BAB 3
METODOLOGI PENELITIAN

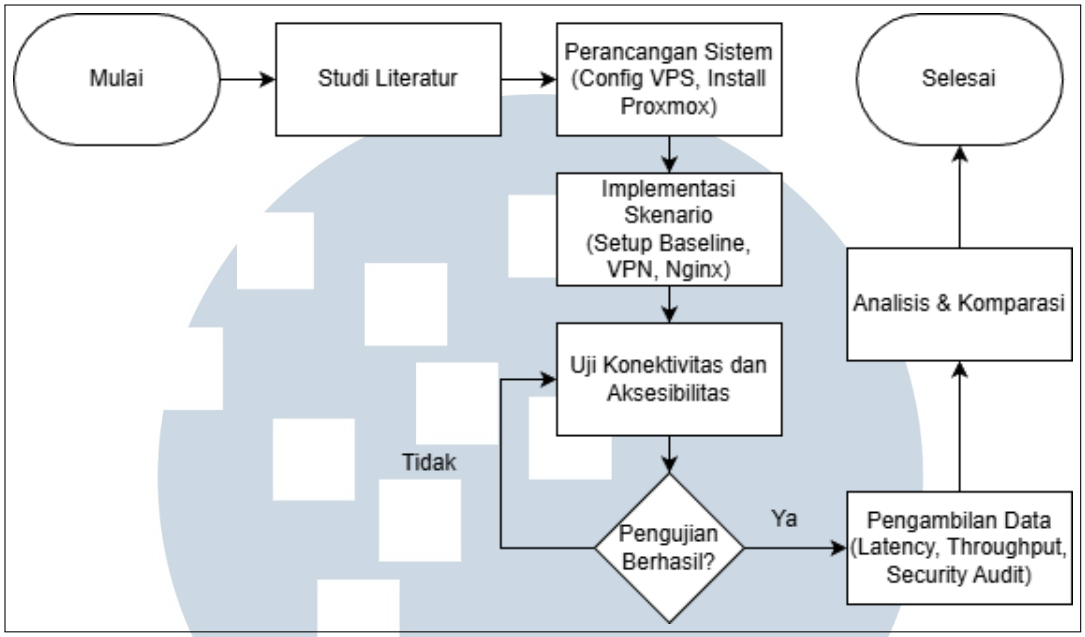
Bab ini menjelaskan metodologi yang digunakan untuk menganalisis kinerja dan keamanan akses jarak jauh (*remote access*) pada NAS. Ruang lingkup pembahasan disusun secara sistematis mulai dari desain penelitian, rancangan arsitektur dan infrastruktur, tahapan implementasi sistem, skenario pengujian, hingga prosedur pengambilan dan analisis data.

3.1 Desain Penelitian

Penelitian ini menggunakan metode eksperimental komparatif dengan pendekatan kuantitatif. Desain penelitian difokuskan untuk mengukur dan membandingkan dampak implementasi keamanan pada dua lapisan berbeda yaitu enkripsi *Network-Layer* dan *Application-Layer* terhadap kinerja sistem dalam lingkungan jaringan yang dibatasi oleh *Carrier-Grade NAT* (CGNAT).

Guna memastikan validitas, objektivitas, dan keterulangan (*reproducibility*) data, seluruh proses akuisisi data performa diotomatisasi menggunakan kerangka kerja (*framework*) berbasis skrip *Bash* yang dikembangkan secara khusus. Mekanisme ini bertujuan untuk meminimalisasi intervensi manual serta mencegah kesalahan manusia (*human error*) dalam pengumpulan total 1.420 titik data pengukuran.

Sementara itu, evaluasi keamanan dilakukan dengan pendekatan *Blue Teaming* yang menitikberatkan pada validasi pertahanan (*security validation*) dan audit konfigurasi sistem, alih-alih melakukan simulasi serangan aktif (*red teaming*). Tahapan metodologi penelitian secara menyeluruh divisualisasikan pada Gambar 3.1.



Gambar 3.1. Alur metodologi penelitian

Setelah menetapkan kerangka kerja metodologis, langkah selanjutnya adalah mendefinisikan spesifikasi teknis dan topologi jaringan yang menjadi objek penelitian. Bagian berikut akan menjabarkan rancangan arsitektur sistem yang digunakan untuk mengatasi kendala konektivitas pada jaringan privat.

3.2 Studi Literatur

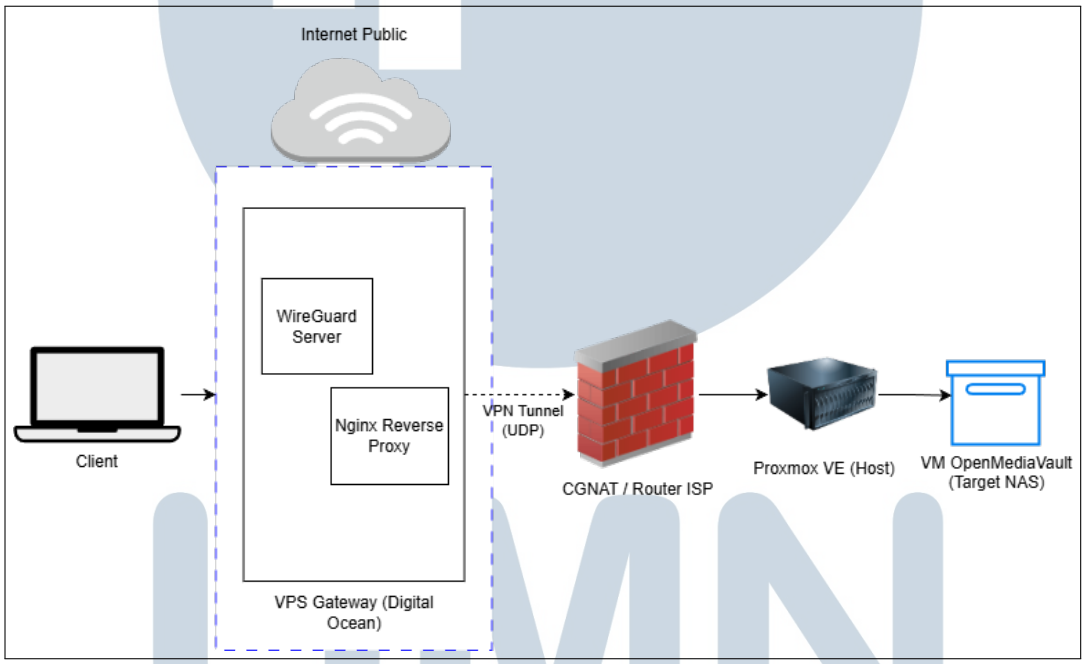
Tahapan ini dilakukan untuk membangun pemahaman teknis yang mendalam sebelum memasuki fase perancangan sistem. Kajian literatur difokuskan pada analisis mekanisme teknologi akses jarak jauh yang mampu beroperasi di balik kendala *Carrier-Grade NAT* (CGNAT), khususnya karakteristik protokol WireGuard dan arsitektur *Reverse Proxy* Nginx.

Selain tinjauan teknologi, studi ini juga mengkaji standar internasional yang digunakan sebagai parameter validasi. Hal ini mencakup pedoman audit keamanan *Blue Teaming* berdasarkan NIST SP 800-115 untuk prosedur penemuan dan verifikasi kerentanan, serta standar klasifikasi kualitas layanan (*Quality of Service*) mengacu pada ETSI TIPHON TR 101 329. Sintesis dari berbagai literatur tersebut menjadi acuan utama dalam penentuan variabel, topologi, dan skenario pengujian yang dirancang pada sub-bab berikutnya.

3.3 Arsitektur Sistem dan Infrastruktur

Dalam penelitian ini, infrastruktur jaringan dibangun menggunakan model arsitektur *Hub-and-Spoke* hibrida. Pendekatan ini mengintegrasikan layanan komputasi awan (*cloud*) sebagai gerbang pusat dengan infrastruktur server lokal (*on-premise*) yang berada di balik jaringan privat.

Visualisasi topologi sistem secara menyeluruh dapat dilihat pada Gambar 3.2. Sebagaimana diilustrasikan, VPS Gateway berfungsi sebagai penghubung (*hub*) yang menjembatani koneksi aman dari klien eksternal menuju server NAS (*spoke*) melalui terowongan VPN untuk menembus batasan CGNAT.



Gambar 3.2. Rancangan topologi jaringan *Hub-and-Spoke* menggunakan VPS Gateway

Adapun rincian spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam arsitektur tersebut adalah sebagai berikut:

Dengan desain arsitektur yang telah tersusun, tahap selanjutnya adalah merealisasikan rancangan tersebut ke dalam infrastruktur yang fungsional. Sub-bab berikut akan menjelaskan prosedur teknis implementasi sistem mulai dari konfigurasi server hingga pembentukan koneksi aman.

3.4 Prosedur Implementasi Sistem

Sebelum dilakukan pengujian kinerja, lingkungan eksperimen dibangun secara bertahap mengikuti topologi yang telah ditetapkan. Proses implementasi dibagi menjadi tiga fase utama untuk memastikan integritas konfigurasi pada setiap simpul jaringan.

3.4.1 Konfigurasi Gerbang Pusat (Cloud Gateway)

Fase awal difokuskan pada penyiapan VPS sebagai titik masuk publik (*public entry point*). Proses ini meliputi:

1. Inisialisasi Lingkungan: Penyebaran (*deployment*) sistem operasi Ubuntu Server pada instans VPS, diikuti dengan pembaruan paket sistem dan konfigurasi *firewall* dasar (UFW) untuk menutup seluruh *port* kecuali SSH.

2. Implementasi WireGuard: Instalasi modul kernel WireGuard, pembuatan antarmuka virtual (`wg0`), serta pembangkitan pasangan kunci kriptografi (*Public/Private Key*) untuk sisi server.

3. Setup Reverse Proxy: Konfigurasi Nginx untuk menangani terminasi SSL/TLS. Sertifikat digital diterbitkan menggunakan *Let's Encrypt* melalui tantangan DNS-01 guna memvalidasi kepemilikan domain tanpa memerlukan akses HTTP port 80 secara langsung.

3.4.2 Konfigurasi Node Lokal (On-Premise)

Fase kedua melibatkan penyiapan infrastruktur di sisi jaringan privat (rumah) yang berada di balik CGNAT:

1. Virtualisasi NAS: Instalasi sistem operasi *OpenMediaVault* (OMV) sebagai mesin virtual di atas *Hypervisor* Proxmox VE, dengan alokasi sumber daya yang terisolasi.

2. Integrasi Klien VPN: Konfigurasi WireGuard pada sisi Proxmox/OMV untuk bertindak sebagai *Peer*. Parameter `PersistentKeepalive` diaktifkan untuk menjaga terowongan tetap terbuka (NAT *Traversal*) meskipun tidak ada lalu lintas data aktif.

3.4.3 Pembentukan Terowongan dan Validasi Koneksi

Fase terakhir adalah penghubungan kedua segmen jaringan. Pertukaran kunci publik dilakukan antara VPS dan Node Lokal untuk membentuk terowongan terenkripsi. Validasi konektivitas dilakukan melalui uji *ping* antar-antarmuka virtual VPN dan pemeriksaan status *handshake* untuk memastikan jalur komunikasi dua arah telah terbentuk dengan enkripsi yang aktif.

Setelah seluruh komponen sistem terintegrasi dan terverifikasi berfungsi, tahap penelitian berlanjut pada eksekusi eksperimen. Bagian selanjutnya akan merinci skenario-skenario pengujian yang dirancang untuk mengevaluasi kinerja sistem tersebut.

3.5 Skenario Pengujian

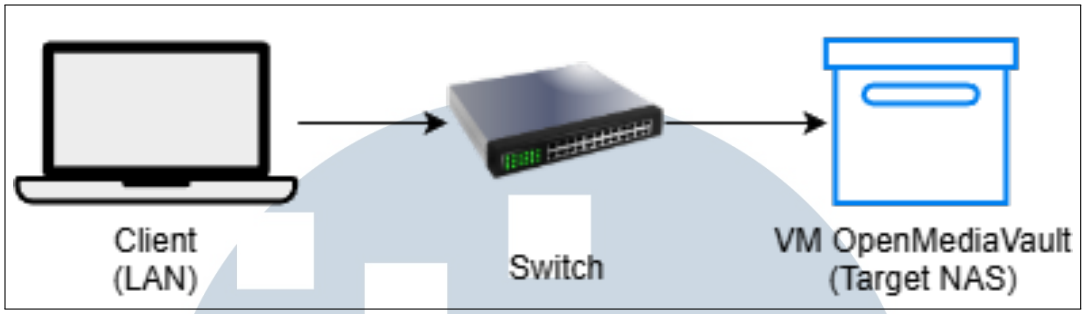
Rangkaian pengujian dalam penelitian ini dirancang secara sistematis ke dalam empat skenario arsitektur yang berbeda. Pembagian skenario ini bertujuan untuk mengevaluasi dampak implementasi setiap lapisan keamanan baik pada level jaringan maupun aplikasi terhadap degradasi performa (*latency* dan *throughput*) serta peningkatan postur keamanan sistem secara keseluruhan.

Berikut adalah rincian konfigurasi teknis untuk masing-masing skenario pengujian:

1. Skenario 1 (*Baseline* Lokal)

   Skenario ini merupakan konfigurasi akses langsung melalui jaringan lokal (LAN) tanpa penerapan mekanisme enkripsi tambahan, sebagaimana diilustrasikan pada Gambar 3.3. Dalam skenario ini, klien terhubung langsung ke *switch* dan berkomunikasi dengan server NAS menggunakan protokol HTTP standar.

   Fungsi utama dari skenario baseline adalah bertindak sebagai kontrol positif (*positive control*) dalam eksperimen. Pengujian ini bertujuan untuk mengukur kinerja murni (*raw performance*) dari perangkat keras tanpa beban komputasi enkripsi.

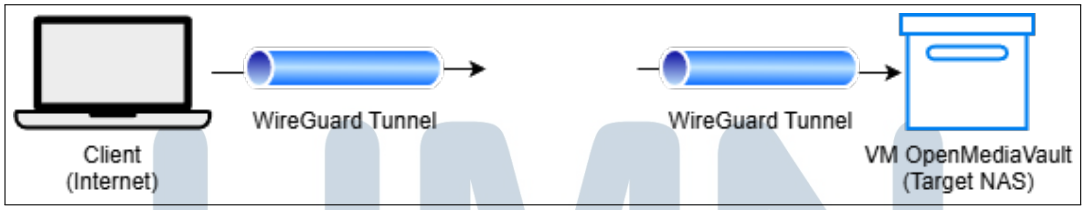


Gambar 3.3. Ilustrasi skenario 1: Koneksi *baseline* lokal (LAN)

2. Skenario 2 (VPN WireGuard)

   Skenario kedua menerapkan pengamanan pada lapisan jaringan (*Network-Layer Security*) menggunakan protokol WireGuard. Seperti yang ditunjukkan pada Gambar 3.4, lalu lintas data dari klien dibungkus (*encapsulated*) ke dalam terowongan VPN yang aman menuju VPS Gateway.

   Pada konfigurasi ini, seluruh paket data, termasuk header IP asli, dienkripsi menggunakan algoritma ChaCha20-Poly1305. VPS bertindak sebagai titik relai (*hub*) yang memungkinkan klien menembus batasan CGNAT.

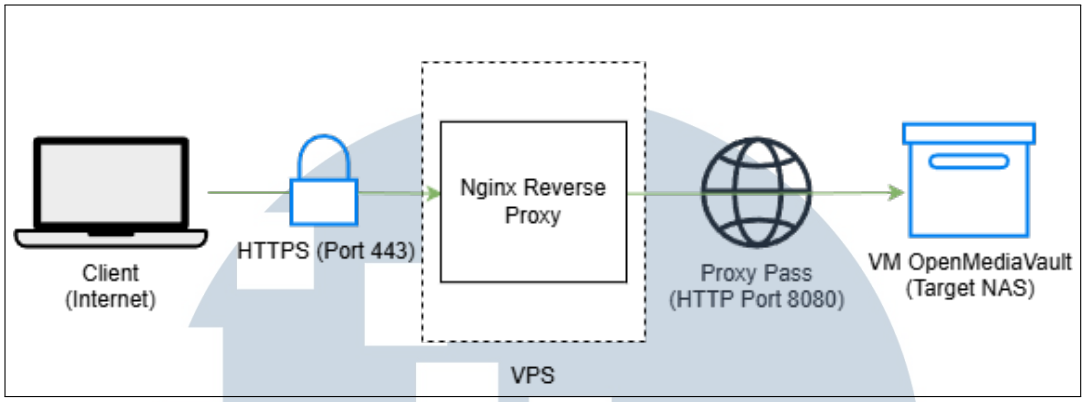


Gambar 3.4. Ilustrasi skenario 2: Koneksi jarak jauh menggunakan VPN WireGuard

3. Skenario 3 (Nginx Reverse Proxy)

   Skenario ketiga berfokus pada pengamanan lapisan aplikasi (*Application-Layer Security*) dengan memanfaatkan Nginx sebagai *Reverse Proxy*. Visualisasi alur data untuk skenario ini dapat dilihat pada Gambar 3.5, di mana koneksi diamankan menggunakan protokol HTTPS (TLS 1.2/1.3).

   Dalam topologi ini, proses terminasi TLS (enkripsi dan dekripsi) dilakukan secara terpusat di VPS Gateway. Nginx bertugas meneruskan permintaan tersebut ke server NAS melalui terowongan WireGuard di sisi *backend*.

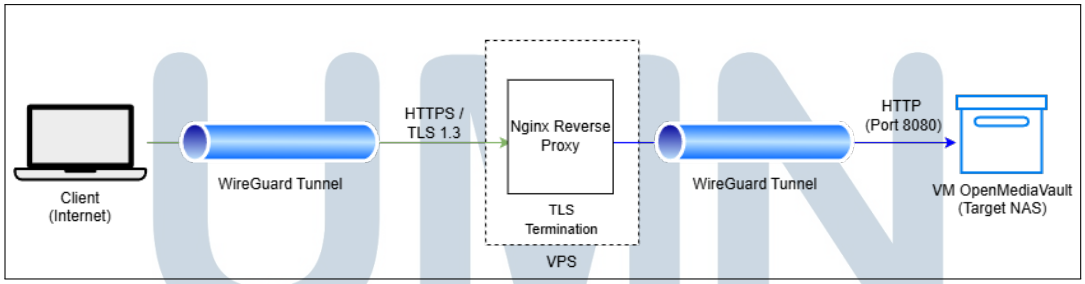


Gambar 3.5. Ilustrasi skenario 3: Koneksi menggunakan Nginx Reverse Proxy (HTTPS)

4. Skenario 4 (*Hybrid*)

   Skenario terakhir menerapkan prinsip *Defense-in-Depth* dengan menggabungkan keunggulan Skenario 2 dan Skenario 3. Sebagaimana digambarkan pada Gambar 3.6, arsitektur ini menciptakan lapisan enkripsi ganda (*double encryption*).

   Lalu lintas HTTPS dari lapisan aplikasi dibungkus kembali di dalam terowongan WireGuard pada lapisan jaringan, menawarkan tingkat kerahasiaan tertinggi.



Gambar 3.6. Ilustrasi skenario 4: Arsitektur keamanan hibrida (VPN + Nginx)

Setelah seluruh desain skenario arsitektur didefinisikan, langkah selanjutnya adalah menentukan indikator keberhasilan pengujian. Bagian berikut akan menjabarkan secara rinci variabel dan metrik ukur yang digunakan untuk mengevaluasi kinerja masing-masing skenario tersebut.

3.6 Verifikasi Fungsionalitas Sistem (Pra-Pengujian)

Sebelum dilakukan pengambilan data performa dan audit keamanan, dilakukan tahap verifikasi fungsionalitas untuk memastikan kesiapan seluruh

komponen arsitektur. Tahapan ini bertujuan memvalidasi bahwa jalur komunikasi antar-node (Klien, VPS Gateway, dan NAS Spoke) telah terbentuk dengan benar dan stabil. Prosedur verifikasi mencakup:

1. Uji Konektivitas Dasar (L3): Melakukan `ping` antar-titik (Point-to-Point) untuk memastikan terowongan WireGuard telah aktif dan dapat melewatkan paket ICMP.
2. Uji Aksesibilitas Layanan (L7): Memastikan antarmuka web OpenMediaVault dapat diakses melalui browser klien, baik via IP privat VPN maupun via domain HTTPS Nginx.
3. Validasi Handshake: Memeriksa status *latest handshake* pada antarmuka WireGuard untuk mengonfirmasi bahwa pertukaran kunci kriptografi telah berhasil.

Apabila salah satu indikator di atas gagal, proses akan dikembalikan ke tahap implementasi untuk perbaikan konfigurasi (*troubleshooting*) sebagaimana digambarkan pada alur penelitian (Gambar 3.1).

3.7 Parameter dan Metrik Pengujian

Evaluasi efektivitas sistem dilakukan melalui pengukuran variabel terikat yang diklasifikasikan ke dalam dua dimensi utama untuk mendapatkan analisis yang komprehensif. Dimensi pertama berfokus pada metrik kinerja jaringan (*network performance*) yang diukur secara kuantitatif guna menilai efisiensi transmisi data. Dimensi kedua berfokus pada postur keamanan (*security posture*) yang divalidasi secara kualitatif untuk memastikan ketahanan sistem terhadap potensi ancaman dan kebocoran data.

3.7.1 Parameter Performa (Kuantitatif)

Parameter ini digunakan untuk merepresentasikan kualitas layanan jaringan secara numerik:

1. *Latency* (*Round-Trip Time*): Waktu tempuh paket bolak-balik dalam milidetik (ms). Diukur menggunakan `ping` dengan ukuran sampel $n = 300$ per skenario untuk memenuhi kaidah *Central Limit Theorem*.

2. *Throughput* (*Bandwidth* Efektif): Laju transfer data maksimum dalam Mbps. Diukur menggunakan `iperf3` (TCP Mode) selama 30 detik dengan $n = 30$ pengulangan (Hanya pada Skenario 1 dan 2).

3. HTTP *Transfer Speed*: Kecepatan unduh berkas riil dalam MB/s. Diukur menggunakan `curl` dengan variasi berkas 500 MB dan 1 GB, diulang sebanyak $n = 40$ kali per skenario.

Hasil pengukuran parameter performa, khususnya *latency*, selanjutnya akan diklasifikasikan ke dalam kategori kualitas layanan (*Quality of Service*) mengacu pada standar ETSI TIPHON TR 101 329. Klasifikasi ini bertujuan untuk menentukan tingkat kelayakan kinerja jaringan pada setiap skenario pengujian sebagaimana telah dijabarkan pada Landasan Teori.

3.7.2 Parameter Keamanan (Kualitatif & Validasi)

Parameter ini bersifat biner atau deskriptif untuk memverifikasi keberhasilan mekanisme proteksi:

1. *Confidentiality* (Kerahasiaan): Status keterbacaan *payload* data saat melintasi jaringan publik (*Plaintext* vs *Ciphertext*).

2. *Attack Surface* (Permukaan Serangan): Jumlah dan jenis *port* yang terbuka pada VPS Gateway yang dapat diakses dari internet publik.

3. *Configuration Strength*: Kekuatan algoritma kriptografi (*Cipher Suite*) dan versi protokol yang dinegosiasikan saat koneksi terjadi.

Guna melakukan pengukuran dan validasi terhadap parameter-parameter di atas secara akurat, penelitian ini memanfaatkan serangkaian perangkat lunak dan alat bantu audit yang terstandarisasi. Rincian spesifikasi instrumen teknis tersebut akan diuraikan pada sub-bab berikut.

3.8 Instrumen dan Alat Pengujian

Ringkasan perangkat lunak dan alat yang digunakan ditampilkan pada Tabel 3.1.

Tabel 3.1. Daftar alat pengujian (*Testing Tools*)

Kategori	Nama Tool	Fungsi Spesifik
Performa	`ping`	Mengukur *latency* dan *jitter* (ICMP)
	`iperf3`	Mengukur *throughput* TCP
	`curl`	Mengukur waktu transfer HTTP/HTTPS
	`bash`	Bahasa skrip untuk otomatisasi
Keamanan	`Wireshark/Tshark`	Analisis paket (*Packet Capture*)
	`Nmap`	Pemindaian *port* (*Attack Surface*)
	`OpenSSL`	Audit sertifikat dan konfigurasi TLS

Dengan tersedianya instrumen pengujian yang memadai, tahapan eksekusi eksperimen dapat dilakukan secara terstruktur. Bagian selanjutnya akan menjelaskan prosedur teknis pengambilan data performa yang dilakukan melalui mekanisme otomatisasi guna menjamin konsistensi hasil pengukuran.

3.9 Prosedur Uji Performa (Otomatisasi)

Pengambilan data performa diorkestrasi menggunakan skrip `run-all-tests.sh`. Skrip ini bertugas memvalidasi koneksi sebelum memulai pengukuran untuk menjaga konsistensi data, sebagaimana ditampilkan pada Kode 3.1.

```
echo "Target: $TARGET_IP"
echo "Required VPN Status: $VPN_STATUS"

# Memastikan kondisi jaringan sesuai skenario sebelum tes dimulai
read -p "Apakah WireGuard sudah $VPN_STATUS? (y/n): " confirm
if [[ $confirm != "y" ]]; then
    echo "Cancelled. Please check WireGuard status!"
    exit 1
fi
```

Kode 3.1: Logika validasi koneksi pada skrip otomatisasi

Untuk metrik *latency*, skrip `test-latency.sh` mengeksekusi *ping* dan menggunakan *Regular Expression* (Regex). Seperti terlihat pada Kode 4.2, proses ini mengekstraksi nilai waktu secara presisi ke format CSV.

```
grep "icmp_seq=" "$TEMPFILE" | while read line; do
    # Parsing nilai sequence dan time menggunakan Regex
    SEQ=$(echo "$line" | grep -oP 'icmp_seq=\K[0-9]+')
```

```
    TIME=$(echo "$line" | grep -oP 'time=\K[0-9.]+')

    echo "$SEQ,$TIMESTAMP,$TIME" >> "$OUTFILE"
done
```

Kode 3.2: Ekstraksi data *latency* ke format CSV

Selain pengukuran kinerja yang bersifat kuantitatif, validasi kualitatif terhadap aspek keamanan juga mutlak diperlukan untuk memastikan integritas arsitektur pertahanan. Oleh karena itu, prosedur berikut difokuskan pada langkah-langkah audit teknis untuk memverifikasi efektivitas enkripsi.

3.10 Prosedur Uji Keamanan (Validasi & Audit)

Sesuai dengan batasan masalah, pengujian keamanan berfokus pada validasi arsitektur pertahanan (*Blue Teaming*). Seluruh rangkaian prosedur pengujian keamanan dalam penelitian ini mengacu pada standar teknis NIST *Special Publication* (SP) 800-115 (*Technical Guide to Information Security Testing and Assessment*), dengan fokus pada fase penemuan (*Discovery*) dan verifikasi kerentanan (*Verification*).

1. Validasi Enkripsi (Fase Verification):
   (a) `Wireshark` dijalankan pada antarmuka jaringan fisik klien.
   (b) Dilakukan transfer data yang mengandung informasi sensitif (simulasi kredensial) melalui jalur Baseline, VPN, dan Proxy.
   (c) Paket yang tertangkap dianalisis menggunakan fitur *Follow TCP Stream*.
   (d) Indikator: Pada Baseline teks harus terbaca jelas, sedangkan pada skenario aman *payload* harus berupa karakter acak (*ciphertext*).
2. Audit Permukaan Serangan (Fase Discovery):
   (a) Melakukan pemindaian terhadap IP Publik VPS menggunakan `Nmap` dengan parameter `-sS` (*TCP SYN Scan*) dan `-p-` (seluruh 65535 port).
   (b) Indikator: Memastikan hanya *port* esensial (misal: SSH, WireGuard, HTTP/S) yang berstatus *Open*.
3. Verifikasi Konfigurasi SSL/TLS (Fase Verification):
   (a) Menggunakan `OpenSSL` untuk melakukan *handshake* manual ke Nginx.

(b) Indikator: Memverifikasi bahwa protokol yang digunakan adalah TLS 1.2 atau 1.3 dan *cipher suite* yang digunakan tergolong kuat.

Data empiris yang diperoleh dari seluruh prosedur pengujian performa dan keamanan di atas selanjutnya akan dikompilasi, diolah, dan dianalisis secara mendalam pada Bab 4 guna menjawab rumusan masalah penelitian.

3.11 Teknik Analisis dan Komparasi Data

Tahap akhir dari metodologi penelitian adalah pengolahan data empiris yang telah dikumpulkan untuk menjawab rumusan masalah. Proses analisis dilakukan melalui dua pendekatan komparatif:

1. Analisis Komparatif Performa (Kuantitatif): Data hasil pengukuran (*latency, throughput, file transfer speed*) dari ketiga skenario akses (WireGuard, Nginx, Hybrid) akan disandingkan dengan data *Baseline* (LAN) menggunakan metode perbandingan statistik deskriptif. Data akan divisualisasikan dalam bentuk grafik batang dan tabel untuk memperjelas degradasi performa (*overhead*) yang ditimbulkan oleh masing-masing mekanisme keamanan. Analisis ini mengacu pada standar klasifikasi *Quality of Service* (QoS) ETSI TIPHON untuk menentukan kategori kelayakan layanan.
2. Analisis Komparatif Keamanan (Kualitatif): Temuan dari audit keamanan akan dipetakan ke dalam matriks perbandingan berdasarkan parameter kerahasiaan (*confidentiality*) dan luas permukaan serangan (*attack surface*). Analisis ini bertujuan untuk menilai efektivitas kontrol keamanan dalam memitigasi risiko, sesuai dengan kriteria validasi yang ditetapkan pada standar NIST SP 800-115.

Hasil dari kedua analisis tersebut kemudian disintesis untuk menentukan rekomendasi metode akses jarak jauh yang menawarkan keseimbangan optimal (*trade-off*) antara kinerja dan keamanan pada lingkungan NAS di balik CGNAT.