptografi sangat bergantung pada kapabilitas arsitektur perangkat keras. Penelitian ini menggunakan perangkat router kelas SOHO dengan arsitektur *Microprocessor without Interlocked Pipelined Stages Big Endian* (MIPSBE) inti tunggal (*single-core*). Arsitektur MIPS pada dasarnya adalah desain prosesor *Reduced Instruction Set Computer* (RISC) yang dirancang untuk efisiensi komputasi umum, namun memiliki keterbatasan dalam menangani beban kriptografi secara intensif (Patterson & Hennessy, 2021).

Salah satu karakteristik perangkat MikroTik seri RB951 adalah ketiadaan modul perangkat keras akselerasi kriptografi (*hardware crypto offloading*), seperti instruksi AES-NI yang umum ditemukan pada prosesor kelas *enterprise* atau peladen (x86_64) (MikroTik, 2017). Akibatnya, proses enkripsi dan dekripsi dijalankan langsung oleh CPU tanpa bantuan akselerator kriptografi khusus. Pada perangkat dengan arsitektur *single-core*, CPU harus menangani proses *routing*, interupsi paket jaringan, dan kalkulasi kriptografi secara bersamaan. Kondisi tersebut dapat meningkatkan beban *Software Interrupt* (*SoftIRQ*) dan mempengaruhi performa router ketika lalu lintas jaringan meningkat. Dalam beberapa kondisi, bottleneck CPU dapat terjadi meskipun *throughput* jaringan belum mencapai kapasitas maksimum *bandwidth* ISP. Karena keterbatasan tersebut, penggunaan protokol dengan *overhead* pemrosesan yang lebih ringan menjadi penting untuk dievaluasi, khususnya pada perangkat router dengan sumber daya komputasi terbatas seperti MIPSBE.

2.7 Fragmentasi Paket: MTU dan *TCP MSS Clamping*

Setiap protokol *tunneling* VPN menambahkan *header* tambahan (enkapsulasi) pada paket data asli, yang secara otomatis mengurangi ruang efektif untuk muatan data (*payload*). Jika standar *Maximum Transmission Unit* (MTU) fisik jaringan Ethernet adalah 1500 *bytes*, penambahan *overhead WireGuard* sebesar 60 *bytes* akan menyebabkan ukuran total paket bertambah melampaui batas fisik antarmuka. Sebagai kompensasi, router akan terpaksa memecah paket

tersebut (fragmentasi) di lapisan IP. Proses reasembli (penyusunan ulang) paket yang terfragmentasi ini sangat membebani utilisasi CPU dan dapat menutupi data performa enkripsi yang sebenarnya saat pengujian berlangsung.

Secara teoretis, protokol jaringan menggunakan mekanisme *Path MTU Discovery* (PMTUD) untuk menegosiasikan ukuran paket secara dinamis menggunakan pesan ICMP *Fragmentation Needed*. Namun, pada jaringan publik internet, administrator keamanan seringkali memblokir lalu lintas ICMP, sehingga menciptakan fenomena kebuntuan yang dikenal sebagai *PMTUD Blackhole* (Deering & Mogul, 1990).

Untuk memitigasi anomali tersebut tanpa membebani pemrosesan pada router, diterapkan teknik intersepsi TCP *Maximum Segment Size* (MSS) *Clamping* (Borman, 2012). Algoritma *firewall* ini akan memproses paket TCP SYN pada fase *handshake* dan secara dinamis memangkas nilai MSS ke batas yang aman (menyesuaikan dengan sisa MTU setelah dikurangi *overhead* VPN), sehingga memastikan lalu lintas data mengalir tanpa mengalami fragmentasi. Dalam penelitian ini, konfigurasi MTU 1420 *bytes* dan *TCP MSS Clamping* diterapkan secara seragam pada seluruh protokol sebagai Standar Operasional Prosedur (SOP) prasyarat pengujian untuk memastikan bahwa seluruh degradasi kinerja (utilisasi CPU, RAM dan *throughput*) yang terekam pada instrumen lebih merepresentasikan beban pemrosesan kriptografi, bukan akibat malfungsi fragmentasi paket.

2.8 Parameter Kinerja Jaringan (Quality of Service)

Pembuktian empiris terhadap efisiensi dan keandalan protokol *tunneling* diukur menggunakan parameter *Quality of Service* (QoS). QoS merupakan mekanisme standar yang merepresentasikan kinerja kolektif dari sebuah layanan jaringan dalam menjamin kualitas transmisi data (International Telecommunication Union, 2023). Dalam penelitian ini, evaluasi QoS difokuskan pada empat metrik fundamental yang diukur menggunakan instrumen Pembebanan Jaringan TCP/UDP, meliputi:

1. ***Throughput*:** Rasio laju transfer data efektif yang berhasil ditransmisikan melintasi medium jaringan dan diterima oleh simpul tujuan pada interval waktu spesifik, umumnya direpresentasikan dalam satuan *Megabits per second* (Mbps).
2. ***Delay (Latency)*:** Akumulasi waktu tempuh yang dibutuhkan oleh sebuah paket data untuk melakukan perjalanan dari titik sumber (*initiator*) hingga mencapai peladen tujuan. Dalam arsitektur VPN, metrik ini sangat dipengaruhi oleh *processing delay* (waktu yang dihabiskan Router untuk melakukan kalkulasi kriptografi enkripsi dan dekripsi).
3. ***Jitter*:** Variasi penundaan (*delay variation*) atau fluktuasi waktu kedatangan antar-paket data yang berurutan. *Jitter* yang tinggi mengindikasikan ketidakstabilan rute atau beban pemrosesan pada router yang fluktuatif.
4. ***Packet Loss*:** Packet loss merupakan persentase paket yang gagal mencapai tujuan. Pada jaringan dengan keterbatasan kapasitas dan sumber daya pemrosesan, *packet loss* dapat dipengaruhi oleh beberapa faktor, antara lain kongesti jaringan, keterbatasan kapasitas buffer, maupun keterbatasan kemampuan perangkat dalam memproses trafik yang masuk. Oleh karena itu, nilai *packet loss* dalam penelitian ini digunakan sebagai indikator kualitas transmisi pada berbagai kondisi beban pengujian.

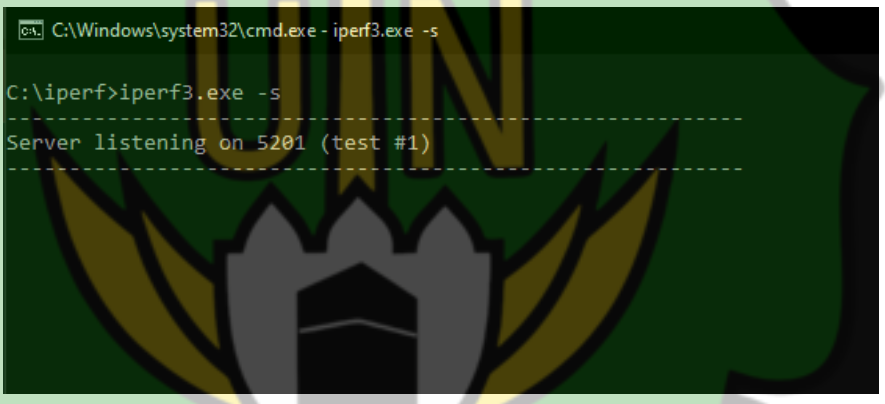
Dalam penelitian ini, hasil pengujian dirangkum menggunakan statistik deskriptif, dengan nilai rata-rata (*mean*) sebagai ukuran utama untuk menggambarkan kinerja masing-masing protokol VPN. Khusus pada parameter *latency*, analisis dilengkapi dengan uji Mann-Whitney U untuk membandingkan distribusi nilai *latency* antara kondisi siang dan malam pada masing-masing protokol. Uji dilakukan pada taraf signifikansi $\alpha = 0{,}05$, dengan nilai $p < 0{,}05$ menunjukkan adanya perbedaan yang signifikan secara statistik.

2.9 Parameter dan Alat Uji Jaringan (*iPerf3*)

Pengujian beban jaringan (*load testing*) pada penelitian ini menggunakan utilitas *iPerf3*. *iPerf3* merupakan perangkat lunak *open-source* yang dirancang untuk melakukan pengukuran aktif terhadap *throughput* maksimum yang dapat

dicapai pada jaringan berbasis IP (Dugan dkk., 2025). Berbeda dengan utilitas pengujian berbasis protokol aplikasi seperti *FTP*, *HTTP*, maupun *SMB* yang rentan terhadap *overhead* pembacaan diska (*disk I/O*), *iPerf3* bekerja dengan menghasilkan dan menerima aliran data langsung dari memori akses acak (RAM). Karakteristik ini membuat hasil pengukuran lebih merepresentasikan kapasitas *bandwidth* jaringan dan latensi pemrosesan protokol tanpa dipengaruhi keterbatasan perangkat keras penyimpanan terminal.

Secara arsitektural, *iPerf3* bekerja menggunakan model *client-server*. Satu titik akhir jaringan harus diinisiasi sebagai peladen pendengar (*server/listener*), sementara titik yang lain bertindak sebagai klien yang mengirimkan aliran paket data (*traffic generator*), sebagaimana diilustrasikan pada gambar 2.3 berikut:



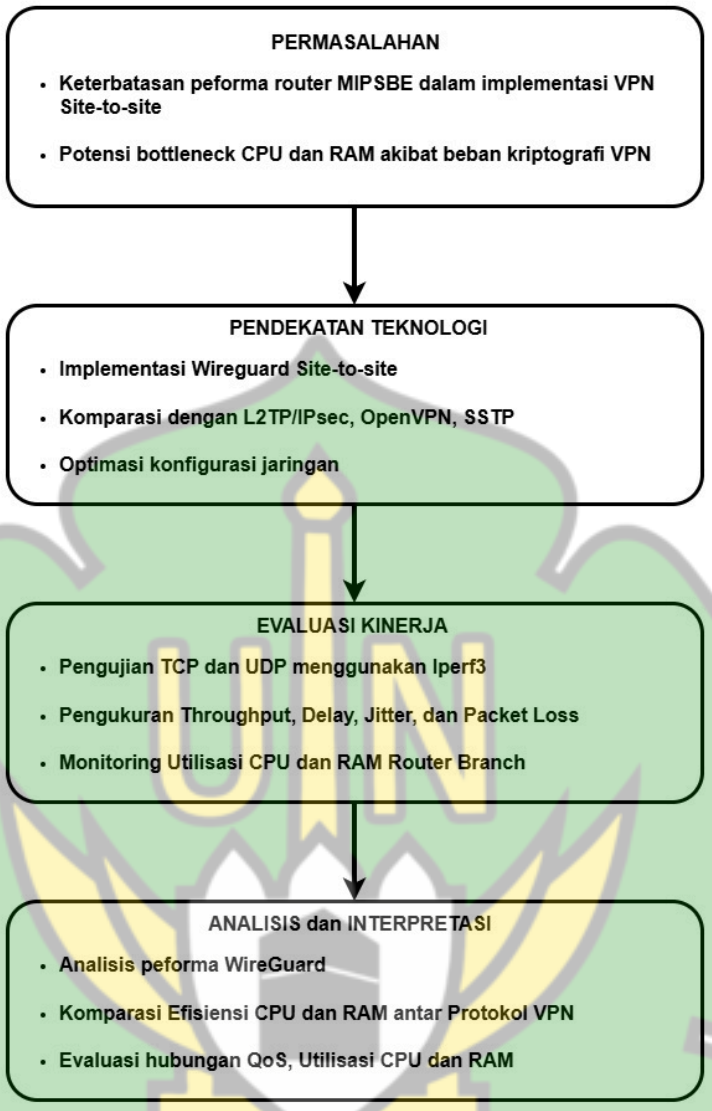
Gambar 2.3 *iPerf3* Server *Listener*

Selain mengukur angka *throughput* maksimum pada protokol TCP, *iPerf3* memiliki kapabilitas untuk melakukan pengujian berbasis datagram UDP secara presisi. Pengujian menggunakan UDP pada *iPerf3* memungkinkan peneliti untuk merekam secara langsung laporan parameter QoS komprehensif, termasuk fluktuasi waktu kedatangan paket (*jitter*) dan persentase paket yang gagal mencapai tujuan (*packet loss*), yang merupakan indikator utama dari efisiensi pemrosesan protokol VPN pada saat jaringan mengalami kongesti (Muhajir & Utami, 2026).

2.10 Kerangka Berpikir

Kerangka berpikir dalam penelitian ini disusun untuk menganalisis permasalahan performa transmisi jaringan antarcabang (*Site-to-Site*) pada perangkat router komoditas berspesifikasi rendah. Permasalahan yang diidentifikasi berkaitan dengan keterbatasan sumber daya komputasi pada router MIPSBE *single-core* ketika menangani proses perutean dan pemrosesan trafik VPN. Kondisi tersebut dapat memengaruhi karakteristik performa jaringan dan utilisasi sumber daya perangkat, sehingga diperlukan evaluasi komparatif untuk melihat perbedaan karakteristik masing-masing protokol VPN pada lingkungan pengujian yang sama. Dalam penelitian ini, *WireGuard* digunakan sebagai protokol utama yang dibandingkan dengan L2TP/IPsec, *OpenVPN*, dan SSTP.

Tahap selanjutnya adalah evaluasi komparatif (*benchmarking*) melalui skenario pembebanan (*load testing*) pada kapasitas jaringan ISP yang digunakan. Pemantauan difokuskan pada Router *Branch* untuk mengevaluasi utilisasi sumber daya pada kondisi lalu lintas asimetris antara arah *Normal* dan *Reverse*. Seluruh protokol dianalisis menggunakan parameter QoS berupa *throughput*, *latency*, *jitter*, dan *packet loss*, serta parameter utilisasi CPU dan RAM. Hasil pengujian kemudian dibandingkan untuk mengidentifikasi karakteristik performa dan efisiensi sumber daya serta trade-off yang muncul pada setiap protokol. Secara visual, alur kerangka berpikir penelitian ini diilustrasikan pada Gambar 2.4 berikut:



Gambar 2.4 Kerangka Berpikir Penelitian

BAB III
METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada dua lokasi yang saling terhubung melalui jaringan internet publik dengan arsitektur VPN *Site-to-Site*. Adapun lokasi dan waktu pelaksanaan penelitian dijelaskan sebagai berikut:

3.1.1 Tempat Penelitian

Penelitian dan pengambilan data performa jaringan dilakukan pada dua lokasi fisik yang saling terhubung melalui jaringan internet publik, yaitu:

1. Sisi *Headquarters* (HQ)

   Bertempat di Laboratorium Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh. Lokasi ini berfungsi sebagai jaringan tujuan (*Headquarters*) pada implementasi VPN *Site-to-Site*.
2. Sisi *Branch*

   Bertempat di lingkungan residensial (rumah) di Banda Aceh yang menggunakan layanan internet broadband asimetris. Lokasi ini berperan sebagai *Branch* sekaligus inisiator *tunnel* VPN untuk mendukung kebutuhan pengujian pada jaringan publik di luar jam operasional laboratorium kampus.

3.1.2 Waktu Penelitian

Penelitian dilaksanakan mulai bulan Mei 2026 hingga Juli 2026. Proses pengambilan data QoS dan utilisasi CPU dilakukan pada dua rentang waktu pengujian untuk mengakomodasi variasi kepadatan trafik pengguna ISP (*contention ratio*), yaitu:

1. Pengujian Siang Hari (13.00 – 17.00 WIB)

   Dilakukan untuk merepresentasikan kondisi jaringan pada periode operasional kampus.

2. Pengujian Malam Hari (20.00 – 00.00 WIB)

   Dilakukan untuk merepresentasikan kondisi jaringan pada periode dengan aktivitas jaringan residensial yang lebih tinggi.

Adapun jadwal pelaksanaan tahapan penelitian secara rinci disajikan pada Tabel 3.1.

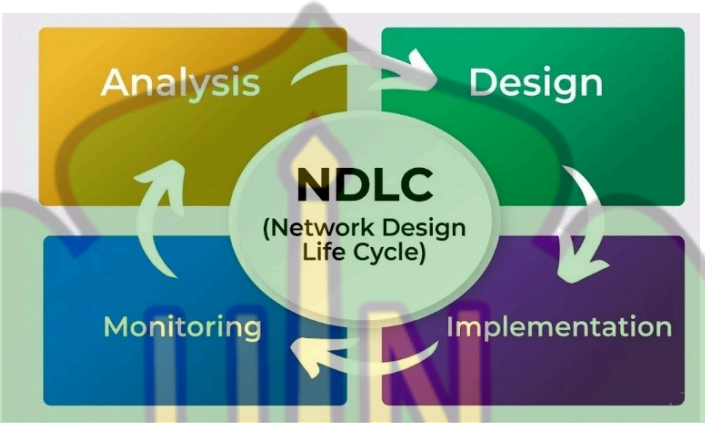
Tabel 3.1 Tahapan Kegiatan Penelitian

No	Tahapan Kegiatan	Mei 2026	Juni 2026	Juli 2026
1	Studi Literatur dan Identifikasi Masalah			
2	Analisis Kebutuhan Jaringan dan Alat Uji			
3	Perancangan Topologi Jaringan *Site-to-Site*			
4	Konfigurasi Perangkat Keras Fisik (HQ & *Branch*)			
5	Pengujian dan Pengambilan Data QoS serta Utilisasi CPU			
6	Analisis Data dan Komparasi Hasil Protokol			

3.2 Tahapan Penelitian

Penelitian ini menggunakan metode *Network Development Life Cycle* (NDLC) sebagai landasan dalam proses perancangan dan implementasi jaringan. Metode NDLC dipilih karena menyediakan tahapan yang sistematis dalam pengembangan infrastruktur jaringan, mulai dari analisis kebutuhan hingga implementasi dan pemantauan sistem (Luthfi & Sinduningrum, 2025). Model NDLC yang dijelaskan oleh Luthfi dan Sinduningrum (2025) terdiri atas lima tahapan, yaitu *Analysis, Design, Simulation Prototype, Implementation, dan*

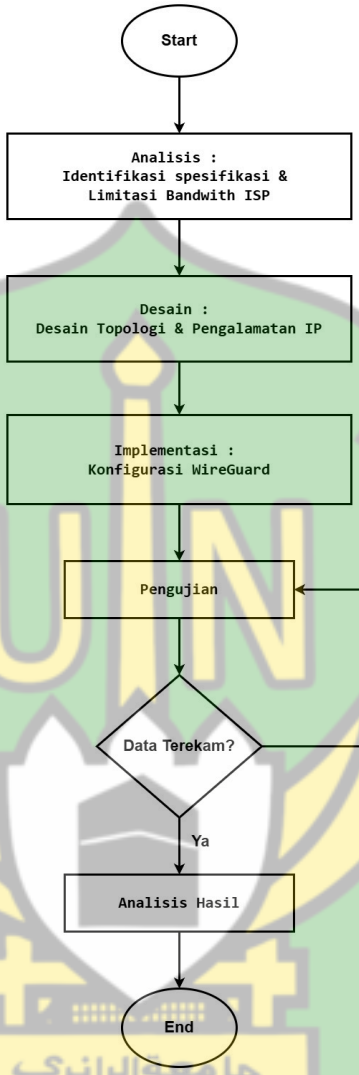
Monitoring. Namun, pada penelitian ini dilakukan penyesuaian dengan tidak menerapkan tahap *Simulation Prototype* karena seluruh konfigurasi dan pengujian dilakukan secara langsung pada perangkat MikroTik fisik tanpa menggunakan lingkungan simulasi jaringan. Dengan demikian, tahapan NDLC yang diterapkan pada penelitian ini terdiri atas *Analysis*, *Design*, *Implementation*, dan *Monitoring*, sebagaimana ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram NDLC

1. **Analisis (*Analysis*):** Tahap ini dilakukan untuk mengidentifikasi keterbatasan pemrosesan pada router MIPSBE *single-core* serta menganalisis kapasitas *bandwidth* aktual dari koneksi ISP yang digunakan pada lokasi pengujian .
2. **Desain (*Design*):** Tahap desain mencakup perancangan topologi jaringan *Site-to-Site* antara jaringan Laboratorium Kampus (*Headquarters*) dan Rumah (*Branch*), termasuk skema pengalamatan IP (*IP Addressing*) .
3. **Implementasi (*Implementation*):** Tahap implementasi dilakukan dengan mengonfigurasi *tunnel WireGuard* pada MikroTik RouterOS, termasuk penerapan konfigurasi MTU dan aturan *TCP MSS Clamping* untuk mencegah fragmentasi paket selama transmisi.
4. **Pemantauan (*Monitoring*):** Tahap ini dilakukan dengan menghasilkan lalu lintas jaringan menggunakan metode *load testing* untuk mengukur parameter *Quality of Service* (QoS) dan utilisasi *resources* router.

Adapun alur kerja tahapan penelitian ini digambarkan melalui *flowchart* pada Gambar 3.2.

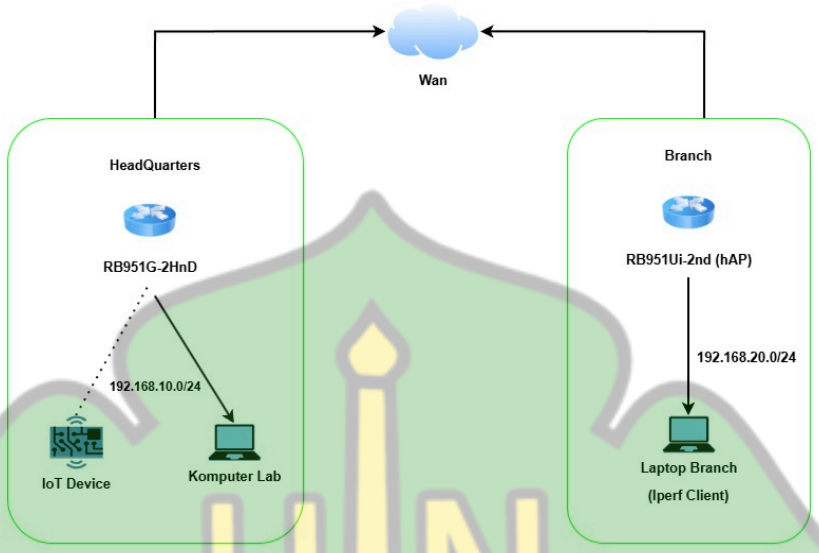


Gambar 3.2 Flowchart Tahapan Penelitian Berdasarkan NDLC

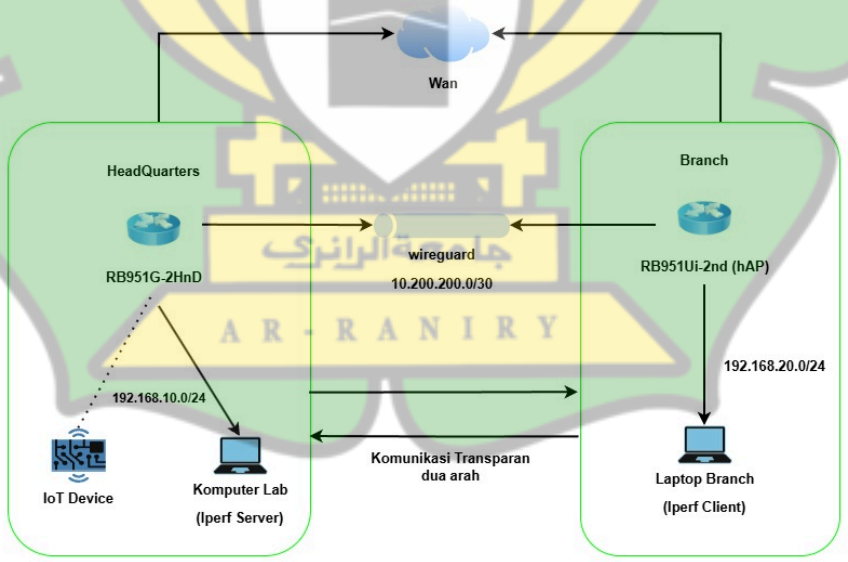
3.3 Topologi Jaringan

Pengujian dilakukan menggunakan infrastruktur jaringan fisik yang menghubungkan jaringan rumah dengan laboratorium kampus melalui koneksi internet publik. Arsitektur yang dirancang menggunakan model *Site-to-Site* secara penuh, sehingga perangkat klien di kedua sisi dapat saling berkomunikasi secara transparan tanpa perlu menginstal aplikasi VPN tambahan.

Alur trafik lalu lintas data pada topologi penelitian ini dapat dilihat pada Gambar 3.3 dan Gambar 3.4:



Gambar 3.3 Kondisi 1 Topologi Jaringan Penelitian *Site-to-Site* sebelum VPN terbentuk



Gambar 3.4 Kondisi 2 Topologi Jaringan Penelitian *Site-to-Site* sesudah *tunnel* aktif.

Sebelum *tunnel WireGuard* terbentuk, kedua jaringan privat tidak dapat saling berkomunikasi secara langsung karena masing-masing berada di belakang NAT router dengan IP publik dinamis dari ISP (Kondisi 1). Setelah konfigurasi *WireGuard* diterapkan dengan parameter *AllowedIPs* yang menjembatani kedua *subnet* melalui *tunnel* logis 10.200.200.0/30, kedua LAN dapat saling berkomunikasi secara transparan (Kondisi 2).

3.4 Spesifikasi Perangkat dan Instrumen Pengujian

Untuk memastikan eksperimen merepresentasikan kondisi jaringan dengan keterbatasan sumber daya komputasi (*resource-constrained environment*), seluruh perangkat keras, infrastruktur jaringan, dan instrumen perangkat lunak yang digunakan dalam penelitian ini dirangkum pada Tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat, Jaringan, dan Instrumen Pengujian

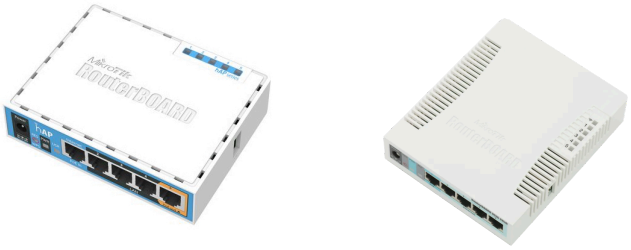
Kategori	**Nama Alat/Sistem**	**Spesifikasi Detail**	**Peran dalam Pengujian**
Perangkat Keras (Router)	MikroTik RB951G-2HnD	Arsitektur CPU MIPSBE *single-core*, tanpa dukungan *hardware crypto acceleration*.	Sebagai *Router Headquarters* (Kampus).
Perangkat Keras (Router)	MikroTik hAP RB951Ui-2nD	Arsitektur CPU MIPSBE *single-core*, tanpa dukungan *hardware crypto acceleration*.	Sebagai *Router Branch* (Rumah) sekaligus titik pemantauan utilisasi CPU.

Perangkat Keras (Klien)	*Personal Computer* (PC)	Spesifikasi standar operasional laboratorium kampus.	Bertindak sebagai Server *iPerf3* (titik penerima lalu lintas data) di sisi Kampus.
Perangkat Keras (Klien)	Laptop	Spesifikasi standar komputasi pengguna.	Bertindak sebagai Klien *iPerf3* (*traffic generator*) di jaringan Rumah.
Kapasitas Jaringan (ISP)	Tautan Internet Asimetris	Kapasitas Unduh (*Downstream*): 19,35 Mbps, Kapasitas Unggah (*Upstream*): 9,67 Mbps.	Berfungsi sebagai batas kapasitas *bandwidth* pengujian.
Instrumen Lunak (QoS)	*iPerf3*	Utilitas pengukuran jaringan berbasis datagram.	Alat utama untuk menghasilkan lalu lintas pengujian dan mengukur *Throughput*, *Jitter*, serta *Packet Loss*.
Instrumen Lunak (QoS)	ICMP Ping	Utilitas perintah baris bawaan (*Command Prompt*).	Mengukur *delay* (*latency*) jaringan.

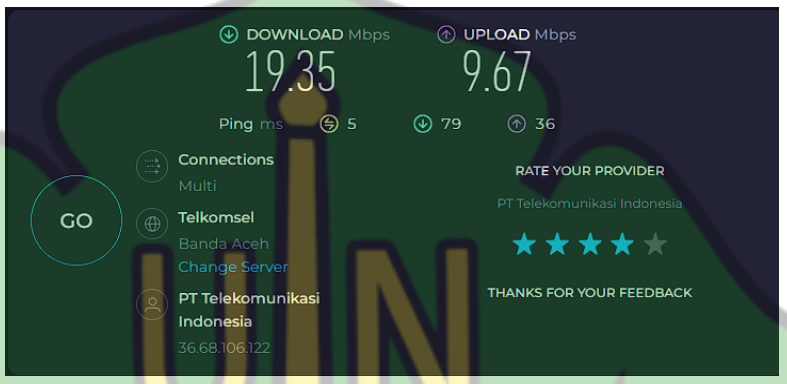
Instrumen Lunak (Sistem)	MikroTik System Resource	Utilisasi CPU, *free-memory* dan *total-memory* real-time	Merekam utilisasi CPU dan RAM selama Pengujian.
Instrumen Lunak (Validasi)	RouterOS/tool/speed-test	Pengujian *throughput*, *ping*, *jitter*, CPU antar router	Validasi silang terbatas terhadap hasil *iPerf3*

Pemantauan performa jaringan pada penelitian ini dilakukan menggunakan tiga instrumen perangkat lunak utama. Pertama, *iPerf3* digunakan sebagai pembangkit dan pengukur lalu lintas jaringan (*traffic generator*) untuk memperoleh *throughput* pada skenario TCP, serta *throughput*, *jitter*, dan *packet loss* pada skenario UDP. Kedua, utilitas ICMP Ping digunakan untuk mengukur parameter *latency* (*delay*) antar-*host* melalui pengiriman paket *Internet Control Message Protocol* (ICMP) *Echo Request*. Ketiga, fitur *MikroTik System Resource* pada RouterOS digunakan untuk memantau utilisasi CPU dan RAM *router* selama pengujian berlangsung, dengan penggunaan RAM dihitung berdasarkan parameter *free-memory* dan *total-memory*. Ketiga instrumen tersebut digunakan pada tahap pengujian sesuai dengan parameter yang diukur sehingga data QoS dan utilisasi sumber daya dapat direkam untuk setiap skenario pengujian.

Adapun dokumentasi fisik dari perangkat router MikroTik yang digunakan dalam penelitian ini diperlihatkan pada Gambar 3.5. Selain itu, kapasitas *bandwidth* jaringan (*bandwidth*) dari penyedia layanan internet di sisi *Branch* yang menjadi acuan beban pengujian ditunjukkan pada Gambar 3.6.



Gambar 3.5 RouterBOARD MikroTik hAP RB951Ui-2nD dan RB951G-2HnD



Gambar 3.6 Kapasitas Jaringan (ISP) *Branch*

3.4.1 Isolasi Lingkungan Pengujian

Untuk memastikan validitas hasil pengujian tidak terkontaminasi oleh *traffic* dari pihak luar, lingkungan pengujian dirancang dengan beberapa lapis isolasi. Pada level kriptografi, *WireGuard* menerapkan mekanisme *Cryptokey Routing* yang hanya mengizinkan *peer* dengan kombinasi *public key* dan *private key* yang valid untuk membentuk *tunnel*. Sebagai bagian dari mekanisme tersebut, parameter *AllowedIPs* secara internal membatasi perutean *enkripsi* dan *dekripsi* paket hanya pada *subnet* 192.168.10.0/24 (HQ) dan 192.168.20.0/24 (*Branch*). Pada level fisik, antarmuka *wireless* Router *Branch* dinonaktifkan selama pengujian berlangsung sehingga koneksi ke jaringan *Branch* hanya dapat dilakukan melalui kabel LAN. Langkah ini memastikan bahwa tidak ada perangkat lain yang dapat terhubung ke jaringan *Branch* selama sesi pengujian, sehingga pengaruh lalu lintas dari perangkat lokal dapat diminimalkan selama proses pengujian.

3.5 Implementasi dan Optimasi Jaringan

Langkah-langkah konfigurasi mencakup pembentukan jalur VPN dan kalibrasi antarmuka untuk mencegah anomali transmisi:

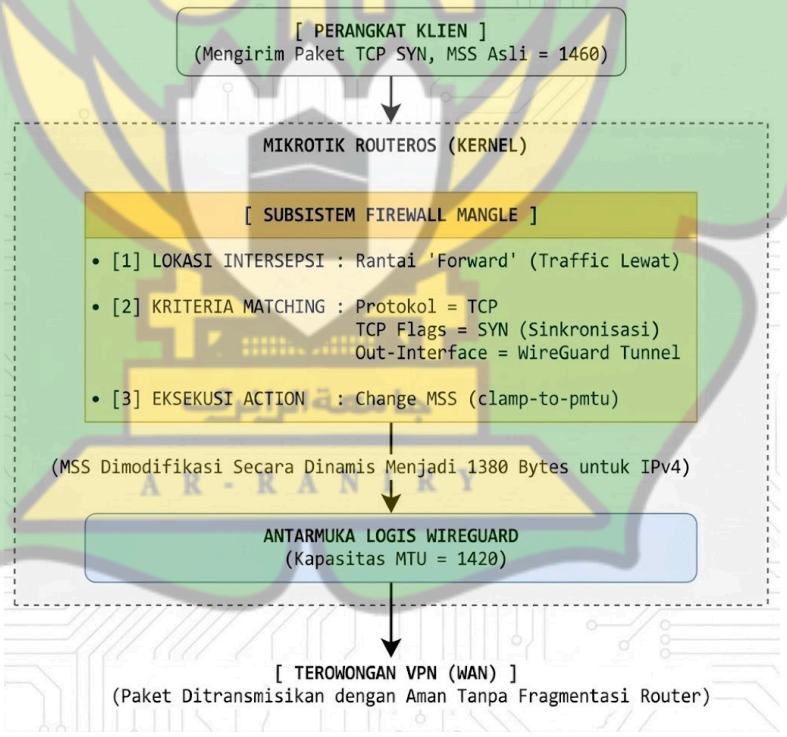
1. **Konfigurasi Routing Kriptografi:**
   Inisiasi antarmuka *WireGuard* pada *Router HQ* dan *Router Branch*, pembentukan pasangan kunci asimetris (*Public/Private Key*), serta konfigurasi parameter *AllowedIPs* guna menjembatani lalu lintas subjaringan 192.168.10.0/24 dan 192.168.20.0/24 melalui *tunnel* logis 10.200.200.0/30.
2. **Penyesuaian MTU (*Maximum Transmission Unit*):**
   MTU antarmuka *WireGuard* dikonfigurasi sebesar 1420 *byte* untuk menyediakan ruang terhadap *overhead* yang ditambahkan selama proses enkapsulasi sehingga risiko fragmentasi paket dapat diminimalkan. Pada jaringan Ethernet standar, ukuran maksimum paket IP yang dapat ditransmisikan melalui media fisik umumnya adalah 1500 *byte* (Ethernet MTU). Ketika sebuah paket IP dikirim melalui *tunnel WireGuard*, paket tersebut dienkapsulasi ke dalam paket UDP dengan menambahkan outer IPv4 *header* sebesar 20 *bytes* dan UDP *header* sebesar 8 *bytes*, sehingga menghasilkan *overhead* transport sebesar 28 *bytes*. Selanjutnya, *header WireGuard* menambahkan 32 *bytes* yang terdiri dari 4 *bytes* tipe paket, 4 *bytes* indeks kunci, 8 *bytes* nonce, dan 16 *bytes* tag autentikasi Poly1305, sebagaimana telah dijelaskan pada Subbab 2.3.3. Dengan demikian, total *overhead* enkapsulasi yang digunakan sebagai dasar penyesuaian MTU dalam penelitian ini adalah sebesar 60 *bytes* untuk transport data message. Oleh karena itu, MTU antarmuka *WireGuard* dikonfigurasi lebih kecil agar ukuran paket setelah proses enkapsulasi tetap berada di bawah batas Ethernet MTU.

   Apabila MTU antarmuka tidak disesuaikan, paket yang dikirim berpotensi melebihi batas MTU Ethernet setelah proses enkapsulasi sehingga dapat menyebabkan fragmentasi pada lapisan IP. Pada

perangkat dengan sumber daya komputasi terbatas, proses fragmentasi dan *reassembly* paket dapat menambah proses pemrosesan paket sehingga berpotensi meningkatkan beban CPU dan menurunkan efisiensi transmisi. Oleh karena itu, konfigurasi MTU yang sesuai merupakan salah satu langkah penting dalam menjaga stabilitas komunikasi pada implementasi *WireGuard*.

3. **Penerapan *TCP MSS Clamping*:**

   Untuk memastikan paket TCP dari klien menyesuaikan ukuran MTU *WireGuard* yang telah direduksi, diterapkan aturan *firewall mangle* pada MikroTik RouterOS. Paket inisiasi TCP (SYN) pada *forward chain* diproses menggunakan instruksi "*action=change-mss new-mss=clamp-to-pmtu"* untuk menyesuaikan nilai *Maximum Segment Size* (MSS) secara dinamis berdasarkan nilai *Path MTU* (PMTU).



Gambar 3.7 Alur Logika Intersepsi *TCP MSS Clamping* pada Rantai *Forward Firewall Mangle* MikroTik

Keberhasilan optimasi transmisi pada *tunnel* VPN bergantung pada penempatan aturan MSS Clam*ping* di dalam alur pemrosesan paket RouterOS. Gambar 3.7 mengilustrasikan alur logika pemrosesan paket (*packet flow*) saat mekanisme *TCP MSS Clamping* diaktifkan pada MikroTik RouterOS. Pada topologi *Site-to-Site*, router berfungsi sebagai *transit gateway*, sehingga aturan *mangle* ditempatkan pada *forward chain*, bukan pada *input chain* maupun *output chain*.

Ketika klien menginisiasi koneksi TCP menggunakan paket SYN, RouterOS akan memproses paket tersebut sebelum dienkapsulasi ke dalam *tunnel WireGuard*. Sistem kemudian menghitung nilai PMTU dan menyesuaikan nilai MSS pada *header* TCP secara dinamis. Pada kondisi MTU Ethernet 1500 *bytes*, nilai MSS IPv4 tanpa *overhead tunnel* umumnya sebesar 1460 *bytes*. Dengan MTU antarmuka *WireGuard* sebesar 1420 *bytes*, nilai MSS yang sesuai untuk trafik TCP IPv4 menjadi 1380 *bytes*. Penyesuaian MSS ini memungkinkan klien dan server menggunakan ukuran segmen TCP yang sesuai sejak proses *three-way handshake*, sehingga dapat mengurangi fragmentasi paket dan beban pemrosesan tambahan pada router MIPSBE selama transmisi berlangsung.

4. **Pemantauan Utilisasi Memori (RAM):**

   Selain pemantauan CPU, dilakukan pula pemantauan utilisasi RAM pada Router *Branch* menggunakan parameter *free-memory* dan *total-memory* dari fitur System Resource RouterOS. RAM yang digunakan dihitung menggunakan persamaan *RAM_used = total_memory - free_memory*, dan dilaporkan dalam satuan megabyte (MB) serta persentase. Pemantauan ini dilakukan karena perangkat RB951Ui-2nD hanya memiliki RAM 64MB, sehingga tekanan memori berpotensi menjadi faktor pembatas tersembunyi yang mempengaruhi performa VPN, khususnya saat CPU mendekati kapasitas maksimum.

3.6 Skenario dan Parameter Pengujian

Untuk menjaga kesetaraan kondisi pengujian dan mengisolasi pengaruh protokol *transport* terhadap hasil pengukuran, *OpenVPN* dikonfigurasi menggunakan UDP. Konfigurasi ini dipilih agar *OpenVPN* menggunakan protokol *transport* yang sama dengan *WireGuard*, yang secara eksklusif beroperasi di atas UDP. Dengan demikian, pengaruh fenomena *TCP-over-TCP* yang diketahui dapat menyebabkan degradasi kinerja dapat dihindari, sehingga hasil pengujian lebih merepresentasikan perbedaan karakteristik implementasi kedua VPN daripada perbedaan mekanisme *transport* (Anyam dkk., 2025; Honda dkk., 2005).

Setiap skenario pengujian dilakukan secara berulang (replikasi) sebanyak dua puluh kali untuk memperoleh nilai rata-rata (*mean*). Skenario pembebanan jaringan (*traffic generation*) mencakup:

1. **Single TCP Stream:** Mengukur *throughput* pada satu sesi koneksi TCP.
2. **Multi TCP Stream (Parameter -P 4):** Digunakan untuk mengevaluasi kemampuan protokol VPN dalam memanfaatkan kapasitas *bandwidth* yang tersedia melalui beberapa aliran TCP paralel.
3. **UDP 5 Mbps:** Evaluasi *delay* dan *jitter* pada laju normal di bawah kapasitas *bandwidth* ISP.
4. **UDP 10 Mbps (*Overload*):** Evaluasi *delay* dan *jitter* yang sengaja melampaui batas fisik *uplink* ISP (9,6 Mbps) untuk mensimulasikan kondisi kongesti jaringan, *queueing delay*, *tail drop*, dan *packet loss*
5. Ping ICMP: Mengukur *latency* .

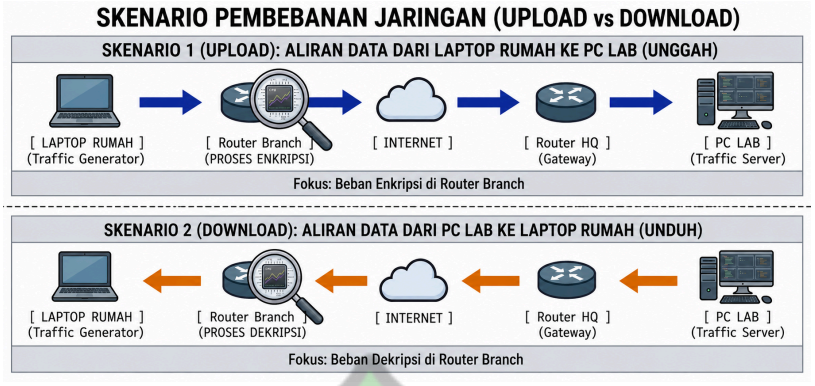
Seluruh skenario pengujian dilakukan pada dua rentang waktu berbeda sebagaimana dijelaskan pada Subbab 3.1.2 untuk mengakomodasi kemungkinan variasi kondisi trafik jaringan ISP antara jam operasional kampus dan jam penggunaan residensial. Pemilihan setiap skenario pengujian dilakukan untuk merepresentasikan karakteristik lalu lintas jaringan yang berbeda. Pengujian TCP *Single Stream* digunakan untuk mengevaluasi performa VPN pada satu aliran

koneksi TCP, yang umum dijumpai pada aktivitas seperti transfer berkas dan akses layanan berbasis web. Pengujian TCP *Multi Stream* digunakan untuk mengevaluasi kemampuan protokol VPN dalam memanfaatkan *bandwidth* melalui beberapa koneksi TCP yang berjalan secara bersamaan. Sementara itu, pengujian UDP 5 Mbps dan UDP 10 Mbps digunakan untuk mengevaluasi performa VPN terhadap lalu lintas real-time yang sensitif terhadap *latency*, *jitter*, dan *packet loss*. Nilai 10 Mbps dipilih karena mendekati kapasitas *upload* efektif koneksi ISP yang digunakan, sedangkan dipilih 5 Mbps sebagai representasi beban lalu lintas UDP pada tingkat sedang sehingga dapat dibandingkan dengan skenario 10 Mbps yang mendekati kapasitas efektif uplink ISP.

Selama pengujian berlangsung, PC pada Laboratorium Kampus (HQ) dijalankan secara persisten sebagai *listener/server* menggunakan *iperf3 -s*. Seluruh proses *traffic generation* dilakukan dari perangkat klien pada sisi Rumah (*Branch*). Pengujian dilakukan menggunakan dua arah aliran lalu lintas sebagai berikut:

1. **Jalur *Upload* (*Branch* ke HQ/*Normal*)**
   Dilakukan menggunakan perintah standar *iperf3 -c [IP_Tunnel]* untuk mengukur beban pemrosesan enkripsi pada Router *Branch*.
2. **Jalur *Download* (HQ ke *Branch*/*Reverse*)**
   Dilakukan menggunakan parameter inversi *iperf3 -c [IP_Tunnel] -R*, sehingga lalu lintas dikirim dari HQ menuju *Branch* untuk mengukur beban pemrosesan dekripsi pada Router *Branch*.

Evaluasi beban komputasi kriptografi dilakukan menggunakan satu titik pemantauan agar perbandingan utilisasi CPU dan RAM antar-skenario dapat dilakukan secara konsisten. Gambar 3.8 merinci dua skenario pembebanan jaringan yang digunakan dalam pengujian ini.



Gambar 3.8 Mekanisme Pembebanan Jaringan Upload dan Download pada Titik Pantau Tunggal

Pada Skenario 1 (*Upload*), Router *Branch* berperan sebagai pengirim *tunnel* yang melakukan proses enkapsulasi dan enkripsi data sebelum paket dikirim melalui jaringan publik. Utilisasi CPU dan RAM yang terekam pada kondisi ini digunakan untuk merepresentasikan beban pemrosesan enkripsi. Sebaliknya, pada Skenario 2 (*Download*), Router *Branch* berperan sebagai penerima *tunnel* yang melakukan proses dekapsulasi dan dekripsi data. Utilisasi CPU dan RAM pada kondisi ini digunakan untuk mengevaluasi beban pemrosesan dekripsi. Penggunaan titik pemantauan yang sama pada kedua skenario bertujuan untuk mengurangi pengaruh perbedaan spesifikasi perangkat keras terhadap hasil pengujian, sehingga perbandingan utilisasi CPU dan RAM antara proses enkripsi dan dekripsi dapat dilakukan secara lebih konsisten.

Beban pengujian pada penelitian ini disesuaikan dengan kapasitas efektif koneksi ISP asimetris yang digunakan pada sisi *Branch*, yaitu sebesar 19,35 Mbps untuk *download* dan 9,67 Mbps untuk *upload*. Penentuan beban tersebut bertujuan agar pengujian merepresentasikan kondisi operasional yang umum dijumpai pada implementasi VPN *Site-to-Site* di lingkungan *Small Office/Home Office* (SOHO). Oleh karena itu, penelitian ini tidak menggunakan beban lalu lintas yang dirancang untuk memaksa utilisasi CPU mencapai titik jenuh (CPU saturation point), melainkan berfokus pada evaluasi kelayakan implementasi *WireGuard* dalam kondisi penggunaan aktual sesuai dengan keterbatasan *bandwidth* yang tersedia. Dengan pendekatan tersebut, hasil pengujian diharapkan dapat memberikan gambaran mengenai performa *WireGuard* pada kondisi operasional

nyata, bukan pada kapasitas maksimum teoritis perangkat. Pemetaan parameter dan instrumen pengujian ditunjukkan pada Tabel 3.3.

Tabel 3.3 Pemetaan Parameter dan Instrumen Pengujian

Parameter Kinerja	**Metode / Protokol Uji**	**Instrumen Uji (Tools)**	**Tujuan Pengukuran**
Throughput	TCP *Stream*	*iPerf3*	Kapasitas maksimal transmisi data murni.
Latency / Delay	ICMP *Echo Request*	*ping*	Mengukur waktu tempuh paket jaringan.
Jitter	UDP *Stream*	*iPerf3*	Fluktuasi kedatangan paket jaringan.
Packet Loss	UDP *Stream*	*iPerf3*	Persentase paket yang gagal mencapai tujuan.
Utilisasi CPU dan RAM	*System Monitoring*	/resource CPU Load & RAM	Beban siklus pemrosesan router saat pengujian.

3.6.1 Validasi Router-to-Router

Sebagai validasi tambahan terhadap hasil *iPerf3*, dilakukan pengujian *throughput* langsung antar router menggunakan fitur bawaan RouterOS /tool/speed-test melalui IP *tunnel WireGuard*. Dokumentasi resmi RouterOS menyatakan bahwa pengujian *bandwidth* sebaiknya dilakukan melalui perangkat yang diuji (through the device), bukan langsung pada perangkat tersebut sebagai

endpoint (on the device), guna menghindari pembebanan CPU akibat proses pembangkitan *traffic* test yang dapat mendistorsi hasil pengukuran (MikroTik, t.t.). Mengingat topologi penelitian ini menempatkan Router HQ dan Router *Branch* sekaligus sebagai endpoint VPN yang diukur dan endpoint speed-test, hasil pengujian ini berpotensi mengandung *overhead* tambahan. Oleh karena itu, pengujian *bandwidth* menggunakan fitur bawaan ini diposisikan murni sebagai data validasi pelengkap silang, bukan pengganti metodologi utama *iPerf3*. Untuk menjaga prinsip konsistensi variabel kontrol dalam eksperimen kuantitatif, validasi ini dilakukan secara setara pada seluruh protokol yang diuji (*WireGuard*, L2TP/IPsec, *OpenVPN*, dan SSTP) serta dieksekusi secara merata pada kedua rentang waktu pengamatan (siang dan malam hari).

3.7 Validitas Penelitian

Penelitian ini menggunakan koneksi internet publik sehingga kualitas jaringan dipengaruhi oleh kondisi operasional ISP yang dapat berubah akibat mekanisme *shared bandwidth*, *routing*, maupun *contention ratio*. Untuk meminimalkan pengaruh tersebut, seluruh pengujian dilakukan menggunakan skenario yang sama pada dua rentang waktu pengamatan, yaitu siang dan malam, dengan konfigurasi perangkat, topologi, parameter VPN, serta kapasitas *bandwidth* yang dijaga tetap konstan. Selain itu, lingkungan pengujian di sisi *Branch* diisolasi melalui beberapa lapis pengamanan sebagaimana telah dijelaskan pada Subbab 3.4.1, sehingga lalu lintas yang terekam selama pengujian didominasi oleh trafik pengujian itu sendiri. Meskipun demikian, hasil penelitian ini merepresentasikan karakteristik performa protokol pada lingkungan pengujian yang menggunakan koneksi ISP publik dan perangkat MikroTik MIPSBE *single-core* sebagaimana dikonfigurasi dalam penelitian. Hasil tersebut tidak dimaksudkan untuk merepresentasikan kapasitas maksimum teoritis seluruh perangkat MikroTik berarsitektur MIPSBE maupun kondisi jaringan pada lingkungan yang berbeda.

3.8 Metode Analisis Data

Data hasil eksperimen diperoleh melalui pengujian yang dilakukan secara berulang (*replikasi*) sebanyak 20 kali pada setiap skenario. Pendekatan replikasi digunakan untuk memperoleh data yang lebih representatif serta mengurangi pengaruh fluktuasi jaringan selama proses pengujian. Penerapan pengujian secara berulang juga sejalan dengan penelitian Muhajir dan Utami (2026) yang menerapkan replikasi pengukuran dalam analisis performa VPN. Selanjutnya, data dianalisis melalui tahapan berikut.

1. **Rekapitulasi Data**

   Hasil pengujian direkapitulasi berdasarkan parameter *throughput*, *latency*, *jitter*, *packet loss*, utilisasi CPU, dan utilisasi RAM untuk masing-masing protokol VPN, arah trafik, dan kondisi waktu pengujian. Data dari setiap skenario dipertahankan dalam bentuk hasil pengukuran individual sebelum dihitung statistik ringkasannya.

2. **Analisis Statistik Deskriptif**

   Statistik deskriptif digunakan untuk menggambarkan kecenderungan hasil pengukuran pada setiap parameter. Nilai mean digunakan untuk merepresentasikan kecenderungan rata-rata pada parameter *throughput*, *jitter*, *packet loss*, utilisasi CPU, dan utilisasi RAM. Sementara itu, parameter *latency* direpresentasikan menggunakan median selain mean karena data *latency* pada jaringan publik lebih rentan terhadap fluktuasi dan pencilan (*outlier*) atau *latency spike*. Nilai minimum dan maksimum digunakan sebagai informasi tambahan untuk melihat rentang hasil pengukuran, tetapi tidak digunakan sebagai metrik utama dalam matriks *trade-off*.

3. **Analisis Statistik Inferensial**

   Uji Mann–Whitney U digunakan untuk menganalisis apakah terdapat perbedaan distribusi *latency* yang signifikan antara kondisi siang dan malam pada masing-masing protokol VPN. Pemilihan uji nonparametrik ini didasarkan pada karakteristik data *latency* yang dapat mengandung pencilan dan tidak selalu mengikuti distribusi

simetris, sehingga pendekatan berbasis peringkat lebih sesuai dibandingkan uji parametrik yang bergantung pada asumsi distribusi tertentu. Uji dilakukan pada taraf signifikansi $\alpha = 0{,}05$ dengan tingkat kepercayaan 95%. Hipotesis yang digunakan adalah H_0: tidak terdapat perbedaan distribusi *latency* antara kondisi siang dan malam, sedangkan H_1: terdapat perbedaan distribusi *latency* antara kedua kondisi tersebut. H_0 ditolak apabila p-value < 0,05.

4. **Analisis Komparatif dan Trade-off**

   Hasil statistik deskriptif dibandingkan antarprotokol berdasarkan kondisi waktu dan arah trafik. Perbandingan tidak digunakan untuk menetapkan satu protokol sebagai pemenang mutlak, tetapi untuk mengidentifikasi perbedaan karakteristik performa dan efisiensi penggunaan sumber daya. Hubungan antara *throughput*, kualitas transmisi, utilisasi CPU, dan penggunaan RAM digunakan sebagai dasar untuk mengidentifikasi trade-off masing-masing protokol.

5. **Visualisasi dan Interpretasi**

   Hasil pengujian disajikan dalam bentuk tabel dan visualisasi untuk memperlihatkan perbedaan performa antarprotokol. Interpretasi dilakukan dengan mempertimbangkan kondisi waktu, arah trafik, kapasitas jaringan ISP, serta keterbatasan perangkat yang digunakan dalam penelitian.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi *WireGuard*

Implementasi *WireGuard* sebagai VPN *Site-to-Site* dilakukan sesuai rancangan yang telah dijelaskan pada Subbab 3.5, meliputi konfigurasi *routing* kriptografi, penyesuaian *Maximum Transmission Unit* (MTU), dan penerapan *TCP MSS Clamping*. Konfigurasi diterapkan pada kedua *router* penelitian, yaitu Router HQ (MikroTik RB951G-2HnD) sebagai representasi jaringan Laboratorium Kampus, dan Router *Branch* (MikroTik RB951Ui-2nD) sebagai representasi jaringan rumah. Kedua *router* dihubungkan melalui *tunnel* logis dengan *subnet* 10.200.200.0/30, menjembatani jaringan lokal 192.168.10.0/24 (HQ) dan 192.168.20.0/24 (*Branch*).

Perbandingan kondisi jaringan sebelum dan sesudah implementasi ditunjukkan pada Gambar 3.3 dan 3.4. Sebelum *tunnel* terbentuk, kedua jaringan lokal tidak memiliki jalur komunikasi langsung karena terpisah oleh NAT dan internet publik. Setelah konfigurasi diterapkan, kedua LAN dapat saling berkomunikasi secara transparan tanpa memerlukan instalasi perangkat lunak VPN tambahan di sisi klien. Hasil konfigurasi *peer WireGuard* pada masing-masing *router* ditunjukkan pada Gambar 4.1 untuk Router HQ dan Gambar 4.2 untuk Router *Branch*.

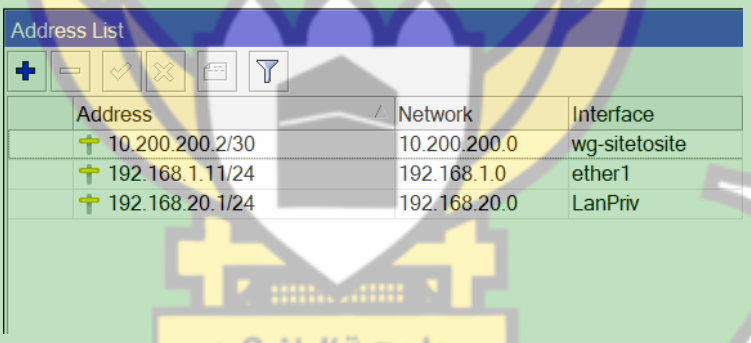
```
[admin@Router-A-HQ] > /interface wireguard peers print detail
Flags: X - disabled
 0   ;;; Peer-to-Router-A
     interface=wg-sitetosite
     public-key="[redacted]"
     endpoint-address=de7a0d09f29d.sn.mynetname.net endpoint-port=13231
     current-endpoint-address=36.68.111.20 current-endpoint-port=13231
     allowed-address=10.200.200.2/32,192.168.20.0/24 persistent-keepalive=25s
     rx=1815.0KiB tx=12.7MiB last-handshake=1m17s
[admin@Router-A-HQ] > /interface wireguard print detail
Flags: X - disabled; R - running
 0  R name="wg-sitetosite" mtu=1420 listen-port=13231
      private-key="[redacted]"
      public-key="[redacted]"
[admin@Router-A-HQ] >
```

Gambar 4.1 Konfigurasi *Peer WireGuard* pada Router HQ

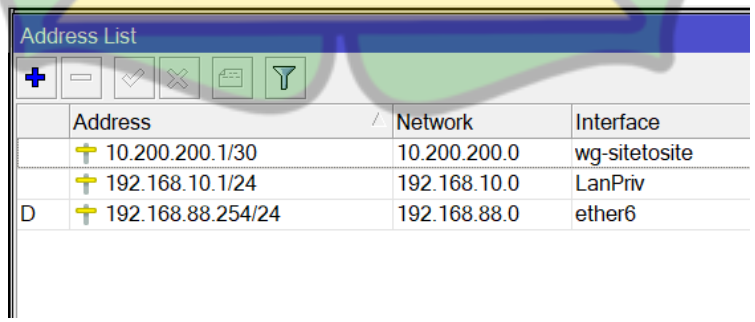
```
[admin@Router-B-Branch] > /interface wireguard peers print detail
Flags: X - disabled; D - dynamic
 0    ;;; Peer-to-Router-HQ
      interface=wg-sitetosite name="peer1"
      public-key="[redacted]"
      private-key="" endpoint-address=965009959267.sn.mynetname.net
      endpoint-port=13231 current-endpoint-address=103.107.187.9
      current-endpoint-port=13231
      allowed-address=10.200.200.1/32,192.168.10.0/24
      preshared-key="" persistent-keepalive=25s client-endpoint=""
      rx=17.0MiB tx=3886.5KiB last-handshake=1m25s
[admin@Router-B-Branch] > /interface wireguard print detail
Flags: X - disabled; R - running
 0  R name="wg-sitetosite" mtu=1420 listen-port=13231
      private-key="[redacted]"
      public-key="[redacted]"
[admin@Router-B-Branch] >
```

Gambar 4.2 Konfigurasi *Peer WireGuard* pada Router *Branch*

Nilai *last-handshake* pada Gambar 4.2 dan 4.3 yang tercatat baru saja terjadi membuktikan bahwa *tunnel* secara aktif melakukan pertukaran kunci sesuai mekanisme *rekeying* periodik *WireGuard*. Selanjutnya, skema pengalamatan IP yang diterapkan untuk menjembatani kedua jaringan ditunjukkan pada Gambar 4.3.



Gambar 4.3 Skema Pengalamatan IP *Tunnel* Router *Branch*



Gambar 4.4 Skema Pengalamatan IP *Tunnel* pada HQ

Untuk memberikan gambaran menyeluruh secara teknis, ringkasan parameter konfigurasi kunci dari kedua *router* disajikan pada Tabel 4.1.

Tabel 4.1 Ringkasan Parameter Konfigurasi *WireGuard Site-to-Site*

Parameter Konfigurasi	Router HQ	Router *Branch*
Antarmuka (*Interface*)	*wg-sitetosite*	*wg-sitetosite*
Listen Port	13231	13231
MTU	1420 *byte*	1420 *byte*
Alamat IP *Tunnel*	10.200.200.1/30	10.200.200.2/30
Alamat IP LAN Lokal	192.168.10.0/24	192.168.20.0/24
Peer *AllowedIPs*	10.200.200.2/32, 192.168.20.0/24	10.200.200.1/32, 192.168.10.0/24
Persistent Keepalive	25 detik	25 detik

Berdasarkan Tabel 4.1, kedua *router* dikonfigurasi dengan MTU 1420 *byte* untuk mengakomodasi *overhead* enkapsulasi *WireGuard* sebesar 60 *byte* sesuai perhitungan pada Subbab 2.3.3. Penyesuaian ini memastikan transmisi paket berukuran maksimum tetap berada dalam batas MTU fisik antarmuka Ethernet (1500 *byte*) tanpa memicu fragmentasi yang dapat membebani CPU.

Keberhasilan implementasi pada tahap ini menunjukkan bahwa konfigurasi telah diterapkan sesuai rancangan. Untuk memastikan *tunnel* benar-benar berfungsi secara fungsional, dilakukan validasi lebih lanjut yang dijelaskan pada Subbab 4.2.

4.2 Validasi Implementasi

Subbab 4.1 telah menunjukkan bahwa konfigurasi *WireGuard* berhasil diterapkan sesuai dengan rancangan penelitian, ditandai dengan status antarmuka (*interface*) yang aktif, konfigurasi *peer* yang sesuai, serta proses *handshake* yang berlangsung pada kedua *router*. Namun demikian, keberhasilan konfigurasi belum secara langsung membuktikan bahwa *tunnel* VPN telah berfungsi secara operasional. Oleh karena itu, dilakukan serangkaian validasi untuk memastikan

bahwa kedua *router* mampu saling berkomunikasi, meneruskan lalu lintas antarjaringan, serta membangun komunikasi melalui *tunnel* VPN sesuai rancangan topologi penelitian.

Tahap validasi implementasi dilakukan untuk memastikan bahwa seluruh konfigurasi protokol VPN telah berfungsi sesuai dengan rancangan sebelum dilakukan evaluasi kinerja. Validasi mencakup pembentukan *tunnel*, perutean antar-*subnet*, konektivitas antar-host, serta kemampuan *tunnel* dalam meneruskan trafik jaringan. Tahap ini diperlukan untuk memastikan bahwa pengujian kinerja dilakukan pada konfigurasi VPN yang telah memenuhi fungsi dasar, sehingga kegagalan konektivitas atau kesalahan konfigurasi tidak menjadi faktor utama yang memengaruhi hasil pengukuran performa. Secara keseluruhan, skenario pengujian fungsional beserta hasil implementasinya disajikan pada Tabel 4.2.

Tabel 4.2 Matriks Pengujian Fungsionalitas Protokol VPN

ID Skenario	**Skenario Pengujian**	**Hasil yang Diharapkan (*Expected*)**	**Hasil Riil Pengujian (*Actual*)**	**Status**
U-01	Inisiasi *Tunnel WireGuard*	Antarmuka *wg-sitetosite* aktif pada kedua *router* dan proses *handshake* berhasil terbentuk.	Antarmuka aktif dan *handshake* berhasil terbentuk serta tercatat pada RouterOS.	**Berhasil**
U-02	Inisiasi *Tunnel L2TP/IPsec*	Terowongan L2TP terbentuk dan memperoleh perlindungan IPsec sesuai konfigurasi yang ditetapkan.	Sesi L2TP berstatus aktif (*bound*) dan *Security Association* (SA) IPsec terbentuk.	**Berhasil**

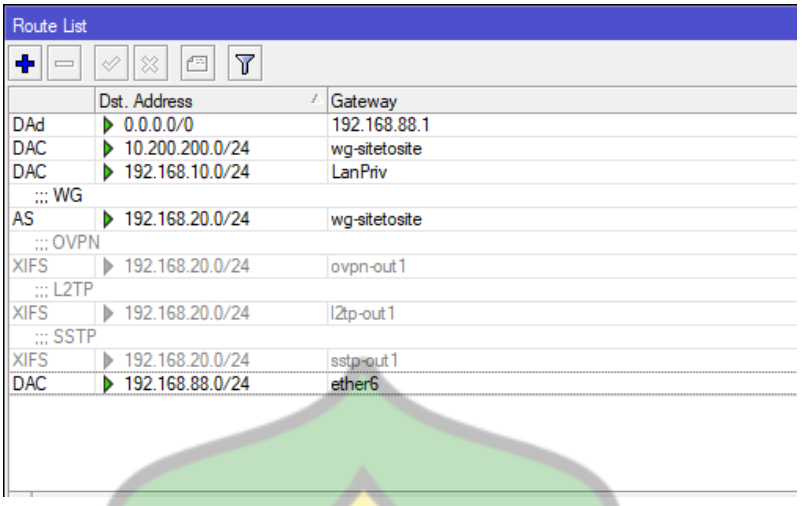
U-03	Inisiasi *Tunnel OpenVPN*	Antarmuka *ovpn-out* berhasil terhubung ke peladen menggunakan protokol UDP sesuai konfigurasi.	*Tunnel* berhasil terkoneksi dan dapat mempertahankan sesi komunikasi pada port UDP 1194.	**Berhasil**
U-04	Inisiasi *Tunnel SSTP*	Antarmuka *sstp-out* berhasil terkoneksi ke *router* melalui port TCP 443.	Koneksi SSTP berhasil terbentuk melalui port TCP 443.	**Berhasil**
U-05	Perutean Lintas *Subnet* (HQ–*Branch*)	*Router* mampu meneruskan paket antara LAN HQ (192.168.10.0/24) dan LAN *Branch* (192.168.20.0/24) melalui *tunnel* VPN.	Rute menuju jaringan tujuan tersedia dan koneksi lintas *subnet* dapat diteruskan melalui konfigurasi VPN.	**Berhasil**
U-06	Pengujian Konektivitas (ICMP Ping)	Pengiriman paket ICMP dari PC *Branch* menuju PC HQ menghasilkan *ping reply* tanpa *packet loss*.	Seluruh paket ICMP memperoleh respons dengan persentase kehilangan paket sebesar 0%.	**Berhasil**
U-07	Verifikasi Jalur (*Traceroute*)	Jalur komunikasi antara PC *Branch* dan PC HQ menunjukkan penggunaan jalur jaringan yang telah dikonfigurasi melalui *tunnel* VPN.	Hasil *traceroute* menunjukkan jalur komunikasi melalui *gateway* dan antarmuka VPN sesuai konfigurasi jaringan.	**Berhasil**

U-08	Stabilitas *Tunnel* di bawah Trafik	*Tunnel* tetap aktif dan mampu meneruskan trafik TCP/UDP tanpa pemutusan koneksi selama pengujian awal.	Trafik TCP dan UDP berhasil diteruskan dan seluruh *tunnel* tetap aktif selama pembebanan awal.	**Berhasil**

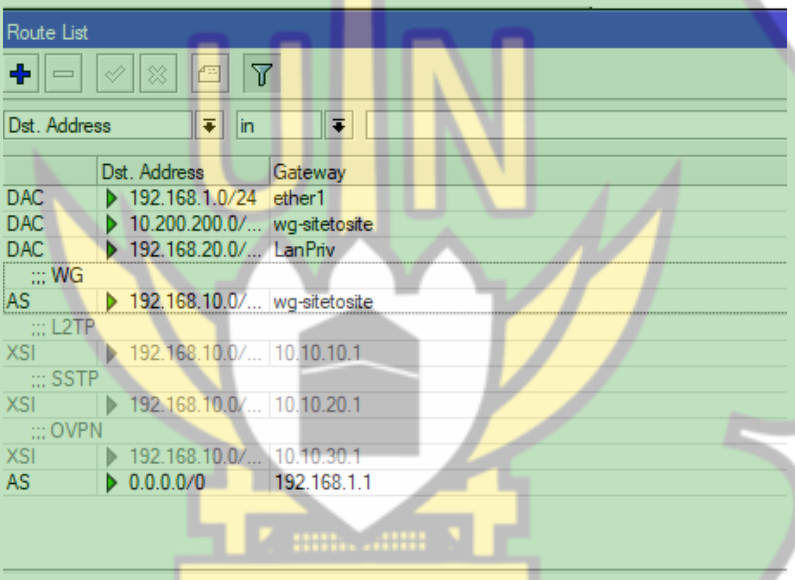
Berdasarkan Tabel 4.2, seluruh delapan skenario validasi memperoleh status Berhasil. Hasil tersebut menunjukkan bahwa seluruh protokol VPN yang digunakan telah berhasil diimplementasikan dan memenuhi fungsi dasar yang diperlukan untuk komunikasi *Site-to-Site*. Pembentukan *tunnel*, perutean antar-*subnet*, konektivitas ICMP, verifikasi jalur komunikasi, serta kemampuan meneruskan trafik selama pengujian awal telah berhasil diverifikasi.

Dengan terpenuhinya seluruh fungsi dasar tersebut, konfigurasi jaringan dinyatakan layak digunakan pada tahap pengujian performa. Hasil validasi ini meminimalkan kemungkinan bahwa kegagalan konektivitas atau kesalahan konfigurasi menjadi penyebab utama perbedaan hasil pada pengukuran QoS dan utilisasi sumber daya perangkat.

Validasi pertama dilakukan dengan memeriksa tabel *routing* pada kedua *router*. Hasil pada Gambar 4.5 dan Gambar 4.6 menunjukkan bahwa rute menuju *subnet* di seberang *tunnel*, yaitu jaringan 192.168.20.0/24 pada Router HQ dan 192.168.10.0/24 pada Router *Branch*, terbentuk melalui antarmuka *wg-sitetosite* sesuai konfigurasi *AllowedIPs* yang telah dijelaskan pada Subbab 3.5. Terbentuknya rute tersebut menunjukkan bahwa kedua *router* telah mengenali jalur menuju jaringan tujuan melalui *tunnel WireGuard*.



Gambar 4.5 Tabel *Routing* Router HQ



Gambar 4.6 Tabel *Routing* Routing *Branch*

Sebagai verifikasi bahwa rute tersebut benar-benar digunakan untuk meneruskan lalu lintas data, dilakukan pemeriksaan terhadap tabel *connection tracking*. Hasil pada Gambar 4.7 dan Gambar 4.8 menunjukkan adanya sesi koneksi TCP aktif antara perangkat pada jaringan *Branch* (192.168.20.254) dan perangkat pada jaringan HQ (192.168.10.1). Informasi tersebut membuktikan bahwa paket yang dikirimkan telah melewati *tunnel* VPN dan dikenali oleh mekanisme pelacakan koneksi (*stateful connection tracking*) pada RouterOS.

```
[admin@Router-A-HQ] > /ip firewall connection print where dst-address~"192.168.10"
Flags: S - SEEN-REPLY; A - ASSURED; C - CONFIRMED
Columns: PROTOCOL, SRC-ADDRESS, DST-ADDRESS, TCP-STATE, TIMEOUT, ORIG-RATE, REPL-RATE, ORIG-PACKETS
#     PRO  SRC-ADDRESS           DST-ADDRESS         TCP-STATE    TIMEOUT    ORIG-RATE  REPL-RATE  ORIG-P
2 SAC tcp  192.168.20.254:54073  192.168.10.1:8291   established  23h59m59s  33.4kbps   170.4kbps  34 064
```

Gambar 4.7 *Connection Tracking* Lintas-*Subnet* Router HQ

```
[admin@Router-B-Branch] > /ip firewall connection print where dst-address~"192.168.10"
Flags: S - SEEN-REPLY; A - ASSURED; C - CONFIRMED
Columns: PROTOCOL, SRC-ADDRESS, SRC-PORT, DST-ADDRESS, DST-PORT, TCP-STATE, TIMEOUT, ORIG-RA>
#     PRO  SRC-ADDRESS     SRC-PORT  DST-ADDRESS   DST-PORT  TCP-STATE    TIMEOUT    ORIG-RA>
0 SAC tcp  192.168.20.254     54073  192.168.10.1      8291  established  23h59m59s  5.9kbps
```

Gambar 4.8 *Connection Tracking* Lintas-*Subnet* Router *Branch*

Selanjutnya dilakukan pengujian konektivitas menggunakan perintah *ping* dan *traceroute* dari perangkat klien pada jaringan *Branch* menuju perangkat klien pada jaringan HQ. Hasil pengujian pada Gambar 4.9 menunjukkan bahwa seluruh paket berhasil mencapai tujuan tanpa kehilangan paket (*packet loss*), sedangkan hasil *traceroute* memperlihatkan bahwa lalu lintas hanya melewati tiga *hop*, yaitu *gateway* LAN *Branch*, *endpoint tunnel WireGuard*, dan perangkat tujuan pada jaringan HQ. Hasil tersebut menunjukkan bahwa komunikasi antarjaringan berlangsung secara transparan melalui *tunnel* VPN sebagaimana rancangan topologi penelitian.

```
C:\Users\hp>ping 192.168.10.254

Pinging 192.168.10.254 with 32 bytes of data:
Reply from 192.168.10.254: bytes=32 time=5ms TTL=126
Reply from 192.168.10.254: bytes=32 time=5ms TTL=126
Reply from 192.168.10.254: bytes=32 time=5ms TTL=126
Reply from 192.168.10.254: bytes=32 time=5ms TTL=126

Ping statistics for 192.168.10.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 5ms, Average = 5ms

C:\Users\hp>tracert 192.168.10.254

Tracing route to 192.168.10.254 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  192.168.20.1
  2     4 ms     3 ms     3 ms  10.200.200.1
  3    48 ms     4 ms     4 ms  192.168.10.254

Trace complete.
```

Gambar 4.9 Hasil Pengujian *Ping* dan *Traceroute* Antar-Klien

Selain pengujian konektivitas, dilakukan validasi tambahan menggunakan fitur *Speed Test* pada RouterOS secara langsung antara Router HQ dan Router

Branch. Berbeda dengan pengujian menggunakan *iperf3* yang melibatkan perangkat klien sebagai pembangkit lalu lintas, *Speed Test* RouterOS membangkitkan dan memproses lalu lintas secara langsung pada kedua *router*. Oleh karena itu, pengujian ini digunakan untuk memverifikasi bahwa implementasi VPN mampu membangun komunikasi antar-*router* sekaligus menunjukkan kemampuan *router* dalam memproses lalu lintas melalui *tunnel* VPN sebelum dilakukan evaluasi kinerja menggunakan *iperf3*. Hasil pengujian *Speed Test* ditunjukkan pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Hasil Validasi *Speed Test* RouterOS TCP

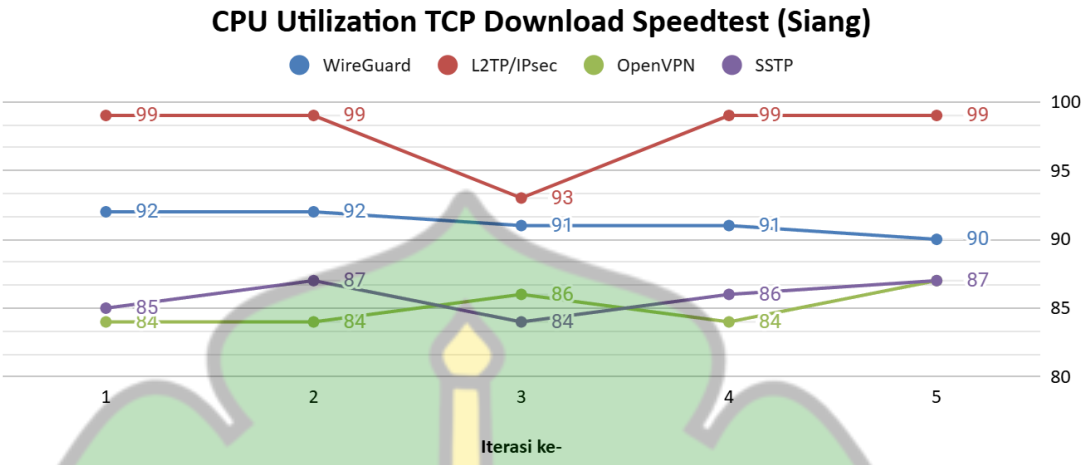
Kondisi	Protokol VPN	TCP Down (Mbps)	CPU Local (%)	TCP Up (Mbps)	CPU Local (%)
Siang	*WireGuard*	18,54	91,2	9,274	46
	L2TP/IPsec	13,62	97,8	8,334	80,8
	OpenVPN	9,438	85	8,74	84,6
	SSTP	11,56	85,8	8,89	70
Malam	*WireGuard*	18,42	92	9,268	52,4
	L2TP/IPsec	15,8	98	8,186	80
	OpenVPN	9,37	85,2	8,776	85,4
	SSTP	11,32	86,2	8,762	69,6

Tabel 4.4 Hasil Validasi *Speed Test* RouterOS UDP

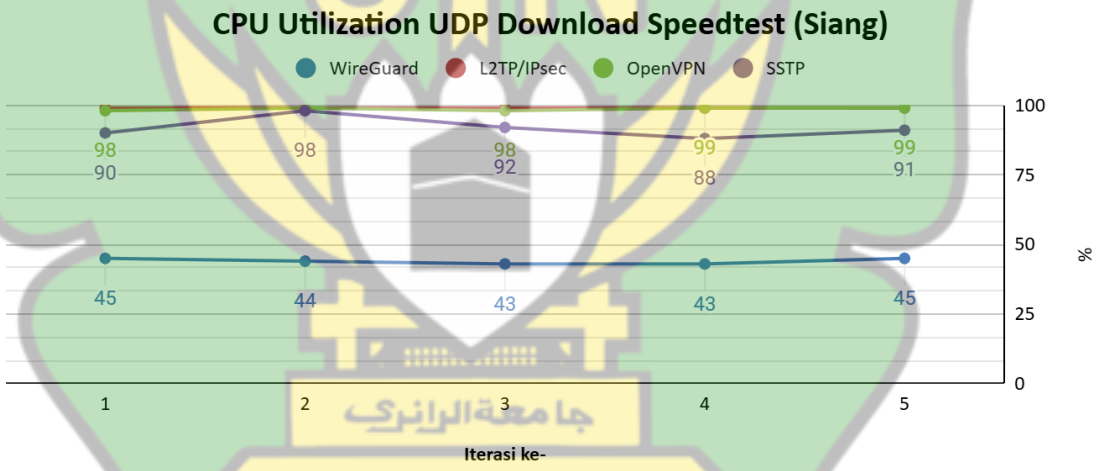
Kondisi	Protokol VPN	UDP Down (Mbps)	CPU Local (%)	UDP Up (Mbps)	CPU Local (%)
Siang	*WireGuard*	17,38	44	8,942	18,6
	L2TP/IPsec	17,66	99	6,528	40,6
	OpenVPN	12,26	98,6	9,004	68,4
	SSTP	16,52	91,8	9,15	57
Malam	*WireGuard*	17,56	44,6	8,964	19,4
	L2TP/IPsec	17,14	99,4	5,02	27,6
	OpenVPN	12,4	98,8	8,99	68,6
	SSTP	16,86	93,4	9,128	58,8

Berdasarkan hasil pada Tabel 4.2 dan Tabel 4.3, seluruh protokol VPN berhasil membangun komunikasi secara langsung antar-*router* melalui *tunnel* VPN yang telah dikonfigurasi. *WireGuard* menghasilkan *throughput* TCP sebesar 18,54 Mbps pada arah *download* dan 9,27 Mbps pada arah *upload*, sedangkan L2TP/IPsec, *OpenVPN*, dan SSTP juga menunjukkan bahwa komunikasi

antar-*router* berlangsung dengan baik meskipun menghasilkan *throughput* yang berbeda.



Gambar 4.10 CPU Utilization TCP Download Speedtest (Siang)



Gambar 4.11 CPU Utilization UDP Download Speedtest (Siang)

Selama proses pengujian, RouterOS menampilkan informasi bahwa hasil *Speed Test* dapat dipengaruhi oleh kemampuan prosesor perangkat (*results can be limited by CPU*). Pada pengujian malam hari, *WireGuard* menghasilkan *throughput* TCP sebesar 18,42 Mbps dengan utilisasi CPU sekitar 92%, sedangkan L2TP/IPsec menghasilkan *throughput* sebesar 15,80 Mbps dengan utilisasi CPU yang mencapai sekitar 98%. Hasil tersebut menunjukkan bahwa

WireGuard mampu memanfaatkan kapasitas *bandwidth* yang tersedia dengan kebutuhan sumber daya komputasi yang lebih rendah dibandingkan L2TP/IPsec pada kondisi pengujian yang sama. Adapun hasil validasi pada pengujian siang hari menunjukkan kecenderungan yang serupa dan disajikan pada Tabel 4.3, sedangkan data pengujian malam hari disajikan pada Tabel 4.4.

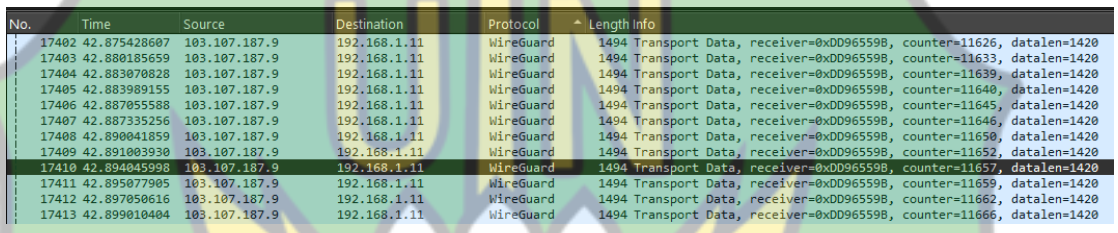
Perbedaan utilisasi CPU antara pengujian *Speed Test* RouterOS dan pengujian menggunakan *iPerf3* disebabkan oleh perbedaan mekanisme pembangkitan trafik pada kedua metode. *Speed Test* RouterOS menjalankan proses pembangkitan dan penerimaan trafik secara langsung pada perangkat router (*on the device*), sehingga CPU router tidak hanya melakukan fungsi *routing* serta enkripsi/dekripsi VPN, tetapi juga menangani proses pembangkitan trafik pengujian. Sebaliknya, pengujian *iPerf3* dilakukan secara *host-to-host*, dengan PC dan laptop bertindak sebagai *endpoint* serta *traffic generator*, sehingga router yang dipantau terutama menangani proses forwarding dan pemrosesan VPN.

Perbedaan mekanisme tersebut menyebabkan utilisasi CPU pada *Speed Test* tidak dapat dibandingkan secara langsung dengan utilisasi CPU pada *iPerf3*. Oleh karena itu, hasil *Speed Test* dalam penelitian ini diposisikan sebagai validasi tambahan terhadap fungsi komunikasi antar-*router*, sedangkan *iPerf3* digunakan sebagai instrumen utama dalam evaluasi performa karena pembangkitan trafik dilakukan pada perangkat *endpoint* dan tidak menempatkan router yang diuji sebagai pembangkit trafik. Hasil ini juga menjelaskan mengapa utilisasi CPU pada *Speed Test* dapat mencapai sekitar 91–92% untuk *WireGuard*, sementara pada pengujian *iPerf3* nilainya lebih rendah.

Perlu ditekankan bahwa hasil *Speed Test* pada penelitian ini tidak digunakan sebagai parameter utama evaluasi kinerja, melainkan sebagai validasi bahwa implementasi VPN telah berfungsi dengan baik dan bahwa kedua *router* mampu membangkitkan serta memproses lalu lintas melalui *tunnel* VPN. Evaluasi kinerja utama pada penelitian ini tetap dilakukan menggunakan *iperf3* sebagaimana metode pengujian yang telah dijelaskan pada Bab III.

Sebagai validasi tambahan terhadap mekanisme keamanan *WireGuard*, dilakukan *packet capture* menggunakan *Wireshark* pada antarmuka WAN selama sesi komunikasi VPN berlangsung. Pengambilan paket ini bertujuan untuk memverifikasi bahwa lalu lintas yang melewati jaringan publik benar-benar menggunakan protokol *WireGuard* serta memastikan bahwa data yang ditransmisikan berada dalam kondisi terenkripsi.

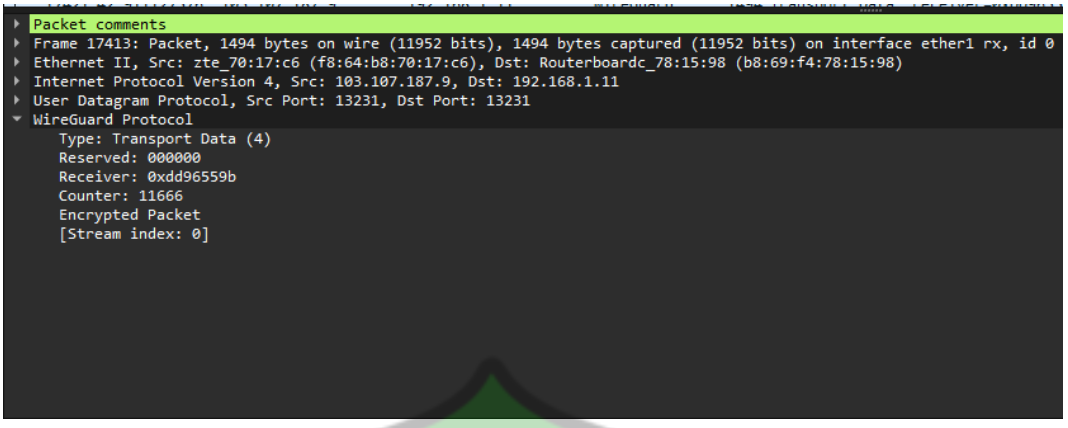
Hasil *packet capture* ditunjukkan pada Gambar 4.12. *Wireshark* berhasil mengidentifikasi paket yang melintas sebagai protokol *WireGuard* dengan tipe *Transport Data*, yang menunjukkan bahwa paket tersebut merupakan lalu lintas data setelah proses *handshake* berhasil dilakukan dan *tunnel* VPN berada pada kondisi aktif.



No.	Time	Source	Destination	Protocol	Length	Info
17402	42.875428607	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11626, datalen=1420
17403	42.880185659	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11633, datalen=1420
17404	42.883070828	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11639, datalen=1420
17405	42.883989155	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11640, datalen=1420
17406	42.887055588	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11645, datalen=1420
17407	42.887335256	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11646, datalen=1420
17408	42.890041859	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11650, datalen=1420
17409	42.891003930	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11652, datalen=1420
17410	42.894045998	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11657, datalen=1420
17411	42.895077905	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11659, datalen=1420
17412	42.897050616	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11662, datalen=1420
17413	42.899010404	103.107.187.9	192.168.1.11	WireGuard	1494	Transport Data, receiver=0xDD96559B, counter=11666, datalen=1420

Gambar 4.12 Hasil *Packet Capture WireGuard* pada Antarmuka WAN

Untuk memverifikasi proses enkripsi, dilakukan inspeksi lebih lanjut terhadap salah satu paket *WireGuard* sebagaimana ditunjukkan pada Gambar 4.13. Hasil inspeksi memperlihatkan bahwa *Wireshark* hanya dapat menampilkan informasi pada bagian *header* protokol, seperti *Type*, *Receiver Index*, dan *Counter*, sedangkan isi paket ditampilkan sebagai *Encrypted Packet*. Hal ini menunjukkan bahwa *payload* tidak dapat dibaca secara langsung karena telah dienkripsi sebelum dikirimkan melalui jaringan publik. Temuan tersebut sesuai dengan mekanisme *WireGuard* yang mengenkripsi seluruh muatan paket (*payload*) dan hanya mempertahankan informasi *header* yang diperlukan untuk proses identifikasi dan pengiriman paket.



Gambar 4.13 Detail Paket *WireGuard* yang Menunjukkan *Payload* Terenkripsi

Berdasarkan seluruh tahapan validasi yang telah dilakukan, meliputi verifikasi konfigurasi dan status *tunnel*, pembentukan *routing* antarjaringan, pelacakan koneksi, pengujian konektivitas menggunakan *ping* dan *traceroute*, validasi *Speed Test* RouterOS, serta verifikasi proses enkripsi menggunakan *Wireshark*, dapat disimpulkan bahwa implementasi *WireGuard Site-to-Site* telah berjalan sesuai rancangan penelitian. Dengan demikian, lingkungan pengujian dinyatakan siap untuk dilakukan evaluasi kinerja dan analisis komparatif terhadap *WireGuard*, L2TP/IPsec, *OpenVPN*, dan SSTP sebagaimana dibahas pada Subbab 4.3.

4.3 Evaluasi Kinerja VPN *Site-to-Site*

Setelah seluruh protokol VPN dinyatakan memenuhi fungsi dasar pada Subbab 4.2, tahap selanjutnya adalah melakukan evaluasi kinerja secara komparatif pada lingkungan pengujian yang sama. Evaluasi dilakukan terhadap *WireGuard*, L2TP/IPsec, *OpenVPN* (UDP), dan SSTP menggunakan topologi, konfigurasi, serta skenario pengujian yang identik sebagaimana dijelaskan pada Bab III. Pengujian dilakukan pada kondisi siang dan malam untuk mengamati karakteristik performa protokol pada kondisi jaringan ISP asimetris yang digunakan dalam penelitian.

Parameter evaluasi meliputi *throughput*, *latency*, *jitter*, *packet loss*, utilisasi CPU, dan utilisasi RAM. Keenam parameter tersebut dipilih karena mewakili dua aspek utama dalam implementasi VPN, yaitu kualitas layanan

jaringan (*Quality of Service/QoS*) dan efisiensi penggunaan sumber daya perangkat. Melalui kombinasi kedua aspek tersebut, penelitian ini bertujuan memperoleh gambaran yang komprehensif mengenai karakteristik performa masing-masing protokol VPN pada router MikroTik berarsitektur MIPSBE *single-core*. Rangkuman pelaksanaan skenario pengujian performa beserta hasil riil dan status pengujiannya disajikan pada Tabel 4.5.

Tabel 4.5 Matriks Pengujian Performa Protokol VPN

ID	Skenario	Parameter Uji	Expected	Actual	Status
P-01	TCP *Single Stream*	Throughput, CPU, RAM	Pengujian dapat dijalankan dan seluruh parameter berhasil direkam.	Data berhasil diperoleh pada 20 replikasi, dua arah, siang dan malam.	Berhasil
P-02	TCP *Multi Stream*	Throughput, CPU, RAM	Pengujian empat aliran TCP paralel dapat dijalankan dan seluruh parameter berhasil direkam.	Data berhasil diperoleh pada 20 replikasi, dua arah, siang dan malam.	Berhasil
P-03	UDP 5 Mbps	Throughput, Jitter, Packet Loss, CPU, RAM	Trafik UDP 5 Mbps dapat diteruskan dan seluruh parameter berhasil direkam.	Data berhasil diperoleh pada 20 replikasi, dua arah, siang dan malam.	Berhasil
P-04	UDP 10 Mbps (*Overload*)	Throughput, Jitter, Packet Loss, CPU, RAM	Trafik UDP 10 Mbps dapat dibangkitkan sebagai pembebanan tinggi dan seluruh parameter berhasil direkam.	Data berhasil diperoleh pada 20 replikasi, dua arah, siang dan malam.	Berhasil
P-05	ICMP Ping	Latency	Pengukuran ICMP dapat dilakukan dan data *latency* berhasil direkam.	Data berhasil diperoleh pada 20 replikasi pada kondisi siang dan malam.	Berhasil

4.3.1 Throughput (TCP *Single Stream* dan TCP *Multi Stream*)

Throughput digunakan untuk mengevaluasi kemampuan masing-masing protokol VPN dalam mentransmisikan data melalui *tunnel* VPN. Pengujian dilakukan menggunakan dua skenario, yaitu TCP *Single Stream* dan TCP *Multi Stream*, dengan arah lalu lintas *Normal* (*upload*) dan *Reverse* (*download*). Seluruh pengujian dilaksanakan pada dua kondisi waktu yang berbeda, yaitu siang dan malam, untuk melihat konsistensi performa terhadap variasi kondisi jaringan ISP. Rangkuman hasil pengujian disajikan pada Tabel 4.6 dan Tabel 4.7.

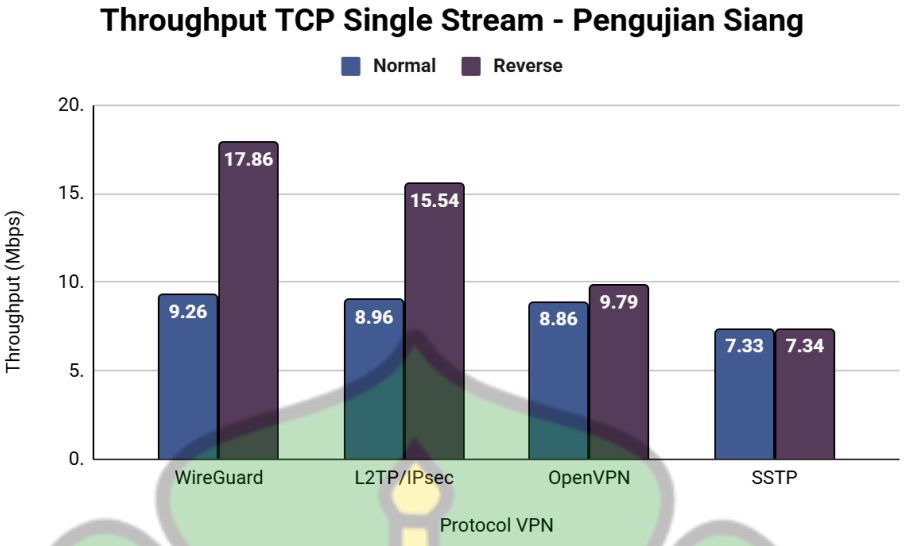
Tabel 4.6 *Throughput* TCP Stream (Siang)

Protocol VPN	Single N (Mbps)	Single R (Mbps)	Multi N (Mbps)	Multi R (Mbps)
WireGuard	9,26	17,86	9,47	17,39
L2TP/IPsec	8,96	15,54	9,12	15,55
OpenVPN	8,86	9,79	9,05	10,09
SSTP	7,33	7,34	8,66	7,95

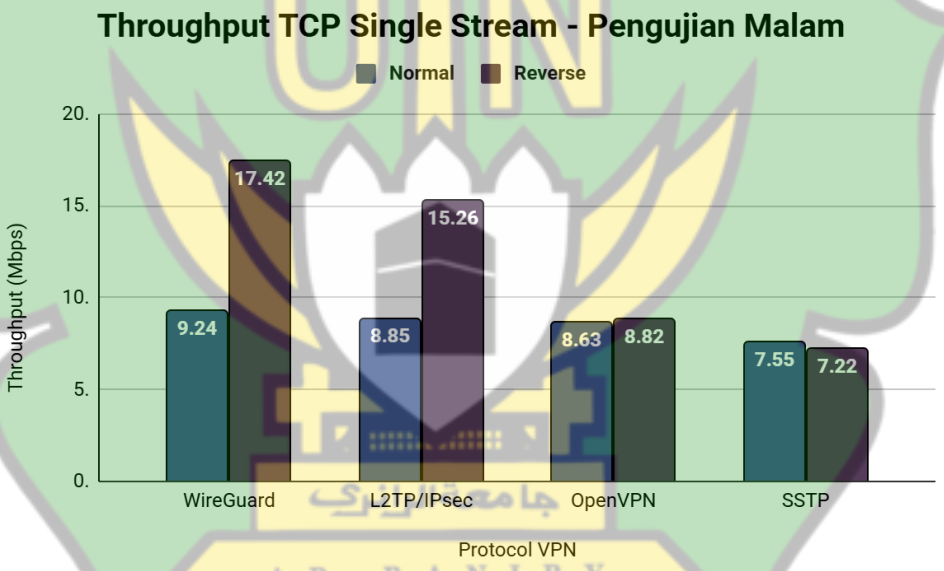
Tabel 4.7 *Throughput* TCP Stream (Malam)

Protocol VPN	Single N (Mbps)	Single R (Mbps)	Multi N (Mbps)	Multi R (Mbps)
WireGuard	9,24	17,42	9,54	18,68
L2TP/IPsec	8,85	15,26	9,06	16,67
OpenVPN	8,63	8,82	9,03	9,52
SSTP	7,55	7,22	8,31	7,45

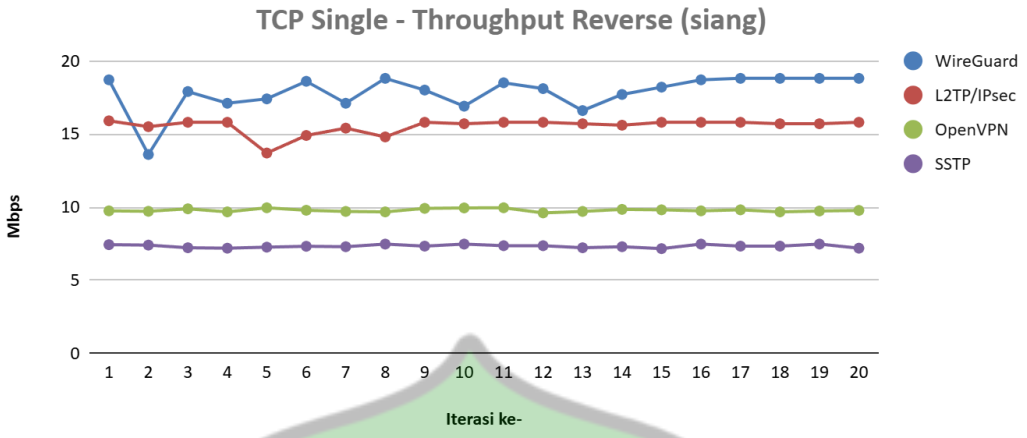
Untuk mempermudah perbandingan antarprotokol, hasil pengujian TCP *Single Stream* divisualisasikan menggunakan grafik batang pada Gambar 4.14 dan Gambar 4.15. Selain itu, sebagai representasi peforma hasil pengujian, Gambar 4.16 menampilkan variasi *throughput* antariterasi pada skenario TCP *Single Stream* arah *Reverse*. Visualisasi tersebut digunakan untuk menunjukkan bahwa nilai rata-rata *throughput* yang diperoleh berasal dari hasil pengujian yang konsisten dengan fluktuasi yang relatif kecil.



Gambar 4.14 *Throughput* TCP *Single Stream* (Siang)



Gambar 4.15 *Throughput* TCP *Single Stream* (Malam)



Gambar 4.16 Variasi hasil antar iterasi

Berdasarkan Gambar 4.16, *throughput* masing-masing protokol menunjukkan pola yang relatif stabil selama 20 iterasi pengujian. *WireGuard* mempertahankan *throughput* tertinggi dengan fluktuasi yang kecil di sekitar nilai rata-ratanya, diikuti oleh L2TP/IPsec, *OpenVPN*, dan SSTP yang juga memperlihatkan pola serupa pada tingkat *throughput* masing-masing. Kondisi tersebut menunjukkan bahwa nilai rata-rata *throughput* yang diperoleh merepresentasikan performa masing-masing protokol secara konsisten dan tidak dipengaruhi oleh fluktuasi pengukuran yang berarti antariterasi.

Berdasarkan Tabel 4.6 dan Tabel 4.7, seluruh protokol menunjukkan pola yang serupa, yaitu *throughput* pada arah *Reverse* (*download*) selalu lebih tinggi dibandingkan arah *Normal* (*upload*). Pola tersebut konsisten baik pada pengujian siang maupun malam serta berkaitan dengan karakteristik koneksi ISP yang digunakan pada penelitian ini, di mana kapasitas *download* (19,35 Mbps) lebih besar dibandingkan kapasitas *upload* (9,67 Mbps). Dengan demikian, kapasitas *bandwidth* yang tersedia pada arah *download* memungkinkan seluruh protokol memperoleh *throughput* yang lebih tinggi dibandingkan arah *upload*.

Pada skenario TCP *Single Stream*, *WireGuard* secara konsisten menghasilkan *throughput* rata-rata tertinggi dibandingkan ketiga protokol pembanding. Pada pengujian siang, *WireGuard* memperoleh *throughput* sebesar 9,26 Mbps pada arah *Normal* dan 17,86 Mbps pada arah *Reverse*. Hasil yang relatif serupa juga diperoleh pada pengujian malam, yaitu 9,24 Mbps pada arah

Normal dan 17,42 Mbps pada arah *Reverse*. Nilai tersebut berada di atas L2TP/IPsec, *OpenVPN*, maupun SSTP pada skenario yang sama, menunjukkan bahwa *WireGuard* mampu memanfaatkan kapasitas *bandwidth* yang tersedia secara lebih efektif pada komunikasi TCP satu aliran (*single stream*).

Pola yang serupa juga terlihat pada pengujian TCP *Multi Stream*. *WireGuard* menghasilkan *throughput* rata-rata sebesar 9,47 Mbps pada arah *Normal* dan 17,39 Mbps pada arah *Reverse* saat pengujian siang, kemudian meningkat menjadi 9,54 Mbps dan 18,68 Mbps pada pengujian malam. Sebaliknya, *OpenVPN* dan SSTP menunjukkan peningkatan *throughput* yang lebih terbatas pada arah *Reverse*. Sebagai contoh, *throughput OpenVPN* meningkat dari 9,05 Mbps menjadi 10,09 Mbps pada pengujian siang, sedangkan SSTP hanya mencapai 7,95 Mbps. Hasil tersebut menunjukkan bahwa pemanfaatan kapasitas *bandwidth* pada kedua protokol tersebut tidak seoptimal *WireGuard* pada skenario pengujian yang dilakukan.

Jika dibandingkan antara pengujian siang dan malam, perbedaan *throughput* pada seluruh protokol relatif kecil. Secara deskriptif, hasil tersebut tidak menunjukkan perubahan performa yang berarti akibat variasi waktu pengujian. Temuan ini mengindikasikan bahwa karakteristik implementasi masing-masing protokol VPN memberikan pengaruh yang lebih dominan terhadap *throughput* dibandingkan perubahan kondisi jaringan ISP pada rentang waktu pengujian.

Hasil pengujian tersebut sejalan dengan karakteristik implementasi *WireGuard* yang telah dijelaskan pada Bab II. *WireGuard* beroperasi di dalam *kernel-space*, sehingga proses enkripsi dan dekripsi dilakukan tanpa perpindahan konteks (*context switching*) yang umumnya terjadi pada VPN berbasis *user-space* seperti *OpenVPN*. Selain itu, mekanisme protokol yang lebih sederhana dibandingkan L2TP/IPsec maupun SSTP menyebabkan *overhead* pemrosesan menjadi lebih rendah, sehingga kapasitas *bandwidth* yang tersedia dapat dimanfaatkan secara lebih efisien pada perangkat MikroTik berarsitektur MIPSBE *single-core*. Temuan ini juga konsisten dengan konsep implementasi *WireGuard* yang dikemukakan oleh Donenfeld (2017), bahwa integrasi protokol ke dalam

kernel memungkinkan proses transmisi paket berlangsung dengan *overhead* yang lebih rendah dibandingkan implementasi VPN pembanding.

Berdasarkan seluruh skenario pengujian TCP, *WireGuard* secara konsisten menghasilkan *throughput* rata-rata tertinggi dibandingkan L2TP/IPsec, *OpenVPN*, dan SSTP. Temuan tersebut menunjukkan bahwa *WireGuard* mampu memanfaatkan kapasitas *bandwidth* yang tersedia secara lebih efektif pada perangkat MikroTik MIPSBE *single-core*. Namun demikian, *throughput* bukan merupakan satu-satunya parameter yang menentukan kualitas implementasi VPN. Oleh karena itu, pada subbab berikutnya dilakukan evaluasi terhadap parameter *latency*, *jitter*, *packet loss*, serta utilisasi sumber daya perangkat untuk memperoleh gambaran yang lebih komprehensif mengenai performa masing-masing protokol.

4.3.2 *Latency*

Latency merupakan salah satu parameter *Quality of Service* (QoS) yang digunakan untuk mengukur waktu tempuh paket dari perangkat pengirim menuju perangkat tujuan melalui *tunnel* VPN. Semakin rendah nilai *latency*, semakin cepat respons komunikasi yang dihasilkan. Pada penelitian ini, pengukuran dilakukan menggunakan paket ICMP sebagaimana dijelaskan pada Subbab 3.6 untuk membandingkan tingkat responsivitas masing-masing protokol VPN ketika beroperasi pada jaringan publik.

Rangkuman hasil pengujian *latency* pada kondisi siang dan malam disajikan pada Tabel 4.8. Perbandingan nilai rata-rata *latency* antarprotokol divisualisasikan pada Gambar 4.17, sedangkan pengujian Mann–Whitney U digunakan untuk mengevaluasi perbedaan hasil *latency* antara kondisi siang dan malam pada masing-masing protokol.

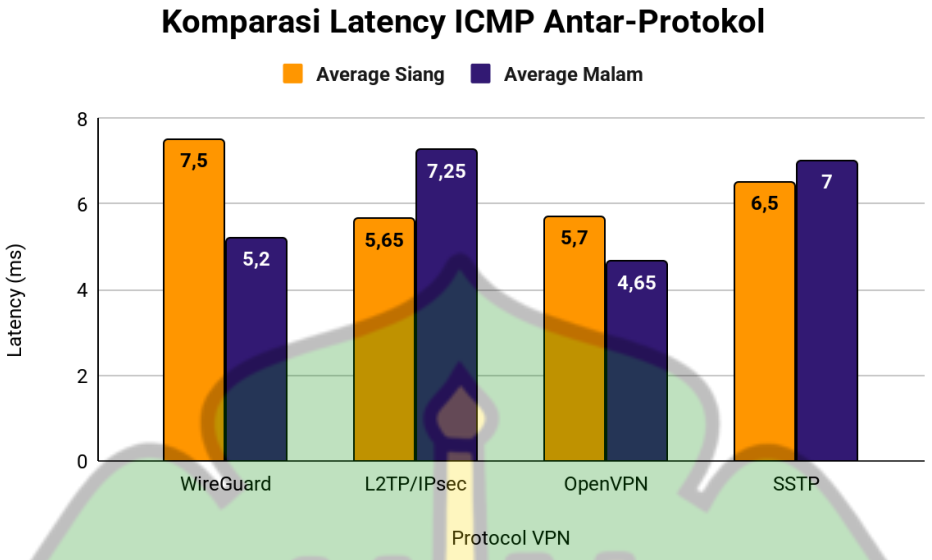
Tabel 4.8 Rangkuman Latency

Kondisi	Protocol VPN	Min (ms)	Max (ms)	Average (ms)	Packet Loss (%)
Siang	*WireGuard*	3	146	7.5	0
	L2TP/IPsec	4	149	5.65	0.1
	OpenVPN	4	153	5.7	0
	SSTP	4	241	6.5	0
Malam	*WireGuard*	3	43	5.2	0
	L2TP/IPsec	4	334	7.25	0.3
	OpenVPN	4	137	4.65	0
	SSTP	4	91	7	0

Pada penelitian ini, mean dan median digunakan untuk tujuan analisis yang berbeda. Mean digunakan untuk mendeskripsikan kecenderungan rata-rata *latency* pada masing-masing kondisi pengujian, sedangkan median digunakan sebagai statistik deskriptif tambahan karena *latency* pada jaringan publik dapat mengalami pencilan (*outlier*) dan lonjakan sesaat (*latency spikes*). Selanjutnya, perbedaan *latency* antara kondisi siang dan malam dianalisis menggunakan uji Mann–Whitney U berdasarkan seluruh observasi hasil pengukuran. Pemilihan uji nonparametrik ini didasarkan pada karakteristik data *latency* yang dapat mengalami fluktuasi akibat perubahan kondisi jaringan publik. Uji *Mann-Whitney U* tidak mensyaratkan distribusi data normal seperti yang diperlukan pada *independent samples t-test*, sehingga lebih sesuai untuk membandingkan dua kelompok pengukuran ketika asumsi parametrik tersebut tidak dapat dipastikan. Dalam penelitian ini, pengujian dilakukan pada taraf signifikansi $\alpha = 0,05$ dengan tingkat kepercayaan 95%, dan hasil dinyatakan signifikan apabila *p-value* < 0,05.

Tabel 4.9 Analisis *Mann-Whitney* Latency

VPN	Median (ms)		Statistik U	p-value	Signifikansi (α = 0,05)	Keterangan
	Siang	Malam				
WireGuard	7,00	5,00	354,5	0,000008	(p < 0,05)	Signifikan
L2TP/IPsec	5,00	6,00	152,5	0,165525	(p ≥ 0,05)	Tidak signifikan
OpenVPN	5,00	5,00	309,0	0,000397	(p < 0,05)	Signifikan
SSTP	5,50	7,00	123,0	0,033229	(p < 0,05)	Signifikan



Gambar 4.17 Average Latency Siang Malam

Berdasarkan Tabel 4.8, nilai rata-rata *latency* pada 20 pengujian berada pada rentang 4,65 ms hingga 7,50 ms. Pada kondisi siang, *WireGuard* menghasilkan rata-rata *latency* sebesar 7,50 ms, sedangkan L2TP/IPsec, *OpenVPN*, dan SSTP masing-masing menghasilkan 5,65 ms, 5,70 ms, dan 6,50 ms. Pada kondisi malam, rata-rata *latency* masing-masing protokol adalah 5,20 ms untuk *WireGuard*, 7,25 ms untuk L2TP/IPsec, 4,65 ms untuk *OpenVPN*, dan 7,00 ms untuk SSTP. Dengan demikian, berdasarkan nilai rata-rata, tidak terdapat satu protokol yang menghasilkan *latency* terendah pada seluruh kondisi pengujian.

Pada kondisi siang, *WireGuard* memiliki rata-rata *latency* tertinggi dibandingkan tiga protokol pembanding. Namun, kondisi tersebut berubah pada malam hari, ketika rata-rata *latency* *WireGuard* menurun menjadi 5,20 ms. Sebaliknya, L2TP/IPsec mengalami peningkatan rata-rata *latency* dari 5,65 ms pada siang hari menjadi 7,25 ms pada malam hari. *OpenVPN* menghasilkan rata-rata *latency* terendah pada malam hari sebesar 4,65 ms, sedangkan SSTP meningkat dari 6,50 ms menjadi 7,00 ms. Perubahan tersebut menunjukkan bahwa performa *latency* masing-masing protokol tidak hanya ditentukan oleh

karakteristik protokol, tetapi juga dapat dipengaruhi oleh kondisi jaringan pada waktu pengujian.

Nilai maksimum *latency* juga menunjukkan adanya variasi yang cukup besar pada beberapa pengujian. Pada kondisi siang, nilai maksimum *latency* *WireGuard*, L2TP/IPsec, *OpenVPN*, dan SSTP masing-masing mencapai 146 ms, 149 ms, 153 ms, dan 241 ms. Pada kondisi malam, nilai maksimum masing-masing adalah 43 ms, 334 ms, 137 ms, dan 91 ms. Nilai maksimum tersebut menunjukkan adanya *latency* spike pada sejumlah iterasi, terutama pada L2TP/IPsec malam hari yang mencapai 334 ms. Karena penelitian ini tidak melakukan pengamatan jalur transmisi dan kondisi jaringan ISP secara real-time pada saat setiap spike terjadi, lonjakan tersebut tidak dapat secara langsung diatribusikan sebagai karakteristik inheren dari protokol VPN.

Berdasarkan Tabel 4.9, *WireGuard*, *OpenVPN*, dan SSTP memiliki p-value kurang dari 0,05 sehingga terdapat perbedaan *latency* yang signifikan antara kondisi siang dan malam. Sebaliknya, L2TP/IPsec memiliki p-value sebesar 0,165525 sehingga tidak terdapat bukti yang cukup untuk menyatakan adanya perbedaan signifikan antara kedua kondisi waktu. Hasil tersebut menunjukkan bahwa pengaruh kondisi waktu terhadap *latency* tidak sama pada seluruh protokol.

Meskipun *OpenVPN* menghasilkan rata-rata *latency* terendah pada malam hari sebesar 4,65 ms, sedangkan *WireGuard* menghasilkan 5,20 ms, selisih keduanya sebesar 0,55 ms. Sebaliknya, *WireGuard* menunjukkan penurunan rata-rata *latency* dari 7,50 ms pada siang hari menjadi 5,20 ms pada malam hari. Dengan demikian, berdasarkan hasil deskriptif, *WireGuard* tidak dapat dinyatakan sebagai protokol dengan *latency* terendah pada seluruh kondisi pengujian. Namun, hasil pengujian menunjukkan bahwa *WireGuard* tetap mampu mempertahankan nilai *latency* yang rendah pada pengujian malam dan memiliki nilai maksimum yang lebih rendah dibandingkan ketiga protokol pembanding pada kondisi tersebut.

Hasil *latency* penelitian ini juga dapat dibandingkan dengan penelitian Ikhwandi dkk. (2025) yang menunjukkan adanya peningkatan *latency WireGuard* akibat kompleksitas rute lintas ISP, dari 29 ms menjadi 67 ms. Pada penelitian ini, *latency WireGuard* berada pada rata-rata 7,50 ms pada siang hari dan 5,20 ms pada malam hari. Perbedaan karakteristik tersebut menunjukkan bahwa perubahan *latency* pada *WireGuard* tidak hanya ditentukan oleh protokol VPN, tetapi juga dipengaruhi oleh kondisi jaringan dan kompleksitas rute yang dilalui. Dengan demikian, hasil penelitian ini memperkuat temuan bahwa performa *latency WireGuard* bersifat kontekstual terhadap lingkungan implementasi, sehingga hasil dari satu topologi jaringan tidak dapat digeneralisasikan secara langsung ke kondisi jaringan lainnya.

Secara keseluruhan, pengujian 20 iterasi menunjukkan bahwa karakteristik *latency* berbeda antara protokol dan kondisi waktu pengujian. *OpenVPN* menghasilkan rata-rata *latency* terendah pada kondisi malam, sedangkan *WireGuard* menunjukkan penurunan *latency* yang cukup besar dari kondisi siang ke malam. Hasil Mann–Whitney U juga menunjukkan bahwa perubahan antara siang dan malam signifikan pada *WireGuard*, *OpenVPN*, dan SSTP, tetapi tidak signifikan pada L2TP/IPsec. Temuan ini menunjukkan bahwa evaluasi *latency* perlu mempertimbangkan distribusi hasil dan variasi kondisi jaringan, bukan hanya membandingkan nilai rata-rata. Selanjutnya, evaluasi kualitas transmisi dilanjutkan melalui analisis *jitter* dan *packet loss* pada Subbab 4.3.3.

Jika dibandingkan dengan temuan Ikhwandi dkk. (2025), yang menunjukkan perubahan *latency WireGuard* pada kondisi rute jaringan yang berbeda, hasil penelitian ini menunjukkan median *latency* yang relatif rendah pada kedua periode pengujian. Perbedaan tersebut menunjukkan bahwa karakteristik *latency WireGuard* dipengaruhi oleh kondisi jalur dan lingkungan implementasi. Oleh karena itu, hasil *latency* pada penelitian ini perlu dipahami dalam konteks topologi, perangkat, dan karakteristik ISP yang digunakan

4.3.3 *Jitter* dan *Packet Loss*

Jitter dan *packet loss* merupakan parameter *Quality of Service* (QoS) yang digunakan untuk mengevaluasi kestabilan pengiriman paket pada lalu lintas berbasis UDP. Pada penelitian ini, pengukuran dilakukan menggunakan dua tingkat pembebanan, yaitu UDP 5 Mbps sebagai representasi beban lalu lintas sedang dan UDP 10 Mbps sebagai representasi beban yang mendekati kapasitas efektif *upload* ISP sebesar 9,67 Mbps.

Ringkasan hasil pengujian UDP 5 Mbps dan UDP 10 Mbps disajikan pada Tabel 4.10 dan Tabel 4.11. Perbandingan nilai *jitter* dan *packet loss* antarprotokol divisualisasikan pada Gambar 4.18 dan Gambar 4.19.

Tabel 4.10 beban UDP 5 Mbps

Kondisi	Protocol VPN	UDP 5 Mbps *Normal*			UDP 5 Mbps *Reverse*		
		Throughput (Mbps)	Jitter (ms)	Packet Loss (%)	Throughput (Mbps)	Jitter (ms)	Packet Loss (%)
Siang	*WireGuard*	4,996	1,855	0	4,999	0,979	0,003
	L2TP/IPsec	4,992	1,953	0,08	4,997	0,977	0,042
	OpenVPN	4,993	1,94	0,071	5	1,253	0,001
	SSTP	4,935	1,919	1,136	5	1,163	0,005
Malam	*WireGuard*	4,993	1,778	0	5	0,999	0,001
	L2TP/IPsec	4,999	1,784	0,008	5	0,934	0,005
	OpenVPN	4,995	1,888	0	5	1,268	0,001
	SSTP	4,946	1,854	0,864	5	1,392	0,003

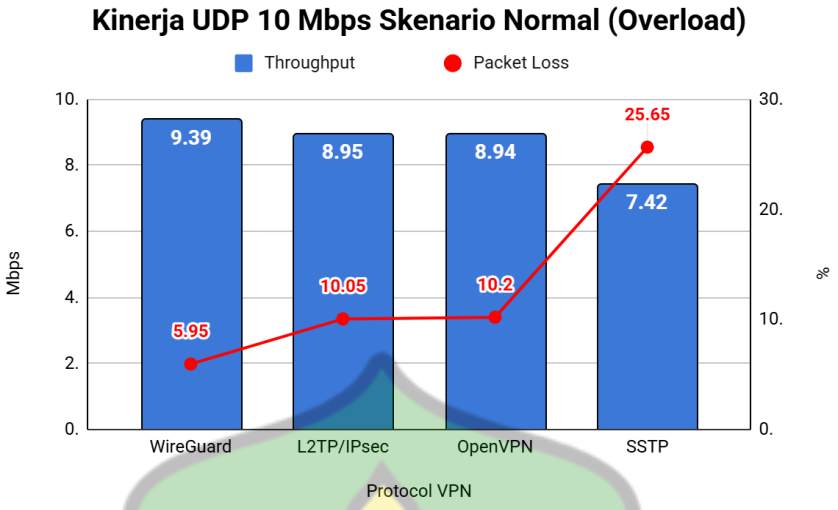
Tabel 4.11 beban UDP 10 Mbps

Kondisi	Protocol VPN	UDP 10 Mbps *Normal*			UDP 10 Mbps *Reverse*		
		Throughput (Mbps)	Jitter (ms)	Packet Loss (%)	Throughput (Mbps)	Jitter (ms)	Packet Loss (%)
Siang	*WireGuard*	9,385	1,743	5,95	9,703	0,98	2,182
	L2TP/IPsec	8,953	1,948	10,05	9,892	1,007	1,053
	OpenVPN	8,943	1,866	10,2	9,986	1,313	0,093
	SSTP	7,416	2,01	25,65	8,657	1,16	13,15
Malam	*WireGuard*	9,45	1,676	5,335	9,993	1,01	0,056
	L2TP/IPsec	8,96	1,921	10,05	9,993	1,042	0,037
	OpenVPN	8,922	1,926	10,5	9,984	1,38	0,074
	SSTP	7,4	1,977	25,75	8,635	1,622	13,15

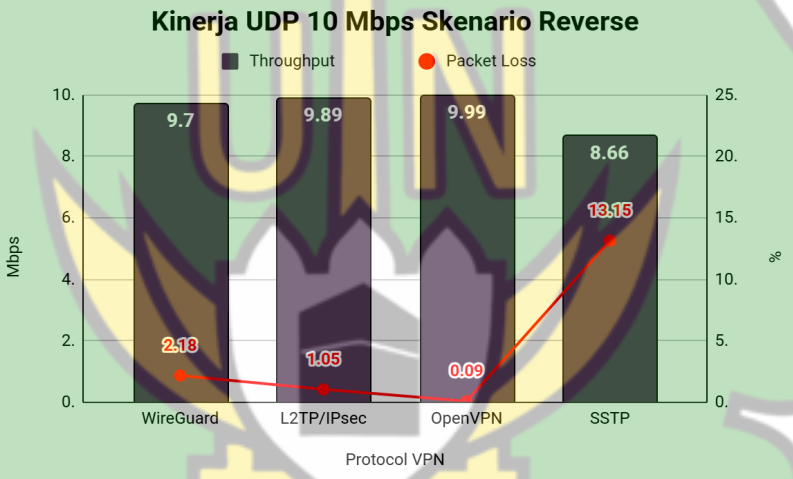
Berdasarkan Tabel 4.10, seluruh protokol mampu mempertahankan kualitas transmisi yang baik pada pembebanan UDP 5 Mbps. Nilai *throughput* seluruh protokol berada sangat dekat dengan laju injeksi sebesar 5 Mbps, sedangkan *packet loss* yang dihasilkan relatif rendah. Pada arah *Normal*, *WireGuard* tidak mengalami kehilangan paket (0%) pada hasil pengujian yang dirangkum, sedangkan L2TP/IPsec dan *OpenVPN* juga menghasilkan *packet loss* yang sangat kecil. SSTP menunjukkan *packet loss* tertinggi dibandingkan protokol lainnya, namun nilainya masih relatif rendah.

Nilai *jitter* pada skenario UDP 5 Mbps juga relatif rendah untuk seluruh protokol, yaitu berada pada kisaran 0,98 ms hingga 1,95 ms. Perbedaan nilai *jitter* antarprotokol relatif kecil sehingga secara deskriptif belum menunjukkan perbedaan kualitas layanan yang berarti. Kondisi tersebut mengindikasikan bahwa ketika lalu lintas masih berada di bawah kapasitas efektif koneksi ISP, seluruh protokol mampu mempertahankan kestabilan waktu kedatangan paket.

Perbedaan karakteristik antarprotokol mulai terlihat ketika laju injeksi ditingkatkan menjadi UDP 10 Mbps, yaitu mendekati kapasitas efektif *upload* ISP. Pada kondisi tersebut seluruh protokol mengalami peningkatan *packet loss*, namun besarnya peningkatan berbeda pada masing-masing protokol.



Gambar 4.18 Kinerja UDP 10 Mbps Skenario *Normal*



Gambar 4.19 Kinerja UDP 10 Mbps Skenario *Reverse*

Pada arah *Normal*, *WireGuard* menghasilkan *packet loss* sebesar 5,95% pada hasil pengujian yang dirangkum, lebih rendah dibandingkan L2TP/IPsec (10,05%), *OpenVPN* (10,20%), maupun SSTP (25,65%). Hasil pengujian pada kondisi malam yang disajikan menunjukkan kecenderungan yang serupa. Selain menghasilkan *packet loss* yang lebih rendah, *WireGuard* juga mempertahankan *throughput* tertinggi dibandingkan ketiga protokol pembanding. Temuan tersebut menunjukkan bahwa pada skenario *upload* yang mendekati kapasitas efektif koneksi ISP, *WireGuard* mampu mempertahankan keberhasilan pengiriman paket dengan lebih baik dibandingkan protokol lain yang diuji.

Pada arah *Reverse*, peningkatan kapasitas *download* ISP menyebabkan *packet loss* seluruh protokol menurun secara signifikan. Berdasarkan hasil pengujian siang hari yang dirangkum pada Tabel 4.9, *OpenVPN* menghasilkan *packet loss* terendah sebesar 0,093%, diikuti oleh L2TP/IPsec sebesar 1,053% dan *WireGuard* sebesar 2,182%, sedangkan SSTP tetap menunjukkan *packet loss* yang relatif tinggi. Fenomena rendahnya *packet loss OpenVPN* pada kondisi tertentu ini juga selaras dengan temuan Muhajir dan Utami (2026) yang menguji kedua protokol pada perangkat MikroTik, di mana *OpenVPN* sukses mencatatkan *packet loss* hingga 0% di seluruh observasi, sementara *WireGuard* mengalami sedikit kehilangan paket dengan rata-rata 0,524%. Namun demikian, hasil pengujian malam hari menunjukkan kecenderungan yang berbeda, di mana L2TP/IPsec menghasilkan *packet loss* terendah sebesar 0,037%, diikuti *WireGuard* sebesar 0,056% dan *OpenVPN* sebesar 0,074%. Perbedaan tersebut menunjukkan bahwa performa relatif antarprotokol pada skenario *Reverse* dapat dipengaruhi oleh variasi kondisi jaringan publik selama waktu pengujian. Meskipun demikian, seluruh protokol memperlihatkan penurunan *packet loss* dibandingkan arah *Normal*, yang menunjukkan bahwa kapasitas *download* ISP yang lebih besar memberikan pengaruh positif terhadap keberhasilan transmisi paket.

Berbeda dengan *packet loss*, perubahan nilai *jitter* akibat peningkatan pembebanan relatif kecil. Pada seluruh pengujian UDP 10 Mbps, nilai *jitter* seluruh protokol masih berada pada kisaran 1 ms hingga 2 ms. *WireGuard* mempertahankan nilai *jitter* sekitar 1,7 ms pada arah *Normal* dan sekitar 1 ms pada arah *Reverse*, sedangkan protokol lainnya menunjukkan nilai yang tidak jauh berbeda. Kondisi tersebut menunjukkan bahwa peningkatan pembebanan lebih memengaruhi tingkat kehilangan paket dibandingkan variasi waktu kedatangan paket.

Secara keseluruhan, peningkatan pembebanan dari UDP 5 Mbps menjadi UDP 10 Mbps menunjukkan bahwa *packet loss* merupakan parameter yang paling sensitif dalam membedakan performa masing-masing protokol ketika lalu lintas mulai mendekati kapasitas efektif koneksi ISP. *WireGuard* menunjukkan kinerja yang lebih baik pada skenario *Normal* (*upload*) dengan mempertahankan *packet*

loss lebih rendah dibandingkan ketiga protokol pembanding, sedangkan pada skenario *Reverse* (*download*) *OpenVPN* dan L2TP/IPsec menghasilkan *packet loss* yang lebih rendah dibandingkan *WireGuard*. Sementara itu, nilai *jitter* seluruh protokol tetap berada pada kisaran yang relatif rendah sehingga tidak menunjukkan perbedaan yang berarti. Kondisi tersebut menunjukkan bahwa pada lingkungan SOHO dengan *bandwidth* asimetris, kemampuan protokol VPN dalam mempertahankan keberhasilan pengiriman paket menjadi faktor yang lebih menentukan dibandingkan perubahan nilai *jitter*. Meskipun demikian, hasil pengujian juga menunjukkan bahwa karakteristik masing-masing protokol dapat berubah pada kondisi jaringan yang berbeda, sehingga interpretasi performa perlu mempertimbangkan keseluruhan skenario pengujian, baik pada kondisi siang maupun malam.

Hasil penelitian ini dapat dibandingkan dengan penelitian Muhajir dan Utami (2026) yang juga melakukan evaluasi performa *WireGuard* dan *OpenVPN* menggunakan pengujian lalu lintas jaringan. Pada penelitian tersebut, *WireGuard* menunjukkan performa yang kompetitif, tetapi terdapat kondisi tertentu ketika *OpenVPN* menghasilkan performa yang lebih baik pada parameter dan skenario tertentu. Hasil penelitian ini menunjukkan pola yang serupa, yaitu keunggulan *WireGuard* tidak bersifat seragam pada seluruh skenario. Pada UDP 10 Mbps arah *Normal* siang hari, *WireGuard* menghasilkan *packet loss* sebesar 5,95%, lebih rendah dibandingkan L2TP/IPsec sebesar 10,05%, *OpenVPN* sebesar 10,20%, dan SSTP sebesar 25,65%. Namun, pada arah *Reverse* siang hari, *OpenVPN* dan L2TP/IPsec menghasilkan *packet loss* lebih rendah, masing-masing sebesar 0,093% dan 1,053%, dibandingkan *WireGuard* sebesar 2,182%. Perbedaan tersebut menunjukkan bahwa performa protokol VPN dapat berubah berdasarkan arah trafik dan karakteristik beban jaringan.

4.3.4 Utilisasi CPU

Utilisasi CPU digunakan untuk mengevaluasi besarnya beban komputasi yang ditimbulkan oleh setiap protokol VPN selama proses enkripsi dan dekripsi paket. Parameter ini menjadi penting karena kedua router pada penelitian ini menggunakan prosesor MIPSBE *single-core* tanpa dukungan *hardware*

cryptographic acceleration, sehingga seluruh proses kriptografi dijalankan menggunakan sumber daya CPU. Pengukuran dilakukan pada seluruh skenario pengujian, yaitu TCP *Single Stream*, TCP *Multi Stream*, UDP 5 Mbps, dan UDP 10 Mbps, baik pada arah *Normal* (enkripsi) maupun *Reverse* (dekripsi).

Rangkuman hasil pengukuran utilisasi CPU pada pengujian siang dan malam disajikan pada Tabel 4.12 dan Tabel 4.13. Untuk memperlihatkan hubungan antara utilisasi CPU dan *throughput*, hasil pengujian divisualisasikan menggunakan *scatter plot* pada Gambar 4.20. Grafik tersebut menggunakan nilai TCP *Multi Stream* arah *Reverse* pada pengujian siang sebagai representasi karena skenario tersebut menghasilkan *throughput* TCP tertinggi sekaligus beban dekripsi terbesar pada masing-masing protokol.

Tabel 4.12 Utilisasi CPU siang

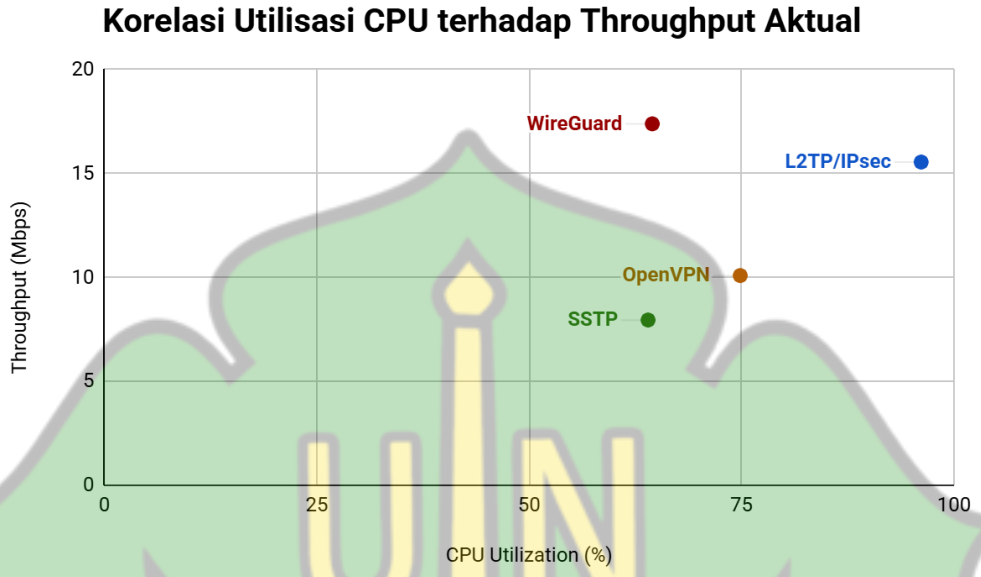
Protocol VPN	**Enkripsi (*Normal/Upload*) %**				**Dekripsi (*Reverse/Download*) %**			
	TCP Single	**TCP Multi**	**UDP 5 M**	**UDP 10 M**	**TCP Single**	**TCP Multi**	**UDP 5 M**	**UDP 10 M**
WireGuard	36,4	40,65	11	20,25	64,75	64,5	14,5	26
L2TP/IPsec	68,2	74,1	29,15	50,2	94,85	96,15	36,95	63,1
OpenVPN	73,3	76,35	29,9	53,45	72,45	74,85	39,2	74,2
SSTP	56,35	63,4	35,7	48	60,55	64	38,75	56,7

Tabel 4.13 Utilisasi CPU malam

Protocol VPN	**Enkripsi (*Normal/Upload*) %**				**Dekripsi (*Reverse/Download*) %**			
	TCP Single	**TCP Multi**	**UDP 5 M**	**UDP 10 M**	**TCP Single**	**TCP Multi**	**UDP 5 M**	**UDP 10 M**
WireGuard	35,65	42,4	11	19,85	63,4	68,2	14,5	27,3
L2TP/IPsec	60,9	68,05	25,35	45,4	88,55	94,6	32,15	58,8
OpenVPN	77,4	81,05	37,25	61,5	73,35	78,45	46,55	81,15
SSTP	58,25	60,75	35,75	48,1	59,3	61,05	37,65	57,05

Untuk memperlihatkan korelasi antara utilisasi CPU dan *throughput* pada kondisi pembebanan (*worst-case scenario*), hasil pengujian TCP *Multi Stream* arah *Reverse* divisualisasikan menggunakan Scatter Plot pada gambar 4.19.

Skenario ini dipilih sebagai representasi karena menghasilkan *throughput* dan beban dekripsi tertinggi dibandingkan pada skenario lain sebagaimana dapat dilihat pada Tabel 4.10 dan 4.11.



Gambar 4.20 Korelasi Utilisasi CPU dengan Throughput

Berdasarkan Tabel 4.12 dan Tabel 4.13, seluruh protokol menunjukkan pola yang serupa, yaitu utilisasi CPU pada arah *Reverse* (dekripsi) lebih tinggi dibandingkan arah *Normal* (enkripsi). Pola tersebut terlihat hampir pada seluruh skenario pengujian. Kondisi ini berkaitan dengan *throughput* yang diproses router pada arah *Reverse*, di mana kapasitas *download* ISP lebih besar dibandingkan kapasitas *upload*. Akibatnya, jumlah paket yang harus diproses pada tahap dekripsi menjadi lebih besar sehingga utilisasi CPU juga meningkat.

Di antara keempat protokol yang diuji, *WireGuard* secara konsisten menunjukkan utilisasi CPU rata-rata terendah pada seluruh skenario pengujian. Pada pengujian TCP *Single Stream*, utilisasi CPU *WireGuard* mencapai 36,4% saat proses enkripsi dan 64,75% saat proses dekripsi pada pengujian siang. Hasil yang relatif serupa juga diperoleh pada pengujian malam, yaitu 35,65% dan 63,4%. Nilai tersebut lebih rendah dibandingkan L2TP/IPsec maupun *OpenVPN*

yang memerlukan utilisasi CPU lebih tinggi untuk menjalankan proses transmisi data.

Pola yang sama juga terlihat pada pengujian TCP *Multi Stream*. *WireGuard* menggunakan CPU sebesar 40,65% pada arah *Normal* dan 64,5% pada arah *Reverse* saat pengujian siang, sedangkan pada pengujian malam tercatat 42,4% dan 68,2%. Sebaliknya, L2TP/IPsec mencapai utilisasi CPU hingga 96,15%, sedangkan *OpenVPN* mencapai 81,05% pada skenario yang sama. Meskipun *throughput* yang dihasilkan kedua protokol tersebut lebih rendah dibandingkan *WireGuard*, kebutuhan sumber daya CPU yang digunakan justru lebih tinggi.

Perbedaan karakteristik antarprotokol juga terlihat pada pengujian UDP. Pada pembebanan UDP 5 Mbps, utilisasi CPU seluruh protokol masih relatif rendah karena laju injeksi paket berada di bawah kapasitas efektif koneksi ISP. Ketika pembebanan ditingkatkan menjadi UDP 10 Mbps, utilisasi CPU seluruh protokol meningkat. Namun demikian, *WireGuard* tetap mempertahankan utilisasi CPU yang lebih rendah dibandingkan protokol pembanding, sedangkan *OpenVPN* dan L2TP/IPsec menunjukkan peningkatan utilisasi CPU yang lebih besar. Kondisi tersebut menunjukkan bahwa peningkatan beban lalu lintas memberikan dampak yang berbeda terhadap kebutuhan komputasi masing-masing protokol.

Hubungan antara *throughput* dan utilisasi CPU terlihat lebih jelas pada Gambar 4.20. *WireGuard* berada pada posisi dengan *throughput* tertinggi (17,39 Mbps) namun hanya memerlukan utilisasi CPU sekitar 64,5%. Sebaliknya, L2TP/IPsec menghasilkan *throughput* yang lebih rendah (15,55 Mbps) tetapi membutuhkan utilisasi CPU tertinggi, yaitu sekitar 96,15%. *OpenVPN* juga memperlihatkan kecenderungan serupa dengan utilisasi CPU sekitar 74,85% untuk menghasilkan *throughput* 10,09 Mbps, sedangkan SSTP menghasilkan *throughput* paling rendah (7,95 Mbps) dengan utilisasi CPU sekitar 64%. Pola tersebut menunjukkan bahwa efisiensi suatu protokol tidak hanya ditentukan oleh besarnya *throughput* atau kecilnya utilisasi CPU secara terpisah, tetapi oleh keseimbangan antara keduanya. Berdasarkan hubungan tersebut, *WireGuard*

menunjukkan efisiensi pemrosesan yang paling baik pada perangkat MikroTik MIPSBE *single-core* yang digunakan dalam penelitian ini.

Hasil tersebut sejalan dengan karakteristik implementasi *WireGuard* yang telah dijelaskan pada Bab II. *WireGuard* diimplementasikan di dalam *kernel-space* dan menggunakan algoritma kriptografi ChaCha20-Poly1305, sehingga proses enkripsi dan dekripsi dapat dilakukan dengan *overhead* pemrosesan yang lebih rendah dibandingkan protokol yang memerlukan mekanisme negosiasi koneksi dan pemrosesan paket yang lebih kompleks. Temuan ini juga konsisten dengan konsep yang dikemukakan oleh (Donenfeld, 2017), bahwa integrasi *WireGuard* ke dalam kernel memungkinkan proses transmisi paket berlangsung dengan lebih efisien dibandingkan implementasi VPN berbasis *user-space*.

Berdasarkan seluruh skenario pengujian, *WireGuard* mampu mempertahankan keseimbangan antara *throughput* yang tinggi dan utilisasi CPU yang relatif rendah dibandingkan ketiga protokol pembanding. Temuan ini menunjukkan bahwa *WireGuard* merupakan protokol yang paling efisien dalam memanfaatkan sumber daya prosesor pada router MikroTik MIPSBE *single-core* yang digunakan dalam penelitian ini. Selanjutnya, untuk melengkapi analisis efisiensi sumber daya perangkat, dilakukan evaluasi terhadap utilisasi RAM pada subbab berikutnya.

4.3.5 Utilisasi RAM

Selain utilisasi CPU, penggunaan memori utama (RAM) juga diamati untuk mengevaluasi kebutuhan sumber daya masing-masing protokol VPN selama proses transmisi data. Berbeda dengan CPU yang merepresentasikan beban komputasi secara langsung, penggunaan RAM pada penelitian ini digunakan untuk mengidentifikasi *memory footprint* masing-masing protokol, yaitu besarnya memori yang dialokasikan selama layanan VPN aktif pada router MikroTik RB951Ui-2nD yang memiliki kapasitas RAM sebesar 64 MB.

Rangkuman hasil pengukuran penggunaan RAM pada pengujian siang dan malam disajikan pada Tabel 4.14 dan Tabel 4.15, sedangkan perbandingan rata-rata penggunaan RAM antarprotokol divisualisasikan pada Gambar 4.21.

Tabel 4.14 Utilisasi RAM Usage Siang

Protocol VPN	**Enkripsi (*Normal/Upload*) MB**				**Dekripsi (*Reverse/Download*) MB**			
	TCP Single	**TCP Multi**	**UDP 5 M**	**UDP 10 M**	**TCP Single**	**TCP Multi**	**UDP 5 M**	**UDP 10 M**
WireGuard	37	37	37	37	37	37	37	37
L2TP/IPsec	40	40	40,85	41	40	40,45	40	41
OpenVPN	41	40,35	41	41	41	40,1	41	41
SSTP	43	43	43	43	43	43	43	43,25

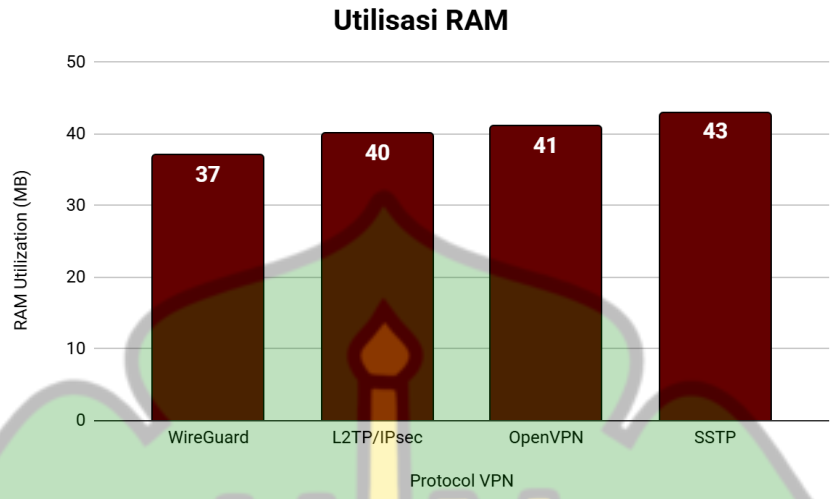
Tabel 4.15 Utilisasi RAM Usage malam

Protocol VPN	**Enkripsi (*Normal/Upload*) MB**				**Dekripsi (*Reverse/Download*) MB**			
	TCP Single	**TCP Multi**	**UDP 5 M**	**UDP 10 M**	**TCP Single**	**TCP Multi**	**UDP 5 M**	**UDP 10 M**
WireGuard	37	37	37	37	37	37	37	37
L2TP/IPsec	39	39	39	39,05	39	39	39	39
OpenVPN	41,9	42	42	42	42	42	42	42
SSTP	44	44	44	44	44	44	44	44

Selain utilisasi CPU, penggunaan memori utama (RAM) juga diamati untuk mengevaluasi kebutuhan sumber daya masing-masing protokol VPN selama proses transmisi data. Berbeda dengan CPU yang merepresentasikan beban komputasi secara langsung, penggunaan RAM pada penelitian ini digunakan untuk mengidentifikasi *memory footprint* masing-masing protokol, yaitu besarnya memori yang dialokasikan selama layanan VPN aktif pada router MikroTik RB951Ui-2nD yang memiliki kapasitas RAM sebesar 64 MB.

Berdasarkan Tabel 4.14 dan Tabel 4.15, penggunaan RAM pada seluruh protokol relatif stabil selama seluruh skenario pengujian. Berbeda dengan utilisasi CPU yang meningkat seiring bertambahnya beban lalu lintas, perubahan penggunaan RAM pada masing-masing protokol sangat kecil. Kondisi tersebut

menunjukkan bahwa peningkatan lalu lintas jaringan lebih banyak memengaruhi beban komputasi CPU dibandingkan kebutuhan alokasi memori.



Gambar 4.21 Utilisasi RAM

WireGuard secara konsisten menunjukkan *memory footprint* paling kecil pada perangkat yang digunakan dalam penelitian ini. Pada seluruh skenario pengujian, baik TCP maupun UDP, serta pada arah *Normal* (enkripsi) maupun *Reverse* (dekripsi), penggunaan RAM tetap berada di sekitar 37 MB tanpa perubahan yang berarti. Kondisi tersebut menunjukkan bahwa peningkatan lalu lintas jaringan tidak memberikan pengaruh yang nyata terhadap kebutuhan memori selama proses pengujian.

L2TP/IPsec menunjukkan kebutuhan memori yang sedikit lebih tinggi dibandingkan *WireGuard*, yaitu berada pada kisaran 39 MB hingga 41 MB. *OpenVPN* menggunakan sekitar 41 MB hingga 42 MB, sedangkan SSTP memiliki *memory footprint* terbesar, yaitu sekitar 43 MB hingga 44 MB selama seluruh pengujian. Meskipun terdapat perbedaan kebutuhan memori antarprotokol, seluruh nilai tersebut masih berada di bawah kapasitas RAM perangkat sebesar 64 MB, sehingga selama penelitian tidak ditemukan indikasi keterbatasan memori (*memory exhaustion*) maupun gangguan operasional akibat kekurangan RAM.

Hasil tersebut menunjukkan bahwa penggunaan RAM lebih banyak ditentukan oleh karakteristik implementasi masing-masing protokol dibandingkan besarnya beban lalu lintas yang diproses. Setelah layanan VPN aktif dan kebutuhan memori dasar (*memory footprint*) dialokasikan, peningkatan jumlah paket yang ditransmisikan tidak menyebabkan penambahan kebutuhan memori yang berarti. Sebaliknya, peningkatan beban lalu lintas lebih banyak meningkatkan aktivitas komputasi pada CPU sebagaimana telah dibahas pada Subbab 4.3.4.

Secara keseluruhan, penggunaan RAM seluruh protokol relatif stabil pada seluruh skenario pengujian. *WireGuard* menunjukkan *memory footprint* paling kecil dibandingkan L2TP/IPsec, *OpenVPN*, dan SSTP, namun selisih penggunaan RAM antarprotokol relatif kecil jika dibandingkan dengan kapasitas total memori perangkat. Temuan ini menunjukkan bahwa pada lingkungan penelitian yang menggunakan router MikroTik MIPSBE *single-core*, RAM bukan merupakan faktor pembatas utama performa VPN. Sebaliknya, efisiensi implementasi protokol lebih banyak dipengaruhi oleh kebutuhan komputasi CPU daripada penggunaan memori.

4.3.6 Sintesis Hasil Evaluasi

Hasil evaluasi pada Subbab 4.3.1 hingga Subbab 4.3.5 menunjukkan bahwa karakteristik performa masing-masing protokol berbeda bergantung pada parameter, arah trafik, dan kondisi waktu pengujian. Oleh karena itu, sintesis hasil tidak dilakukan dengan menetapkan satu protokol sebagai pemenang mutlak, tetapi dengan membandingkan statistik representatif dari seluruh iterasi pengujian. Pendekatan tersebut digunakan untuk mengidentifikasi hubungan antara kapasitas transmisi, kualitas layanan, dan penggunaan sumber daya perangkat sebagai dasar analisis trade-off antarprotokol.

Pada kondisi *load testing* penelitian ini, keterbatasan *throughput* pada sisi *Branch* juga dipengaruhi oleh kapasitas *bandwidth* ISP asimetris, sementara utilisasi CPU *WireGuard* pada *Router Branch* yang dipantau masih berada pada tingkat operasional yang memadai dan penggunaan RAM relatif stabil. Temuan

tersebut menunjukkan bahwa *WireGuard* mampu beroperasi pada perangkat MikroTik kelas SOHO yang digunakan dalam ruang lingkup pengujian ini, tetapi tidak dimaksudkan untuk menggeneralisasi kondisi *Router HQ* yang tidak dipantau secara langsung maupun lingkungan jaringan di luar skenario penelitian.

Temuan tersebut melengkapi *research gap* yang telah diidentifikasi pada Bab I dan Bab II dengan memberikan bukti empiris mengenai karakteristik *WireGuard* pada router MikroTik MIPSBE *single-core* dalam kondisi ISP asimetris. Selain parameter QoS, penelitian ini mengaitkan performa transmisi dengan utilisasi CPU dan *memory footprint* perangkat, sehingga memberikan gambaran mengenai trade-off antara kualitas transmisi dan efisiensi sumber daya pada lingkungan SOHO yang menjadi ruang lingkup penelitian.

Sintesis hasil dilakukan dengan membandingkan nilai statistik representatif dari seluruh iterasi pengujian, bukan dengan menetapkan satu protokol sebagai pemenang mutlak. Nilai *mean* digunakan untuk TCP *throughput*, utilisasi CPU, utilisasi RAM, dan UDP *packet loss*, sedangkan *median* digunakan untuk *latency* karena lebih tahan terhadap pengaruh pencilan (*outlier*) dan *latency spikes*. Pendekatan tersebut digunakan untuk mengidentifikasi *trade-off* performa dan efisiensi sumber daya antarprotokol.

Dalam penyusunan matriks trade-off, mean digunakan sebagai statistik representatif untuk parameter TCP *throughput*, utilisasi CPU, utilisasi RAM, dan UDP *packet loss*. Penggunaan mean bertujuan menggambarkan kecenderungan hasil pengukuran pada seluruh iterasi pengujian. Sementara itu, *latency* direpresentasikan menggunakan median karena data *latency* pada jaringan publik lebih rentan terhadap pencilan dan *latency* spike. Dengan demikian, matriks trade-off menggunakan mean untuk *throughput*, CPU, RAM, dan *packet loss*, serta median untuk *latency*. Nilai yang ditampilkan merupakan statistik representatif dari iterasi pengujian pada masing-masing kondisi.

Tabel 4.16 Matriks *Trade-off* Performa Protokol VPN Kondisi Siang

Parameter	Arah	*WireGuard*	L2TP/IPsec	*OpenVPN*	SSTP
TCP *Throughput (Mean, Mbps)*	*Normal*	9,26	8,96	8,86	7,33
TCP *Throughput (Mean, Mbps)*	*Reverse*	17,86	15,54	9,79	7,34
CPU (Mean, %)	*Normal*	36,40	68,20	73,30	56,35
CPU (Mean, %)	*Reverse*	64,75	94,85	72,45	60,55
RAM (Mean, MB)	*Normal*	37	40	41	43
RAM (Mean, MB)	*Reverse*	37	40	41	43
Latency (Median, ms)	ICMP	7,00	5,00	5,00	5,50
UDP *Packet Loss (Mean, %)*	*Normal*	5,95	10,05	10,2	25,65
UDP *Packet Loss (Mean, %)*	*Reverse*	2,182	1,053	0,093	13,15

Berdasarkan Tabel 4.16, *WireGuard* menunjukkan kecenderungan performa yang menguntungkan pada *throughput* TCP dan efisiensi penggunaan sumber daya pada kondisi siang. Throughput arah *Normal* dan *Reverse* masing-masing mencapai 9,26 Mbps dan 17,86 Mbps, sedangkan utilisasi CPU masing-masing sebesar 36,40% dan 64,75%. Penggunaan RAM juga tercatat sebesar 37 MB pada kedua arah. Kombinasi tersebut menunjukkan bahwa *WireGuard* mampu mempertahankan kapasitas transmisi yang tinggi dengan kebutuhan sumber daya yang relatif rendah pada kondisi pengujian tersebut.

Namun, keunggulan tersebut tidak berlaku pada seluruh parameter. Median *latency* sebesar 7,00 ms masih lebih tinggi dibandingkan L2TP/IPsec dan *OpenVPN* yang masing-masing sebesar 5,00 ms. Pada UDP 10 Mbps arah *Reverse*, *OpenVPN* menghasilkan *packet loss* terendah sebesar 0,093%,

L2TP/IPsec sebesar 1,053%, sedangkan *WireGuard* sebesar 2,182%. Dengan demikian, keunggulan *WireGuard* pada *throughput* dan efisiensi sumber daya disertai *trade-off* pada beberapa parameter QoS.

Tabel 4.17 Matriks *Trade-off* Performa Protokol VPN Kondisi malam

Parameter	**Arah**	***WireGuard***	**L2TP/IPsec**	***OpenVPN***	**SSTP**
TCP *Throughput (Mean, Mbps)*	*Normal*	9,24	8,85	8,63	7,55
TCP *Throughput (Mean, Mbps)*	*Reverse*	17,42	15,26	8,82	7,22
CPU (Mean, %)	*Normal*	35,65	60,9	77,40	58,25
CPU (Mean, %)	*Reverse*	63,4	88,55	73,35	59,3
RAM (Mean, MB)	*Normal*	37	39	41,9	44
RAM (Mean, MB)	*Reverse*	37	39	42	44
Latency (Median, ms)	ICMP	5,00	6,00	5,00	7,00
UDP *Packet Loss (Mean, %)*	*Normal*	5,335	10,05	10,50	25,75
UDP *Packet Loss (Mean, %)*	*Reverse*	0,056	0,037	0,074	13,15

Berdasarkan Tabel 4.16, *WireGuard* mempertahankan karakteristik *throughput* dan efisiensi sumber daya yang relatif baik. Throughput TCP mencapai 9,24 Mbps pada arah *Normal* dan 17,42 Mbps pada arah *Reverse*, dengan utilisasi CPU masing-masing sebesar 35,65% dan 63,40%. Penggunaan RAM tetap sebesar 37 MB pada kedua arah. Hasil tersebut menunjukkan bahwa karakteristik efisiensi sumber daya *WireGuard* juga terlihat pada kondisi malam

Namun, pada UDP 10 Mbps arah *Reverse*, L2TP/IPsec menghasilkan *packet loss* terendah sebesar 0,037%, diikuti *WireGuard* sebesar 0,056% dan *OpenVPN* sebesar 0,074%. Hasil tersebut kembali menunjukkan bahwa tidak

terdapat satu protokol yang unggul pada seluruh parameter. Dalam kondisi pengujian malam, perbedaan *packet loss* antarprotokol menjadi lebih kecil dibandingkan kondisi siang, sehingga karakteristik performa juga dipengaruhi oleh kondisi jaringan pada saat pengujian.

Secara keseluruhan, hasil sintesis menunjukkan bahwa *WireGuard* memiliki keunggulan terutama pada kombinasi *throughput* TCP dan efisiensi penggunaan CPU serta RAM, sedangkan L2TP/IPsec dan *OpenVPN* menunjukkan keunggulan pada parameter QoS tertentu dalam skenario tertentu. Oleh karena itu, hasil evaluasi menunjukkan bahwa tidak terdapat satu protokol yang memberikan hasil terbaik pada seluruh parameter dan kondisi pengujian, melainkan adanya *trade-off* antara kapasitas transmisi, kualitas layanan, dan efisiensi sumber daya. Pada perangkat MikroTik MIPSBE *single-core* dalam kondisi ISP asimetris yang diuji, *WireGuard* menunjukkan karakteristik yang menguntungkan ketika efisiensi sumber daya dan *throughput* TCP menjadi prioritas, dengan tetap mempertimbangkan karakteristik *packet loss* UDP pada kondisi dan arah trafik tertentu.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. *WireGuard* berhasil diimplementasikan sebagai VPN *Site-to-Site* pada *router* MikroTik berarsitektur MIPSBE *single-core* dalam kondisi jaringan ISP asimetris dan beroperasi sesuai dengan rancangan penelitian. Berdasarkan hasil pengujian komparatif, *WireGuard* menunjukkan karakteristik performa yang kompetitif, terutama pada *throughput* TCP, dengan nilai yang lebih tinggi dibandingkan protokol pembanding pada sebagian besar skenario pengujian. Pada parameter *latency*, *WireGuard* menunjukkan nilai median sebesar 7,00 ms pada siang hari dan 5,00 ms pada malam hari. Namun, keunggulan tersebut tidak bersifat mutlak pada seluruh parameter dan skenario. Pada pengujian UDP 10 Mbps arah *Reverse* siang hari, *packet loss WireGuard* sebesar 2,182% lebih tinggi dibandingkan *OpenVPN* sebesar 0,093% dan L2TP/IPsec sebesar 1,053%. Dengan demikian, karakteristik performa *WireGuard* bergantung pada jenis trafik, arah transmisi, dan kondisi jaringan yang digunakan dalam pengujian.
2. Pada skenario *load testing*, *WireGuard* menunjukkan karakteristik efisiensi sumber daya yang lebih baik pada sebagian besar kondisi pengujian, terutama dari sisi utilisasi CPU dan penggunaan RAM. Penggunaan RAM *WireGuard* tercatat stabil sebesar 37 MB, sedangkan pada pengujian TCP *Multi Stream Reverse* siang hari *WireGuard* menghasilkan *throughput* sebesar 17,39 Mbps dengan utilisasi CPU 64,5%, sementara L2TP/IPsec mencapai utilisasi CPU 96,15% dengan *throughput* yang lebih rendah. Di sisi lain, *WireGuard* tidak menjadi protokol dengan hasil terbaik pada seluruh parameter, sebagaimana ditunjukkan oleh *packet loss* UDP arah *Reverse* yang lebih rendah pada *OpenVPN* dan L2TP/IPsec. Oleh karena

itu, terdapat *trade-off* antara efisiensi sumber daya, kapasitas transmisi, dan keandalan trafik pada masing-masing protokol. Dalam ruang lingkup penelitian ini, karakteristik tersebut menunjukkan bahwa *WireGuard* layak diterapkan sebagai VPN *Site-to-Site* pada perangkat MikroTik kelas SOHO, khususnya ketika efisiensi sumber daya dan *throughput* TCP menjadi prioritas, dengan tetap mempertimbangkan *trade-off* pada keandalan trafik UDP.

5.2 Saran

Berdasarkan hasil penelitian dan ruang lingkup yang telah ditetapkan, terdapat beberapa aspek yang masih dapat dikembangkan pada penelitian selanjutnya.

1. Penelitian selanjutnya dapat mengevaluasi implementasi *WireGuard* pada perangkat MikroTik dengan arsitektur prosesor yang berbeda, seperti ARM atau ARM64, termasuk perangkat yang dilengkapi *hardware cryptographic acceleration*. Pengujian tersebut dapat digunakan untuk mengetahui sejauh mana karakteristik perangkat keras memengaruhi performa dan efisiensi penggunaan sumber daya *WireGuard* dibandingkan dengan protokol VPN lainnya.
2. Penelitian selanjutnya dapat memperluas skenario pengujian pada kondisi jaringan yang berbeda, seperti koneksi ISP simetris, kapasitas *bandwidth* yang lebih tinggi, serta jumlah pengguna atau beban trafik yang lebih besar. Pengembangan skenario tersebut dapat memberikan gambaran mengenai karakteristik performa *WireGuard* pada kondisi jaringan yang lebih beragam dibandingkan dengan lingkungan penelitian ini.
3. Penelitian ini berfokus pada evaluasi performa melalui skenario *load testing* dalam periode pengujian yang terbatas. Oleh karena itu, penelitian selanjutnya dapat mengembangkan evaluasi jangka panjang (*long-term performance evaluation*) dengan mengamati kestabilan koneksi, perubahan utilisasi CPU dan RAM selama operasi berkelanjutan, serta kemungkinan terjadinya perubahan atau degradasi performa selama periode penggunaan yang lebih panjang.

DAFTAR PUSTAKA

Anyam, J., Singh, R. R., Larijani, H., & Philip, A. (2025). Empirical Performance *Analysis* of *WireGuard* vs. *OpenVPN* in Cloud and Virtualised Environments Under Simulated Network Conditions. *Computers*, *14*(8), 326. https://doi.org/10.3390/computers14080326

Borman, D. (2012). *TCP Options and Maximum Segment Size (MSS)* (Request for Comments RFC 6691). Internet Engineering Task Force. https://doi.org/10.17487/RFC6691

Cai, P., & Karsten, M. (2023). Kernel vs. User-Level Networking: Don't Throw Out the Stack with the Interrupts. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *7*(3), 1–25. https://doi.org/10.1145/3626780

Deering, S. E., & Mogul, J. (1990). *Path MTU discovery* (Request for Comments RFC 1191). Internet Engineering Task Force. https://doi.org/10.17487/RFC1191

Demirdelen, T., & Kirmizi, S. (2025). Performance Evaluation of *WireGuard* and IPSec Protocols in Various Network Configurations. *Osmaniye Korkut Ata University Journal of The Institute of Science and Technology*, *8*(3), 1353–1369. https://doi.org/https://doi.org/10.47495/okufbed.1660352

Donenfeld, J. A. (2017). *WireGuard*: Next Generation Kernel Network Tunnel. *Proceedings 2017 Network and Distributed System Security Symposium*. Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2017.23160

Dugan, J., Elliott, S., Mah, B. A., Poskanzer, J., & Prabhu, K. (2025). *iPerf3: A TCP, UDP, and SCTP network bandwidth measurement tool* (Versi 3.20) [Perangkat lunak]. Energy Sciences Network (ESnet). https://software.esnet.org/iperf/

Gleeson, B., Lin, A., Heinanen, J., Armitage, G., & Malis, A. (2000). A Framework for IP Based Virtual Private Networks (RFC 2764). Internet Engineering Task Force (IETF). https://doi.org/10.17487/RFC2764

Hermawan, R., & Saputra, Y. M. (2025). 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 (JISE)*, 6(1), 1–12.

Høiland-Jørgensen, T., Brouer, J. D., Borkmann, D., Fastabend, J., Herbert, T., Ahern, D., & Miller, D. (2018). The eXpress data path: Fast programmable packet processing in the operating system kernel. *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, 54–66. https://doi.org/10.1145/3281411.3281443

Honda, O., Ohsaki, H., Imase, M., Ishizuka, M., & Murayama, J. (2005). *Understanding TCP over TCP: Effects of TCP tunneling on end-to-end*

throughput and latency (M. Atiquzzaman & S. I. Balandin, Ed.; hlm. 60110H). https://doi.org/10.1117/12.630496

Hussein, S. G., & Rehman, S. M. F. U. (2025). Protocol Efficiency and Resource Utilization in VPN Technologies: A Comparative *Analysis* of *OpenVPN* and *WireGuard*. *Journal of Techniques*, *7*(4), 37–42. https://doi.org/10.51173/jt.v7i4.2768

Ikhwandi, M. I., Azinar, A. W., & Sumarno. (2025). Implementasi Wireguard Sebagai Koneksi Menggunakan Routing Mikrotik. *SMATIKA JURNAL*, *15*(02), 292–301. https://doi.org/10.32664/smatika.v15i02.1898

International Telecommunication Union. (2023). *Internet protocol data communication service – IP packet transfer and availability performance parameters (Recommendation ITU-T Y.1540)*. ITU-T. https://www.itu.int/rec/T-REC-Y.1540-202303-I!Amd2/en

Jumakhan, H., & Mirzaeinia, A. (2024). *Wireguard: An Efficient Solution for Securing IoT Device Connectivity* (arXiv:2402.02093). arXiv. https://doi.org/10.48550/arXiv.2402.02093

Kurose, J. F., & Ross, K. W. (2022). *Computer Networking: A Top-Down Approach* (8th Edition (Global Edition)). Pearson Education Limited.

Luthfi, A., & Sinduningrum, E. (2025). Desain Pengamanan Jaringan Nirkabel Menggunakan Radius Server di Fakultas Teknologi Industri dan Informatika Uhamka. *Jurnal Pendidikan dan Teknologi Indonesia*, *5*(12), 3602–3609. https://doi.org/10.52436/1.jpti.1192

MikroTik. (t.t.-a). *Speed Test—RouterOS - MikroTik Documentation*. Diambil 2 Juli 2026, dari https://help.mikrotik.com/docs/spaces/ROS/pages/8978571/Speed+Test

MikroTik. (t.t.-b). *WireGuard—RouterOS - MikroTik Documentation*. Diambil 25 Maret 2026, dari https://help.mikrotik.com/docs/spaces/ROS/pages/69664792/*WireGuard*

MikroTik. (2017). *hAP (RB951Ui-2nD) specifications* (No. 170908). MikroTikls SIA. https://mikrotik.com/product/RB951Ui-2nD#product_specification

Muhajir, F., & Utami, E. (2026). Analisis Performa *WireGuard* dan *OpenVPN* pada VPN Perbankan Berbasis MikroTik menggunakan ICMP, *iPerf3*, dan Mann–Whitney. *JURIKOM (Jurnal Riset Komputer)*, *13*(1), 71–84. https://doi.org/10.30865/jurikom.v13i1.9438

Patterson, D. A., & Hennessy, J. L. (2021). *Computer organization and design MIPS edition: The hardware/software interface* (6th Edition). Morgan Kaufmann (Elsevier).

Rahman, I. K., & Harnaningrum, L. N. (2024). Analisa Quality of Service (QoS) Pada Jaringan L2TP IPSec Dan Wireguard VPN untuk mengamankan VoIP. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, *7*(1), 10–20. https://doi.org/10.31598/jurnalresistor.v7i1.1553

Santoso, B., Sani, A., Husain, T., & Hendri, N. (2021). VPN SITE TO SITE IMPLEMENTATION USING PROTOCOL L2TP AND IPSEC. *TEKNOKOM*, *4*(1), 30–36. https://doi.org/10.31943/teknokom.v4i1.59

Stallings, W. (2020). *Cryptography and Network Security: Principles and Practice* (8th Edition (Global Edition)). Pearson Education Limited.