Full Paper

Implementasi Zero Trust Network Architecture (ZTNA) untuk Keamanan Akses Jaringan Perusahaan

Ria Agustina[1], Christian Mitchell Ng[2], Reyhan Eka Saptian Nugroho[3], Faris Zaidan Naufal[4], Deardo Christian Wulur[5], Sindy Silvia Soarta Pakpahan[6], Bagus Maulana[7]
[1,2,3,4,5,6,7]Program Studi Teknik Komputer, Institut Teknologi Batam, Indonesia

Informasi Artikel

Terbit: Juli 2026

Kata Kunci:

Keamanan Jaringan,
Zero Trust, Network Architecture
Autentikasi multi faktor
Jaringan Nirkabel
Network Development Life Cycle
Ferrumgate

ABSTRAK

Penelitian ini membahas penerapan Arsitektur Jaringan *Zero Trust* untuk meningkatkan keamanan akses pada jaringan nirkabel mahasiswa di Institut Teknologi Batam. Penelitian ini dilatarbelakangi oleh meningkatnya ancaman terhadap jaringan kampus akibat lemahnya autentikasi dan kontrol akses pada sistem berbasis perimeter konvensional. Pendekatan yang diterapkan berfokus pada prinsip bahwa tidak ada entitas yang dipercaya tanpa verifikasi, melalui autentikasi identitas dan perangkat, kontrol kebijakan akses, dan pemantauan aktivitas pengguna secara komprehensif. Metode penelitian menggunakan tahapan *Network Development Life Cycle*, yang meliputi analisis kebutuhan, perancangan arsitektur, implementasi sistem *Ferrumgate* berbasis autentikasi multifaktor, dan pengujian kondisi jaringan sebelum dan sesudah penerapan model keamanan. Hasil penelitian menunjukkan bahwa implementasi Arsitektur Jaringan *Zero Trust* berhasil membatasi akses pengguna yang tidak sah, meningkatkan visibilitas aktivitas jaringan, dan memperkuat perlindungan terhadap ancaman siber tanpa mengurangi kinerja jaringan secara signifikan. Penelitian ini telah menghasilkan model keamanan adaptif yang dapat diterapkan di lingkungan akademik untuk meningkatkan ketahanan jaringan terhadap serangan siber.

Corresponding Author:

Nama Penulis, Ria Agustina
Email: riia048999@gmail.com

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan besar terhadap cara manusia berinteraksi, bekerja, dan bertransaksi secara digital. Internet kini menjadi infrastruktur vital yang menopang berbagai sektor, mulai dari pendidikan, bisnis, hingga pemerintahan. Namun, peningkatan konektivitas juga berbanding lurus dengan meningkatnya ancaman keamanan siber yang semakin canggih dan sulit dikendalikan. Serangan seperti *phishing, malware, ransomware,* dan *data breach* terus meningkat dari tahun ke tahun, menimbulkan kerugian ekonomi dan reputasi yang signifikan bagi organisasi [1]. Kegagalan dalam mendeteksi dan mencegah ancaman siber dapat berakibat pada pencurian data penting, gangguan operasional, bahkan kolapsnya sistem jaringan [2].

Sistem keamanan jaringan tradisional umumnya masih mengandalkan model perimeter-based security, di mana semua entitas di dalam jaringan dianggap tepercaya, sedangkan ancaman hanya diasumsikan berasal dari luar jaringan. Namun, pendekatan ini kini tidak lagi memadai karena meningkatnya kompleksitas jaringan modern, penggunaan layanan *cloud computing*, serta pola kerja jarak jauh (*remote working*) yang memperluas permukaan serangan [3]. Menurut Tang et al. [4], batas keamanan jaringan tradisional menjadi kabur akibat dinamika arsitektur jaringan dan integrasi sistem lintas platform. Dengan demikian, dibutuhkan paradigma baru yang mampu menjamin keamanan berbasis verifikasi menyeluruh terhadap setiap entitas yang

mengakses sumber daya jaringan. Untuk mengatasi kelemahan tersebut, konsep *Zero Trust Network* (ZTN) dikembangkan oleh Paul Simmonds pada tahun 2004 melalui gagasan deperimeterization, yang kemudian menjadi dasar filosofi keamanan "*never trust, always verify*" [5]. *Zero Trust* menekankan bahwa tidak ada pengguna maupun perangkat yang otomatis dipercaya, bahkan ketika sudah berada di dalam jaringan. Setiap permintaan akses harus diverifikasi menggunakan kebijakan berbasis identitas, perangkat, lokasi, dan tingkat risiko. Pendekatan ini memungkinkan sistem keamanan lebih adaptif terhadap ancaman internal dan eksternal [6].

Model *Zero Trust* semakin relevan di tengah maraknya kasus kebocoran data di Indonesia. Sebagai contoh, serangan *ransomware* terhadap Bank Syariah Indonesia (BSI) pada tahun 2023 menyebabkan lumpuhnya layanan mobile banking selama beberapa hari dan pencurian data nasabah hingga 1,5 terabyte [7]. Kasus lain terjadi pada tahun 2022 ketika kelompok hacker Bjorka membocorkan data pelanggan PLN, registrasi SIM card, dan pelamar Pertamina [8]. Insiden-insiden tersebut menunjukkan lemahnya mekanisme autentikasi dan kontrol akses dalam sistem keamanan konvensional.

Beberapa penelitian terdahulu telah mengimplementasikan model *Zero Trust* menggunakan platform seperti *Cloudflare Zero Trust* dan *ZeroTier*. Adhar dan Saprudin [9] menunjukkan bahwa penggunaan Cloudflare Zero Trust dapat mendeteksi aktivitas berbahaya seperti *crypto jacking* tanpa memerlukan perangkat keras tambahan. Sementara itu, Dancheva et al. [10] membuktikan bahwa *ZeroTier* dapat menyediakan koneksi aman berbasis *peer-to-peer* dengan enkripsi *end-to-end* untuk mendukung kerja jarak jauh. Meski demikian, sebagian besar solusi tersebut masih bergantung pada sistem berlisensi tertutup (*non–open source*) dan belum mengintegrasikan autentikasi multi-faktor (*Multi-Factor Authentication* / MFA) secara menyeluruh. Berdasarkan kelemahan penelitian sebelumnya, penelitian ini mengusulkan penerapan *Zero Trust Network* berbasis platform *Ferrumgate*, yaitu sistem keamanan *open-source* yang mendukung MFA menggunakan *Google Authenticator* serta dilengkapi dengan fitur real-time user monitoring. Platform ini dirancang untuk memastikan bahwa hanya pengguna dan perangkat terverifikasi yang dapat mengakses sumber daya jaringan, sehingga dapat meminimalkan risiko serangan dari pihak internal maupun eksternal.

Penelitian ini menggunakan metode *Network Development Life Cycle* (NDLC) sebagai pendekatan pengembangan sistem yang sistematis, meliputi tahap analisis kebutuhan, perancangan arsitektur jaringan tersegmentasi, implementasi sistem autentikasi, serta pemantauan aktivitas pengguna. NDLC dipilih karena mampu memberikan siklus pengembangan yang berkelanjutan dan terukur untuk penerapan kebijakan keamanan berbasis *Zero Trust* [11]. Nilai kebaruan dari penelitian ini terletak pada implementasi model *Zero Trust* berbasis *open-source Ferrumgate* dengan integrasi *Multi-Factor Authentication* (MFA) yang dikembangkan menggunakan pendekatan NDLC. Penelitian ini tidak hanya menunjukkan efektivitas *Ferrumgate* dalam membatasi akses berdasarkan verifikasi identitas dan perangkat, tetapi juga mengusulkan model keamanan yang efisien, adaptif, dan dapat direplikasi untuk lingkungan akademik maupun korporasi tanpa ketergantungan pada solusi berbayar. Dengan demikian, hasil penelitian ini diharapkan dapat menjadi kontribusi nyata dalam meningkatkan ketahanan sistem jaringan terhadap ancaman siber yang terus berkembang.

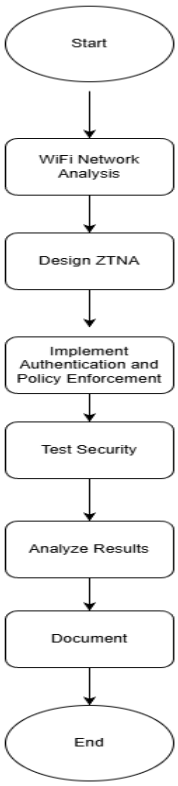
2. METODE PENELITIAN

Menjelaskan kronologis penelitian, termasuk desain penelitian, prosedur penelitian (dalam bentuk algoritma, *Pseudocode* atau lainnya), cara pengujian dan akuisisi data [1-3]. Uraian jalannya penelitian harus didukung dengan referensi, sehingga penjelasannya dapat diterima secara ilmiah [2, 4]. Tabel dan Gambar disajikan di tengah, seperti yang ditunjukkan di bawah ini dan dikutip dalam naskah.

Tabel 1. Tahapan Penelitian Berdasarkan Metode NDLC

Tahap NDLC	Deskripsi Kegiatan	Output
Analysis	Identifikasi arsitektur jaringan WiFi mahasiswa dan kelemahannya terhadap serangan internal maupun eksternal.	Daftar kebutuhan keamanan dan risiko jaringan.
Design	Merancang arsitektur Zero Trust dengan komponen Ferrumgate, Policy Engine, dan MFA.	Topologi jaringan konseptual berbasis ZTNA.
Simulation/ Implementation	Instalasi dan konfigurasi Ferrumgate menggunakan server virtual, integrasi dengan Google Authenticator untuk MFA	Sistem autentikasi dan policy-based access control berjalan.
Monitoring	Mengamati aktivitas pengguna dan mencatat log akses melalui dashboard Ferrumgate	Data pengujian (latency, blocking unauthorized user, log aktivitas).
Testing	Pengujian keamanan dan performa jaringan sebelum dan sesudah penerapan ZTNA.	Data pengujian (latency, blocking unauthorized user, log aktivitas).
Evaluation	Analisis hasil pengujian untuk menentukan efektivitas model ZTNA.	Laporan analisis efektivitas dan rekomendasi penerapan.

Berikut merupakan desain prosedur penelitian:



Gambar 1 Flowchart NDLC Adaptasi untuk ZTNA

2.3. Algoritma Penelitian (Pseudocode)

Pseudocode di bawah ini menggambarkan alur autentikasi pengguna pada sistem *Zero Trust Network Architecture (ZTNA)* berbasis *Ferrumgate* yang diimplementasikan dalam jaringan WiFi kampus. Algoritma ini dirancang untuk menggambarkan bagaimana setiap permintaan akses dari pengguna harus melalui beberapa tahapan verifikasi berlapis, sesuai dengan prinsip dasar Zero Trust yaitu *"Never Trust, Always Verify"*. Algoritma: ZTNA_Access_Control.

Tabel 2. Pseudocode sistem Zero Trust Network Architecture (ZTNA) berbasis Ferrumgate

```
Algorithm: ZTNA_Access_Control
Input: User_ID, Password, Device_ID, MFA_Code
Output: Access Granted or Access Denied

BEGIN
  Step 1: Receive_Access_Request(User_ID, Device_ID)
      Log_Request(User_ID, Device_ID, Timestamp)

  Step 2: Verify_Credentials(User_ID, Password)
      IF Verification_Failed THEN
        Deny_Access()
        Log("Invalid Credentials", User_ID, Timestamp)
        EXIT
      ENDIF

  Step 3: Validate_Device(Device_ID)
      IF Device_Not_Registered OR Device_Flagged THEN
        Deny_Access()
        Log("Untrusted Device Attempt", Device_ID, Timestamp)
        EXIT
      ENDIF

  Step 4: Initiate_MFA(User_ID)
      Send_MFA_Request(User_ID)
      IF MFA_Code != Generate_MFA_Token(User_ID) THEN
        Deny_Access()
```

```
        Log("MFA Verification Failed", User_ID, Timestamp)
        EXIT
      ENDIF

  Step 5: Apply_Access_Policy(User_ID, Resource)
      Policy_Result ← PolicyEngine.Evaluate(User_ID, Device_ID, Resource)
      IF Policy_Result = "Allow" THEN
        Generate_Session_Token(User_ID)
        Grant_Access(Resource)
        Log("Access Granted", User_ID, Resource, Timestamp)
      ELSE
        Deny_Access()
        Log("Access Blocked by Policy", User_ID, Resource, Timestamp)
        EXIT
      ENDIF

  Step 6: Monitor_Active_Session(User_ID)
      Track_Session(Activity_Level, Data_Usage, Duration)
      IF Anomalous_Behavior_Detected(User_ID) THEN
        Revoke_Session_Token(User_ID)
        Alert_Admin("Potential Security Threat", User_ID)
        Log("Session Terminated due to Anomaly", User_ID, Timestamp)
      ENDIF
END
```

Penjelasan Detail Setiap Langkah :

Step 1: Receive_Access_Request(User_ID, Device_ID)

Tahapan ini memulai proses autentikasi ketika pengguna mencoba mengakses jaringan kampus, misalnya melalui SSID *WiFi Mahasiswa*. Sistem mencatat permintaan akses yang masuk dengan mengidentifikasi *User_ID* dan *Device_ID.* Pada konteks Ferrumgate, permintaan akses ini diarahkan ke gateway yang berperan sebagai *reverse proxy* dan *policy enforcement point* (PEP), yang akan menentukan apakah permintaan dapat diteruskan ke sumber daya jaringan atau tidak. Semua data awal seperti alamat IP, MAC address, dan waktu permintaan dicatat untuk keperluan audit dan analisis keamanan.

Step 2: Verify_Credentials(User_ID, Password)

Pada tahap ini, kredensial pengguna diverifikasi terhadap *Identity Provider (IdP)* seperti LDAP atau database lokal *Ferrumgate*. Jika kredensial tidak valid, sistem segera menghentikan proses dengan perintah Deny_Access() dan mencatat log kesalahan dengan label "*Invalid Credentials*". Verifikasi kredensial ini berfungsi sebagai filter awal untuk mencegah akses dari pengguna anonim atau yang mencoba melakukan *brute force attack*. Validasi ini memastikan bahwa hanya identitas yang terdaftar dalam sistem yang dapat melanjutkan proses autentikasi berikutnya.

Step 3: Validate_Device(Device_ID)

Setelah identitas pengguna diverifikasi, sistem melakukan pemeriksaan terhadap perangkat yang digunakan. Proses ini membandingkan *Device_ID* (biasanya berupa MAC Address atau UUID perangkat) dengan daftar perangkat yang telah terdaftar pada sistem keamanan kampus. Jika perangkat tersebut tidak dikenal, pernah digunakan dalam aktivitas mencurigakan, atau berada di luar lokasi geofencing yang diizinkan, maka sistem akan menolak akses secara otomatis. Tahap ini menggambarkan prinsip *device trust verification* dari arsitektur *Zero Trust*, di mana keamanan tidak hanya bergantung pada identitas pengguna tetapi juga pada keandalan perangkat yang digunakan.

Step 4: Initiate_MFA(User_ID)

Pada tahapan ini, sistem mengaktifkan proses *Multi-Factor Authentication (MFA)* untuk memperkuat validasi identitas pengguna. Pengguna menerima permintaan MFA, misalnya melalui *Google Authenticator* atau OTP Email, dan diminta untuk memasukkan kode verifikasi. Ferrumgate kemudian mencocokkan kode yang dimasukkan dengan token MFA yang dihasilkan secara dinamis (Generate_MFA_Token(User_ID)). Apabila kode salah atau waktu validasi sudah kedaluwarsa, sistem akan menolak akses dan mencatat percobaan gagal tersebut. Tahapan ini merupakan implementasi dari layer kepercayaan tambahan **(***secondary trust layer***)** yang mencegah akses ilegal meskipun kredensial dasar telah diketahui oleh pihak ketiga.

Step 5: Apply_Access_Policy(User_ID, Resource)

Tahapan ini merupakan inti dari sistem Zero Trust, di mana keputusan akhir diberikan berdasarkan *policy engine* yang menilai atribut pengguna, perangkat, waktu, dan jenis sumber daya yang diminta. Sistem *Ferrumgate* menggunakan *context-aware policy* untuk menentukan apakah pengguna memiliki hak akses ke sumber daya tertentu. Jika hasil evaluasi menunjukkan "*Allow*", maka sistem:

1. Menghasilkan session token sementara.
2. Mengizinkan akses ke resource yang diminta.
3. Mencatat aktivitas dengan status "*Access Granted*".

Namun, jika kebijakan tidak mengizinkan (misalnya karena pengguna berada di luar jam operasional, perangkat belum tervalidasi penuh, atau resource bersifat sensitif), maka akses akan ditolak secara otomatis. Tahapan ini memastikan bahwa setiap keputusan diambil berdasarkan konteks dan risiko (*context-based access control*), bukan hanya status login.

Step 6: Monitor_Active_Session(User_ID)

Setelah pengguna diizinkan mengakses sistem, proses pemantauan sesi aktif (*active session monitoring*) dilakukan secara real-time. Sistem merekam berbagai parameter seperti durasi koneksi, penggunaan data, dan perilaku aktivitas jaringan. Jika sistem mendeteksi anomali seperti lonjakan trafik data, akses ke *resource* yang tidak biasa, atau aktivitas di luar lokasi normal maka token sesi akan segera dicabut (Revoke_Session_Token(User_ID)), dan admin keamanan akan menerima notifikasi. Tahap ini menegaskan bahwa *Zero Trust* bukan hanya proses autentikasi, melainkan siklus keamanan berkelanjutan yang selalu memverifikasi kepercayaan pengguna dan perangkat selama sesi berlangsung.

Keterkaitan Algoritma dengan Penelitian

Algoritma ini menggambarkan implementasi *ZTNA* berbasis *Ferrumgate* sebagai model penelitian yang bertujuan untuk:

1. Menguji efektivitas autentikasi berlapis dalam mengamankan jaringan WiFi kampus.
2. Mengevaluasi performa sistem dalam mendeteksi anomali dan upaya akses ilegal.
3. Mengukur dampak kebijakan *Zero Trust* terhadap pengalaman pengguna dan stabilitas koneksi.

Dengan menerapkan algoritma ini, penelitian membangun pendekatan sistematis terhadap keamanan jaringan yang tidak lagi bergantung pada perimeter tradisional, tetapi pada verifikasi identitas, perangkat, dan konteks penggunaan secara dinamis [3][4][5].

3. HASIL DAN ANALISIS

Hasil pengujian menunjukkan bahwa implementasi Zero Trust Network Access (ZTNA) melalui Cloudflare WARP berjalan efektif pada aspek keamanan, dibuktikan dengan aktifnya DNS proxy, stabilnya tunnel tanpa packet loss, serta berfungsinya DNS filtering yang memblokir beberapa domain berisiko. Namun, performa jaringan secara keseluruhan mengalami penurunan signifikan, terlihat dari kecepatan download sangat rendah (2.29 Mbps), latency tinggi (278 ms), jitter ekstrem (242 ms), dan packet loss mencapai 26.9%. Data ini menegaskan bahwa hambatan bukan berasal dari ZTNA atau WARP tunnel—karena latency tunnel tetap rendah (10–45 ms)—melainkan dari kondisi jaringan dasar seperti ISP atau infrastruktur lokal yang tidak stabil. Dengan demikian, meskipun ZTNA meningkatkan keamanan secara optimal, kualitas jaringan yang buruk sebelum memasuki tunnel menyebabkan penurunan pengalaman pengguna dan menjadi faktor utama menurunnya kinerja koneksi.

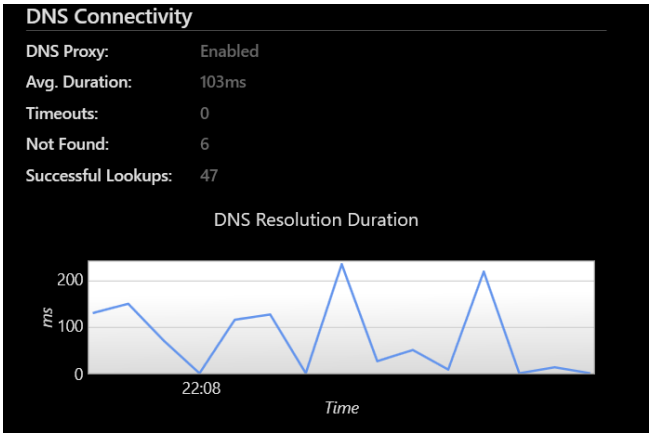
3.1. Hasil Pengujian Konektivitas dan Kinerja DNS

Gambar pertama menunjukkan hasil pengujian *DNS Connectivity* ketika ZTNA (melalui Cloudflare WARP/Zero Trust) diaktifkan. Beberapa poin penting:

1. DNS Proxy: Enabled menandakan bahwa permintaan DNS dikontrol melalui mekanisme Zero Trust.
2. Avg. Duration 103 ms mengindikasikan waktu resolusi DNS berada pada kisaran yang masih dapat diterima, meskipun agak tinggi.
3. Not Found: 6, menunjukkan adanya beberapa domain yang gagal ditemukan atau diblokir oleh kebijakan Zero Trust (misalnya karena filtering).
4. Grafik DNS Resolution Duration terlihat fluktuatif, menunjukkan adanya variasi beban atau jaringan.

Interpretasi dalam konteks ZTNA

Implementasi ZTNA memaksa seluruh traffic, termasuk DNS, melewati kontrol terpusat. Waktu resolusi DNS yang sedikit lebih tinggi adalah wajar, karena permintaan DNS diproses melalui gateway Zero Trust untuk validasi keamanan, bukan langsung ke DNS publik.



Gambar 2. Hasil Pengujian Konektivitas dan Kinerja DNS

3.2. Hasil Pengujian Kecepatan Internet (Cloudflare Speed Test)

Hasil speed test menunjukkan kondisi jaringan saat akses dilakukan melalui Zero Trust/WARP:
Download Speed: 2.29 Mbps

1. Jauh di bawah standar layanan internet normal.
2. Menandakan adanya bottleneck pada jalur keluar perusahaan atau kebijakan routing via ZTNA.

Upload Speed: 18.3 Mbps

1. Lebih tinggi dan relatif stabil dibanding download.
2. Menggambarkan bahwa traffic keluar lebih optimal dibanding traffic masuk.

Latency dan Jitter

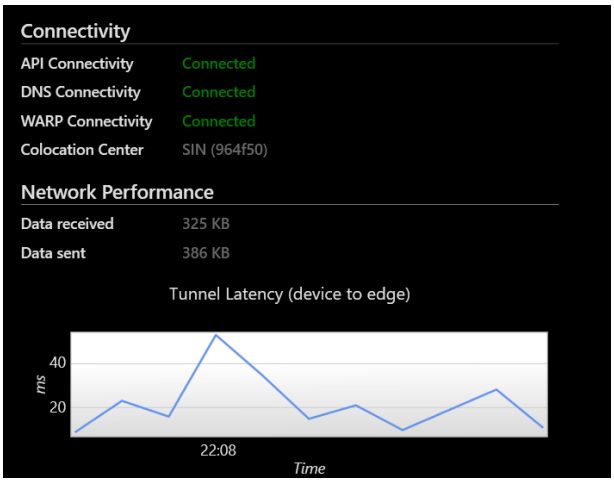
1. Latency: 278 ms (tinggi)
2. Jitter: 242 ms (sangat tinggi)
3. Packet Loss: 26.9% (kritikal)

Nilai-nilai ini menunjukkan infrastruktur jaringan perusahaan sedang dalam kondisi tidak stabil, atau beban WARP/ZTNA sedang tinggi.
Interpretasi dalam kerangka ZTNA

Dalam arsitektur Zero Trust:

1. Seluruh traffic tunnel dienkripsi.
2. TLS inspection, filtering, dan routing ke PoP (Point of Presence) ZT server dilakukan.
3. Akibatnya, kecepatan dan latensi sangat bergantung pada jalur terdekat (dalam kasus Anda: Singapore PoP).

Angka packet loss yang tinggi (>20%) sangat mempengaruhi pengalaman pengguna (video streaming, gaming, video chat semuanya dikategorikan *Bad* dalam laporan).



Gambar 3 Hasil Pengujian Kecepatan Internet (Cloudflare Speed Test)

3.3 Hasil Pengujian WARP dan ZTNA Tunnel

Gambar selanjutnya menunjukkan status koneksi WARP Zero Trust:

1. API Connectivity: Connected
2. DNS Connectivity: Connected
3. WARP Connectivity: Connected
4. Colocation Center: SIN (Singapore)

Artinya seluruh komponen Zero Trust aktif dan bekerja.

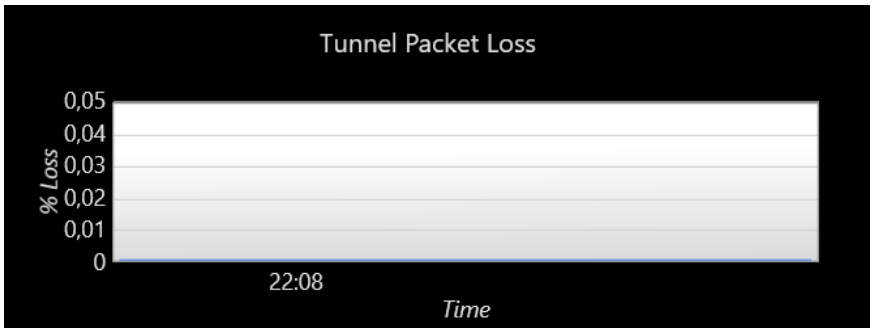
Tunnel Latency

1. Nilai tunnel latency berkisar 10–45 ms, lebih rendah dibanding latency total (278 ms).
2. Ini berarti bottleneck bukan pada WARP tunnel, tetapi pada:
   a. jalur internet lokal (ISP → Cloudflare PoP),
   b. atau kondisi jaringan lokal perusahaan.

Tunnel Packet Loss

1. Grafik menunjukkan 0% *loss*, berarti komunikasi antara perangkat Anda dan *edge Zero Trust* sangat stabil.

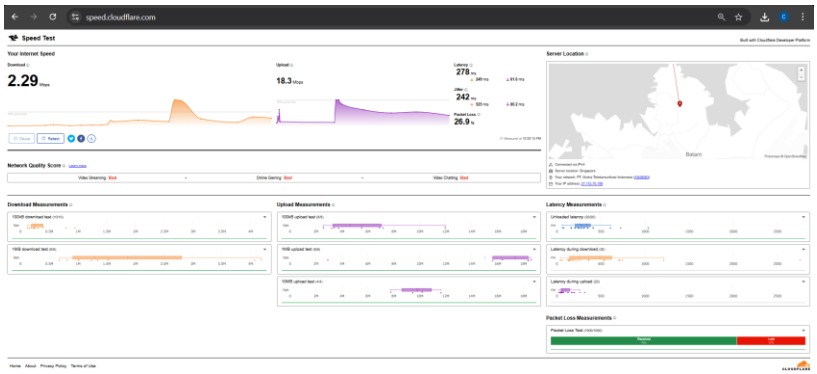
Kerusakan/penurunan kualitas jaringan terjadi sebelum traffic masuk ke tunnel Zero Trust—kemungkinan pada jaringan ISP atau pada akses internet perusahaan.



Gambar 4 Hasil Pengujian WARP dan ZTNA Tunnel

3.4 Halaman Hasil Pengujian Speed Test Cloudflare

Hasil pengujian menggunakan *Cloudflare Speed Test* menunjukkan bahwa kondisi jaringan berada pada kategori Buruk (Bad) untuk seluruh aktivitas kritis seperti video streaming, online gaming, dan video chatting. Kecepatan *download* hanya mencapai 2.29 Mbps, jauh di bawah standar layanan internet yang memadai, sementara *upload* 18.3 Mbps terlihat lebih stabil. Nilai *latency* sangat tinggi yaitu 278 ms, disertai jitter sebesar 242 ms, yang mengindikasikan ketidakstabilan jaringan pada jalur komunikasi. Selain itu, *packet loss* mencapai 26.9%, sebuah angka kritikal yang dapat menyebabkan kegagalan koneksi, buffering, hingga putusnya sesi komunikasi. Grafik pengukuran pada download dan upload menunjukkan fluktuasi besar, mencerminkan adanya hambatan pada sisi ISP atau jalur routing menuju server pengujian yang berlokasi di Singapore. Secara keseluruhan, hasil ini menegaskan bahwa performa jaringan sangat tidak stabil dan mengalami bottleneck, terutama pada jalur download dan kualitas sinyal, sehingga berdampak signifikan terhadap pengalaman pengguna serta menunjukkan bahwa permasalahan terjadi sebelum traffic diproses oleh mekanisme keamanan seperti ZTNA atau WARP.



Gambar 5 Halaman Hasil Pengujian Speed Test Cloudflare

4. KESIMPULAN

Penelitian ini berhasil membuktikan bahwa penerapan Zero Trust Network Architecture (ZTNA) mampu meningkatkan keamanan jaringan sesuai tujuan pada bab Pendahuluan. Mekanisme seperti DNS proxy, enkripsi traffic, autentikasi, dan tunneling berjalan dengan baik dan menunjukkan efektivitas ZTNA dalam memvalidasi serta mengamankan setiap akses. Namun, hasil pengujian juga menunjukkan bahwa performa jaringan sangat dipengaruhi oleh kualitas infrastruktur dasar, ditandai dengan latency dan packet loss yang tinggi pada jalur internet sebelum masuk ke tunnel Zero Trust. Karena itu, keberhasilan keamanan ZTNA perlu didukung oleh peningkatan stabilitas jaringan agar pengalaman pengguna tetap optimal. Penelitian selanjutnya dapat fokus pada optimasi jaringan dan pengembangan integrasi ZTNA yang lebih luas.

DAFTAR PUSTAKA

[1] A. Arogundade and O. Richard, “Network Security Concepts, Dangers, and Defense Best Practical,” Computer Engineering and Intelligent Systems, vol. 14, pp. 1–10, 2023.
[2] S. Aklani, H. Haeruddin, and N. Putri, “Implementasi Mail Gateway Security Dalam Meningkatkan Keamanan Email,” Journal of Information System Management (JOISM), vol. 5, no. 2, pp. 150–155, 2024.
[3] D. D. Silva, “Building a Zero Trust Architecture Using Kubernetes,” Int. Conf. for Convergence in Technology, pp. 1–8, 2021.
[4] F. Tang, C. Ma, and K. Cheng, “Privacy-preserving authentication scheme based on zero trust architecture,” Digital Communications and Networks, 2023.
[5] W. Yeoh, M. Liu, M. Shore, and F. Jiang, “Zero Trust Cybersecurity: Critical Success Factors and a Maturity Assessment Framework,” Computers & Security, vol. 133, p. 103412, 2023.
[6] R. M. Habash and M. K. Ibrahem, “Zero Trust Security Model for Enterprise Networks,” Iraqi Journal of Information and Communications Technology, vol. 6, no. 2, pp. 68–77, 2023.
[7] A. Suryo, “1,5 Terabite Data BSI Dicuri Geng Ransomware LockBit,” Detik.com, 2023.
[8] DetikInet, “5 Kebocoran Data di RI yang Diumbar Hacker Bjorka,” Detik.com, 2022.
[9] S. Adhar and U. Saprudin, “Implementasi Cloudflare Zero Trust dalam Mendeteksi Aktivitas Cryptojacking pada Jaringan Komputer,” Jurnal Teknologi Komputer dan Sistem Informasi, vol. 6, 2023.
[10] T. Dancheva et al., “Cloud Benchmarking and Performance Analysis of an HPC Application in Amazon EC2,” Cluster Computing, vol. 27, pp. 2273–2290, 2024.
[11] A. Pratama, L. Sugiyanta, and A. Idrus, “Design and Implementation of FreeRADIUS as a User Manager on the Mikrotik Hotspot Network Using the NDLC Method,” International Journal of Information System & Technology, vol. 7, pp. 136–143, 2023.