



Analisis Dampak Route Summarization Terhadap Performa OSPF Multi-Area Menggunakan GNS3 MikroTik

Muhammad Ilfan Asyuri*, Zainal Arifin, Mochammad Faid

Teknik, Teknologi Informasi, Universitas Nurul Jadid, Probolinggo, Indonesia

Email: ¹*ilfanhs88bp@gmail.com, ²zainal@unuja.ac.id, ³mfaid@unuja.ac.id

Email Penulis Korespondensi: ilfanhs88bp@gmail.com

Abstrak—Jaringan Open Shortest Path First (OSPF) multi-area pada skala menengah hingga besar menghadapi masalah pembengkakan Link State Database (LSDB) akibat propagasi Link State Advertisement (LSA) yang tidak terkendali, terutama saat setiap subnet diiklankan secara individual oleh Area Border Router (ABR) ke seluruh area jaringan. Kondisi ini meningkatkan konsumsi memori, beban kalkulasi algoritma Shortest Path First (SPF), dan keterlambatan konvergensi routing. Penelitian ini bertujuan menganalisis secara kuantitatif dampak penerapan route summarization pada ABR terhadap jumlah LSA per tipe, ukuran LSDB, entri routing table, waktu konvergensi, serta utilisasi CPU dan memori perangkat MikroTik RouterOS CHR dalam lingkungan OSPF multi-area. Metode yang digunakan adalah kuantitatif eksperimental dengan dua skenario terkontrol yakni tanpa route summarization sebagai baseline atau skenario 1 dan dengan route summarization pada ABR sebagai kondisi perlakuan atau skenario 2. Topologi dibangun pada platform simulasi GNS3 menggunakan 12 router MikroTik CHR yang tersebar pada tiga area dengan 24 subnet aktif dan 1.000 dummy prefix. Hasil penelitian menunjukkan bahwa route summarization mereduksi LSA Type-3 sebesar 99,71%, menurunkan ukuran LSDB sebesar 99,32%, mengurangi entri routing table sebesar 99,22%, menurunkan peak CPU ABR dari 11,33% menjadi 8,33%, serta menghasilkan isolasi kegagalan yang sangat efektif pada control plane Area 0 dengan rata-rata waktu konvergensi 0,2 ms dibandingkan 1.023 ms pada kondisi baseline. Penelitian ini memberikan kontribusi praktis sebagai panduan bagi administrator jaringan dalam mengoptimalkan penggunaan memori router serta mempercepat konvergensi pada arsitektur jaringan multi-area.

Kata Kunci: OSPF Multi-Area; Route Summarization; Area Border Router; Link State Advertisement; GNS3; MikroTik RouterOS CHR

Abstract—Open Shortest Path First (OSPF) multi-area networks at medium to large scales face the problem of Link State Database (LSDB) bloat caused by uncontrolled Link State Advertisement (LSA) propagation, particularly when each subnet is individually advertised by the Area Border Router (ABR) across all network areas. This condition increases memory consumption, the computational load of the Shortest Path First (SPF) algorithm, and routing convergence delays. This study aims to quantitatively analyze the impact of applying route summarization on the ABR regarding the number of LSAs per type, LSDB size, routing table entries, convergence time, and CPU and memory utilization of MikroTik RouterOS CHR devices in an OSPF multi-area environment. The method employed is quantitative experimental with two controlled scenarios: without route summarization as a baseline and with route summarization on the ABR as the treatment condition. The topology was built on the GNS3 simulation platform using 12 MikroTik CHR routers distributed across three areas with 24 active subnets and 1,000 dummy prefixes. Results show that route summarization reduced Type-3 LSAs by 99.71%, decreased LSDB size by 99.32%, reduced routing table entries by 99.22%, lowered peak ABR CPU usage from 11.33% to 8.33%, and achieved perfect fault isolation on the Area 0 control plane with an average convergence time of 0.2 ms compared to 1,023 ms in the baseline condition. This study contributes practical guidelines for network administrators to optimize router memory utilization and accelerate convergence in multi-area network architectures.

Keywords: OSPF Multi-Area; Route Summarization; Area Border Router; Link State Advertisement; GNS3; MikroTik RouterOS CHR

1. PENDAHULUAN

Pertumbuhan infrastruktur jaringan komputer pada institusi pendidikan, pemerintahan, dan korporasi skala menengah hingga besar mendorong meningkatnya kompleksitas pengelolaan rute secara signifikan. Kebutuhan terhadap mekanisme *routing* yang adaptif dan skalabel semakin mendesak seiring bertambahnya jumlah subnet dan layanan berbasis jaringan seperti *e-learning*, VoIP, dan sistem informasi terpusat (Wirya et al., 2025). Dalam konteks ini, pemilihan protokol *routing* yang tepat menjadi faktor kritis yang menentukan keandalan dan efisiensi operasional jaringan secara keseluruhan. *Routing* statis tidak lagi memadai karena ketidakmampuannya merespons perubahan topologi secara otomatis, sehingga protokol *routing* dinamis menjadi solusi yang lebih relevan (Irwan et al., 2025).

Di antara protokol *routing* dinamis yang tersedia, *Open Shortest Path First* (OSPF) menonjol sebagai protokol berbasis *link-state* yang paling banyak diimplementasikan pada jaringan skala *enterprise*. Berbeda dengan protokol *distance-vector* yang hanya menukar informasi jarak antar-router, OSPF membangun peta topologi jaringan yang lengkap melalui pertukaran *Link State Advertisement* (LSA) antar-router, kemudian menghitung jalur terpendek menggunakan algoritma Dijkstra. Kemampuan ini menjadikan OSPF lebih akurat dan responsif terhadap perubahan topologi dibandingkan protokol *distance-vector* seperti RIP (Shah & Kang, 2022). Untuk menangani jaringan berskala besar, OSPF mengadopsi arsitektur *multi-area* yang membagi domain *routing* menjadi beberapa area hierarkis dengan *Backbone Area* (Area 0) sebagai pusat (Moy, 1998). Dalam arsitektur ini, *Area Border Router* (ABR) bertanggung jawab mendistribusikan informasi topologi antar-area melalui *Summary LSA* (Type-3). Tanpa mekanisme pengendali, setiap subnet yang diiklankan secara individual oleh ABR menyebabkan pembengkakan *Link State Database* (LSDB), meningkatkan konsumsi memori, beban kalkulasi *Shortest Path First* (SPF), dan memperlambat konvergensi *routing* (Sembiring et al., 2025).



Salah satu mekanisme yang dirancang untuk mengatasi permasalahan tersebut adalah *route summarization*, yaitu teknik penggabungan sejumlah prefix jaringan menjadi satu prefix agregat sebelum diiklankan ke area lain melalui LSA Type-3. Dengan mengonfigurasi perintah *area range* pada ABR, seluruh subnet spesifik di suatu area dapat diringkas menjadi satu entri agregat tunggal yang dikirimkan ke area lain, sehingga jumlah LSA yang beredar berkurang secara drastis (MikroTik, 2025). Secara teoritis, mekanisme ini mampu menekan *overhead control plane*, mempercepat konvergensi, dan memberikan isolasi kegagalan yang lebih baik antar-area (Pan et al., 2024). Namun, validasi empiris terhadap dampak kuantitatifnya, khususnya pada platform MikroTik RouterOS CHR dalam lingkungan simulasi GNS3 masih sangat terbatas dalam literatur yang tersedia. Implementasi MikroTik Cloud Hosted Router (CHR) dalam lingkungan virtual saat ini semakin diminati karena menawarkan skalabilitas tinggi tanpa batasan perangkat keras fisik, sebagaimana dikonfirmasi dalam kajian kelayakan performa RouterOS MikroTik pada jaringan internet (F. P. E. Putra et al., 2024). Kendati demikian, arsitektur router virtual memiliki kerentanan khusus terhadap fluktuasi beban komputasi. Tanpa adanya kontrol propagasi LSA melalui *route summarization*, router virtual dapat mengalami saturasi pada *control plane* akibat membanjirnya paket pembaruan topologi. Hal ini akan memaksa *hypervisor* atau *host* untuk mengalokasikan siklus CPU secara berlebihan demi memfasilitasi algoritma *Shortest Path First* (SPF) yang berjalan berulang kali. Jika dibiarkan, kondisi ini berisiko menyebabkan *bottleneck* yang menjalar pada keseluruhan arsitektur jaringan. Mengingat signifikansi dampak tersebut, simulasi komprehensif menggunakan platform GNS3 menjadi langkah preventif yang mutlak diperlukan. Simulasi ini memungkinkan pemodelan stres kontrol lalu lintas secara presisi, sehingga batasan kinerja dan tingkat efisiensi alokasi sumber daya pada MikroTik CHR dapat terukur dengan jelas sebelum diimplementasikan secara nyata pada skala produksi (MikroTik, 2026; Sembiring et al., 2025).

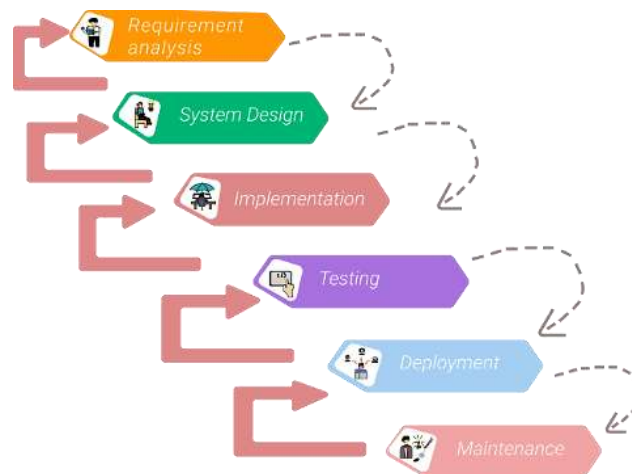
Berbagai penelitian terdahulu telah mengkaji OSPF dari beragam perspektif, namun masih menyisakan kesenjangan yang signifikan. Maulana mengkaji penerapan protokol OSPF pada router MikroTik dan analisis kinerjanya menggunakan *traffic generator tool*, namun tidak membahas analisis tipe LSA maupun pengaruh *route summarization* terhadap *overhead LSDB* (Maulana, 2024). Irwan et al. mengkaji implementasi *dynamic routing OSPF* pada lingkungan MikroTik dan jaringan institusi, namun tidak menyentuh mekanisme agregasi rute pada ABR (Irwan et al., 2025). Yahya et al. membandingkan OSPF *single-area* dan *multi-area* menggunakan Cisco Packet Tracer dan mengonfirmasi keunggulan arsitektur *multi-area* dari sisi skalabilitas, namun keterbatasan simulator membatasi kedalaman analisis internal protokol, khususnya terkait dinamika LSA dan perilaku LSDB (Yahya et al., 2024). Pratama et al. membandingkan kinerja routing OSPF *multi-area IPv4* dengan OSPFv3 *IPv6* pada MikroTik OS menggunakan simulasi GNS3, namun penelitian tersebut berfokus pada perbandingan versi protokol dan tidak mengkaji dampak agregasi rute pada ABR (Pratama et al., 2024). Huda dan Andriyani mengukur waktu konvergensi pada konfigurasi *Virtual Link* dan *GRE Tunnel* dalam topologi OSPF *multi-area*, namun tidak mengkaji pengaruh *route summarization* terhadap propagasi LSA secara spesifik (Huda & Andriyani, 2025). Shah dan Kang serta Soomro et al. memvalidasi GNS3 sebagai platform eksperimen yang andal untuk pengujian protokol *routing* dinamis termasuk OSPF, namun penelitian mereka tidak mencakup arsitektur OSPF *multi-area* dengan peran ABR dan mekanisme agregasi rute (Shah & Kang, 2022; Soomro et al., 2023). Aulia et al. mengimplementasikan Spanning Tree Protocol pada jaringan OSPF *multi-area* berbasis GNS3 untuk mengatasi routing loop, namun tidak mengevaluasi dampaknya terhadap beban LSA maupun waktu konvergensi ABR (Aulia et al., 2024). Sembiring et al. memvalidasi kombinasi GNS3 dan MikroTik CHR sebagai platform simulasi *routing* dinamis yang menghasilkan data konsisten, namun tidak mengkaji skenario *route summarization* secara eksplisit (Sembiring et al., 2025). Berdasarkan pemetaan tersebut, belum ada penelitian yang secara spesifik menganalisis dampak kuantitatif *route summarization* ABR terhadap jumlah LSA per tipe, ukuran LSDB, waktu konvergensi, serta konsumsi CPU secara bersamaan pada platform MikroTik RouterOS CHR dalam lingkungan GNS3.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan menganalisis secara kuantitatif dampak *route summarization* yang dikonfigurasi pada ABR dalam lingkungan OSPF *multi-area*, dengan menggunakan platform simulasi GNS3 berbasis MikroTik RouterOS CHR. Pengukuran dilakukan terhadap jumlah LSA per tipe, ukuran LSDB, entri *routing table*, waktu konvergensi pada skenario kegagalan topologi, serta konsumsi CPU dan memori perangkat dalam dua kondisi yang dikontrol, tanpa *route summarization* sebagai *baseline* dan dengan *route summarization* pada ABR sebagai kondisi perlakuan (MikroTik, 2025; Pan et al., 2024). Kontribusi utama penelitian ini adalah menghadirkan data kuantitatif yang komprehensif dan terverifikasi secara empiris mengenai efektivitas *route summarization* dalam menekan *overhead control plane* pada topologi OSPF *multi-area* berbasis MikroTik, yang selama ini belum tersedia dalam literatur yang ada. Hasil penelitian ini diharapkan dapat menjadi rujukan praktis bagi administrator jaringan dalam merancang dan mengoptimalkan topologi OSPF *multi-area* berskala menengah hingga besar, serta membuka arah penelitian lanjutan pada platform perangkat keras fisik maupun topologi dengan jumlah ABR yang lebih kompleks. Kontribusi utama dari penelitian ini adalah menyajikan analisis kuantitatif yang komprehensif mengenai efisiensi beban perangkat (CPU dan memori) serta reduksi tabel routing melalui teknik *route summarization* pada skenario *multi-area* secara riil.

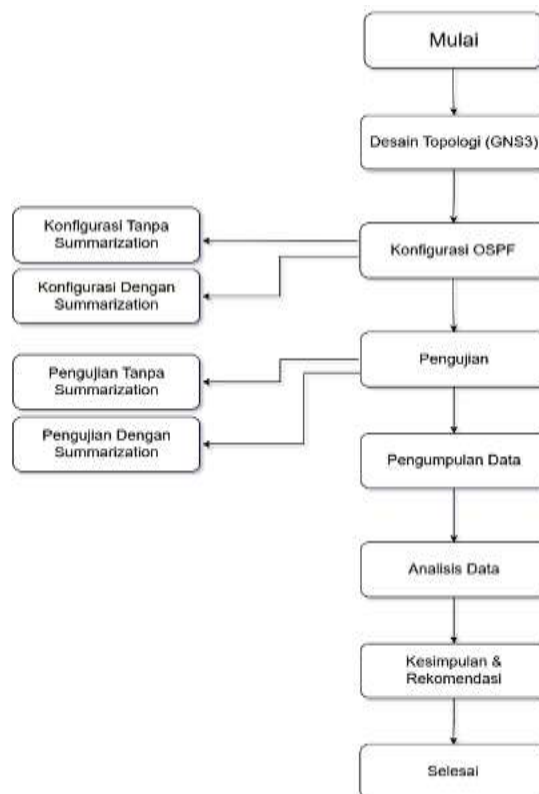
2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode kuantitatif dengan pendekatan eksperimental komputasional. Model pengembangan yang diadaptasi adalah model *Waterfall* yang berfokus pada tahapan kerja secara terstruktur dan sekuensial, mulai dari

desain topologi, konfigurasi, pengujian, pengumpulan data, analisis, hingga penarikan kesimpulan (A. P. Putra & Darwis, 2025). Pendekatan ini dipilih karena sangat relevan untuk perancangan dan evaluasi arsitektur jaringan, di mana setiap tahapan konfigurasi dan pengujian metrik performa harus dilakukan secara sistematis dan terkontrol.



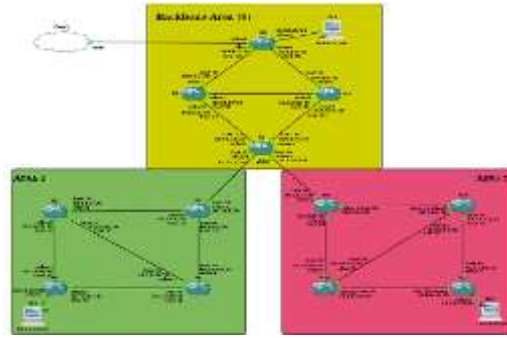
Gambar 1. Model Pengembangan *Waterfall*



Gambar 2. Langkah-Langkah Penelitian

2.1 Desain Topologi Jaringan

Arsitektur jaringan simulasi dirancang menggunakan topologi OSPF *multi-area* yang terdiri dari 12 node router MikroTik CHR (R1–R12) dan 3 node *Virtual PC Simulator* (VPCS) sebagai representasi klien *end-to-end*. Jaringan disegmentasi menjadi tiga area: *Backbone Area* (Area 0) sebagai pusat interkoneksi utama, Area 1, dan Area 2. Router R4 difungsikan sebagai ABR tunggal yang menjembatani pertukaran informasi *routing* dari Area 1 dan Area 2 menuju Area 0 (Yahya et al., 2024). Skema pengalamatan IPv4 dirancang secara hierarkis (*contiguous network*), Area 0 menggunakan blok 10.0.x.x/30, Area 1 menggunakan blok 10.1.x.x/30, dan Area 2 menggunakan blok 10.2.x.x/30. Pengelompokan alamat yang terstruktur ini merupakan prasyarat agar ABR dapat meringkas puluhan entri subnet spesifik menjadi satu *supernet* LSA Type-3 secara efektif (MikroTik, 2025). Tautan antar-router dirancang redundan dengan metrik *Cost* OSPF yang bervariasi secara manual (10, 20, dan 30) untuk menguji presisi algoritma Dijkstra dan memfasilitasi evaluasi perilaku konvergensi saat simulasi kegagalan tautan, sejalan dengan pendekatan berbasis *cost* untuk pemilihan jalur failover pada OSPF (Eislam et al., 2023).



Gambar 3. Desain Topologi OSPF *Multi-Area* pada GNS3

2.2 Skenario Konfigurasi dan Pengujian

Pengujian dilakukan melalui dua skenario komparatif yang dijalankan secara terpisah dan independen. Seluruh tautan OSPF antar-router dikonfigurasi menggunakan tipe jaringan *point-to-point* untuk mengeliminasi proses pemilihan *Designated Router* (DR) dan *Backup Designated Router* (BDR), sehingga konvergensi LSA berlangsung lebih presisi (Moy, 1998).

Untuk mensimulasikan beban komputasi jaringan berskala menengah-besar, diterapkan teknik *route injection* pada router R7 di Area 1. Sebanyak 1.000 alamat IPv4 statis (*dummy*) dari rentang 172.16.0.0–172.19.255.0/24 disuntikkan menggunakan *script* perulangan otomatis pada RouterOS, kemudian diiklankan ke dalam OSPF Area 1 (MikroTik, 2025). Adapun sintaks *script route injection* yang dieksekusi melalui antarmuka CLI MikroTik adalah sebagai berikut:

```
/interface bridge add name=loopback-dummy
:for i from=1 to=250 do={
  /ip address add address=("172.16." . $i . ".1/24") interface=loopback-dummy
  /ip address add address=("172.17." . $i . ".1/24") interface=loopback-dummy
  /ip address add address=("172.18." . $i . ".1/24") interface=loopback-dummy
  /ip address add address=("172.19." . $i . ".1/24") interface=loopback-dummy
}
/routing ospf network add network=172.16.0.0/14 area=area1
```

Pada Skenario 1 (*baseline*), ABR meneruskan seluruh 1.000 entri tersebut ke Area 0 dalam bentuk *Summary LSA* (Type-3) individual. Pada Skenario 2 (perlakuan), ABR dikonfigurasi menggunakan perintah *area range* untuk mengagregasi seluruh subnet tersebut menjadi satu prefix ringkasan 172.16.0.0/14 sebelum diteruskan ke Area 0, sehingga hanya satu LSA Type-3 tunggal yang diiklankan sebagai pengganti 1.000 LSA spesifik (Moy, 1998).

Di dalam masing-masing skenario, disimulasikan satu kejadian kegagalan tautan dengan menonaktifkan antarmuka *ether2* pada router R5 secara paksa. Tautan R5–R6 dipilih karena merepresentasikan jalur utama intra-area dengan metrik terendah menuju jaringan target, sehingga penonaktifannya memicu pergeseran rute ke jalur cadangan melalui router R8 secara deterministik. Setiap skenario dijalankan sebanyak 5 kali pengulangan independen, dan nilai yang dilaporkan merupakan rata-rata (*mean*) dari kelima pengulangan tersebut (Sembiring et al., 2025).

Pendekatan replikasi sebanyak lima kali pengulangan independen pada setiap skenario uji coba ini diadopsi secara khusus untuk meminimalisasi bias atau *margin of error* yang mungkin timbul akibat fluktuasi latensi internal dari lingkungan virtualisasi *host*. Dengan mengandalkan nilai rata-rata (*mean*) dari serangkaian iterasi tersebut, potensi anomali data yang bersifat transien dapat direduksi secara optimal. Metodologi pengumpulan data yang terstruktur ini memastikan bahwa hasil pengukuran, baik pada metrik waktu konvergensi maupun fluktuasi utilisasi prosesor ABR, benar-benar mencerminkan perilaku dan kinerja murni dari protokol OSPF secara objektif, bukan sekadar efek samping dari *overhead* sistem simulator (Sembiring et al., 2025).

2.3 Teknik Pengumpulan Data dan Parameter Uji

Pengumpulan data dilaksanakan menggunakan tiga instrumen utama. Pertama, metrik LSA dan LSDB diekstraksi melalui perintah *Command Line Interface* (CLI) RouterOS pada ABR (R4) dan router R1 sebagai representasi beban *routing* murni di Area 0. Kedua, waktu konvergensi *control plane* diukur menggunakan Wireshark yang di-*deploy* pada tautan R1–R2, difilter khusus untuk protokol OSPF. Waktu konvergensi dihitung berdasarkan selisih *timestamp* antara dua titik acuan sesuai RFC 4061 (Manral et al., 2005) dan RFC 6412 (Imhoff et al., 2011):

$$T_{konvergensi} = T_1 - T_0 \quad (1)$$

Di mana T_0 adalah *timestamp* paket LS *Update* pertama yang menyiarkan informasi kegagalan tautan (*Convergence Event Instant*), dan T_1 adalah *timestamp* paket LS *Acknowledge* terakhir yang mengkonfirmasi sinkronisasi LSDB secara utuh (Ari Hidayatulloh & Septiadi, 2025). Ketiga, utilisasi CPU dan memori pada ABR direkam menggunakan *script* AuditResource yang dikonfigurasi melakukan *polling* data setiap 10 ms selama periode



kegagalan tautan berlangsung. Resolusi 10 ms dipilih agar lonjakan CPU yang bersifat transien dapat tertangkap secara akurat. Implementasi *script* AuditResource yang dijalankan pada terminal RouterOS adalah sebagai berikut dan seluruh parameter uji yang dikumpulkan dirangkum pada Tabel 1.

```
/system script add name=AuditCPU source={
:log warning "--- START AUDIT CPU & MEMORY OSPF ---"
:for i from=1 to=500 do={
:local cpuLoad [/system resource get cpu-load]
:local freeMem ([/system resource get free-memory] / 1048576)
:if ($cpuLoad > 0) do={
:log info "Peak: CPU $cpuLoad % | Free Mem: $freeMem MB (Sampel ke-$i)"
}
:delay 10ms
}
:log warning "--- END AUDIT CPU & MEMORY OSPF ---"
}
```

Tabel 1. Parameter Uji

Parameter uji	Instrumen pengukuran	Lokasi pengukuran	Satuan
Jumlah LSA Type-1 Router LSA	RouterOS CLI	ABR	Entri
Jumlah LSA Type-3 Summary LSA	RouterOS CLI	ABR	Entri
Total entri Routing Table <i>OSPF</i>	RouterOS CLI	R1 (Area 0)	Entri
Ukuran <i>Link State Database</i> LSDB	RouterOS CLI	R1 (Area 0)	Entri
Waktu Konvergensi	Wireshark	Link R1-R2	Ms
Peak CPU ABR	Script AuditResource interval 10 ms	ABR	%
<i>Free Memory</i> ABR	Script AuditResource interval 10 ms	ABR	MB

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan analisis komparatif seluruh parameter yang diukur pada dua skenario konfigurasi: Skenario 1 sebagai kondisi *baseline* (tanpa *route summarization*) dan Skenario 2 sebagai kondisi perlakuan (dengan *route summarization* pada ABR). Setiap skenario dijalankan sebanyak 5 kali pengulangan independen, nilai yang dilaporkan adalah rata-rata (*mean*) dari kelima pengulangan uji coba tersebut kecuali dinyatakan lain.

3.1 Analisis Reduksi LSA dan Ukuran LSDB

Hasil pengukuran terhadap jumlah LSA, ukuran LSDB, dan entri *routing table* pada kedua skenario disajikan pada Tabel 2 berikut.

Tabel 2. Perbandingan Data LSA, LSDB, dan Routing Table

Parameter uji	Skenario 1	Skenario 2	Selisih penurunan	Persentase penurunan
Jumlah LSA Type-1	5	5	0	0
Jumlah LSA Type-3	1020	3	1017	99,71
Ukuran LSDB	1024	7	1017	99,32
Jumlah Entri Routing Table	1026	8	1018	99,22

Dari Tabel 2 dapat dilihat bahwa penerapan *route summarization* menghasilkan reduksi dramatis pada hampir seluruh parameter *control plane*. LSA Type-1 (*Router LSA*) tercatat identik pada kedua skenario (5 entri), hasil yang sepenuhnya konsisten dengan spesifikasi RFC 2328 (Moy, 1998) yang menetapkan bahwa Type-1 hanya beredar dalam area asalnya sehingga tidak terpengaruh oleh perintah *area range* pada ABR. Ketidak berubahnya nilai LSA Type-1 ini justru memvalidasi bahwa konfigurasi pengujian berjalan benar sesuai standar protokol, sekaligus menjadi kontrol internal yang mengonfirmasi bahwa isolasi antar-area OSPF berfungsi sebagaimana mestinya.

LSA Type-3 (*Summary LSA*) pada ABR (R4) turun drastis dari 1.020 entri menjadi 3 entri, merepresentasikan reduksi sebesar 99,71%. Angka 1.020 pada kondisi *baseline* terdiri dari 1.000 LSA yang berasal dari *dummy routes* di R7 ditambah entri subnet aktif Area 1 dan Area 2. Sementara itu, 3 entri yang tersisa pada skenario perlakuan merepresentasikan satu LSA Type-3 agregat 172.16.0.0/14 yang menggantikan seluruh 1.000 *dummy prefix*, ditambah sejumlah kecil LSA untuk subnet aktif yang tidak tercakup dalam blok agregasi (MikroTik, 2025; Moy, 1998). Fenomena ini membuktikan secara kuantitatif bahwa ABR yang dikonfigurasi *area range* beralih fungsi dari penerus informasi (*information relay*) menjadi pengagregasi topologi (*topology abstractor*) yang menyembunyikan kompleksitas internal Area 1 dari router-router di Area 0 (Pan et al., 2024).

Ukuran LSDB pada router R1 (Area 0) turun dari 1.024 entri menjadi 3 entri (99,32%), sementara jumlah entri *routing table* di R1 turun dari 1.026 menjadi 8 entri (99,22%). Selisih satu entri antara LSDB dan *routing table* pada skenario perlakuan merupakan kontribusi dari *route* agregat 172.16.0.0/14 sebagai entri *best-path* dalam tabel



forwarding, konsisten dengan perilaku standar OSPF di mana setiap entri LSDB yang berhasil dikomputasi Dijkstra menghasilkan tepat satu entri *routing table* (Moy, 1998). Penurunan 99,22% pada *routing table* secara langsung berimplikasi pada percepatan proses *route lookup* karena *longest prefix match* diselesaikan terhadap tabel yang jauh lebih kecil. Hasil ini juga selaras dengan temuan Pan et al. yang menyatakan bahwa reduksi ukuran LSDB secara proporsional menurunkan kompleksitas komputasional algoritma SPF, sehingga berpotensi mempercepat konvergensi secara keseluruhan (Pan et al., 2024).

Penurunan drastis ukuran LSDB ini tidak hanya berdampak pada satu siklus komputasi saat terjadi perubahan topologi, tetapi juga memberikan efisiensi jangka panjang pada operasional rutin jaringan. Sesuai dengan spesifikasi protokol OSPF, setiap LSA yang berada di dalam LSDB memiliki waktu hidup maksimum (*MaxAge*) dan secara *default* harus diperbarui (*refresh*) oleh router penghasilnya setiap 30 menit melalui pengiriman paket *Link State Update* secara periodik. Pada skenario *baseline* dengan 1.024 entri LSDB, proses pembaruan berkala ini menciptakan latar belakang lalu lintas (*background traffic*) yang konstan dan secara laten mengonsumsi *bandwidth* pada seluruh tautan antar-router, khususnya di dalam Area 0. Dengan diterapkannya *route summarization*, proses penyegaran LSA Type-3 berkurang secara eksponensial menjadi hanya 3 entri. Hal ini secara langsung mereduksi *overhead* pemeliharaan LSDB secara masif. Lebih jauh lagi, reduksi pada tabel *routing* meminimalisasi ukuran *Forwarding Information Base* (FIB) yang harus dimuat ke dalam memori *cache* router, mengoptimalkan proses *longest prefix match* (LPM) di tingkat per-paket, dan membebaskan kapasitas pemrosesan untuk tugas-tugas *firewall* atau *Network Address Translation* (NAT) yang umumnya dijalankan secara bersamaan pada arsitektur perangkat *edge* seperti MikroTik. Dampak sekunder dari efisiensi tabel *routing* ini juga terlihat pada proses pemecahan masalah (*troubleshooting*) sehari-hari, di mana tabel yang ringkas secara drastis mempercepat administrator dalam mengidentifikasi anomali rute melalui antarmuka konsol tanpa harus menyortir ribuan baris entri yang tidak relevan (MikroTik, 2025; Moy, 1998).

3.2 Evaluasi Utilisasi CPU pada Area Border Router

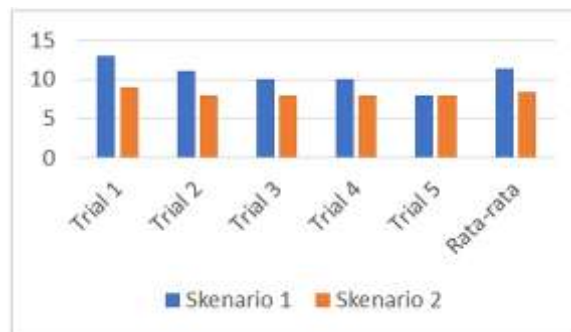
Hasil pengukuran *peak CPU* dan *free memory* ABR pada kedua skenario disajikan pada Tabel 3 dan Tabel 4 berikut.

Tabel 3. Data *Peak CPU* dan *Free Memory* ABR (R4) Tanpa *Summarization* (Skenario 1)

Percobaan/Trial	Peak load CPU (%)	Free memory saat peak (MB)
Trial 1	13	317
Trial 2	11	317
Trial 3	10	317
Trial 4	10	317
Trial 5	8	317
Rata-rata	11,33	317

Tabel 4. Data *Peak CPU* dan *Free Memory* ABR (R4) Dengan *Summarization* (Skenario 2)

Percobaan/Trial	Peak load CPU (%)	Free memory saat peak (MB)
Trial 1	9	317
Trial 2	8	317
Trial 3	8	317
Trial 4	8	317
Trial 5	8	317
Rata-rata	8,33	317



Gambar 4. Grafik Perbandingan Rata-Rata *Peak CPU* ABR antara Skenario 1 dan Skenario 2

Dari Tabel 3 dan Tabel 4 dapat dilihat perbandingan beban komputasi ABR antara kedua skenario secara menyeluruh. Rata-rata *peak CPU* pada kondisi *baseline* mencapai 11,33% dengan variasi antar-*trial* yang cukup terlihat (8%–13%). Lonjakan ini merepresentasikan beban gabungan tiga proses yang berlangsung secara bersamaan pada ABR: penerimaan dan validasi lebih dari 1.000 paket *LS Update* dari Area 1, pembangkitan ulang LSA Type-3 individual untuk setiap *prefix* yang diteruskan ke Area 0, dan pemicuan kalkulasi ulang algoritma Dijkstra akibat perubahan topologi yang diterima dari kedua sisi area (Sembiring et al., 2025).



Pada kondisi perlakuan, *peak* CPU turun menjadi rata-rata 8,33% dengan variasi yang jauh lebih stabil (8%–9%), mewakili penurunan sebesar 26,4% secara relatif. Penurunan ini bersumber dari simplifikasi fundamental tugas ABR alih-alih memproses 1.000 LSA Type-3 individual, ABR hanya perlu membangkitkan satu LSA Type-3 agregat tunggal, sehingga beban *flooding* dan kalkulasi ulang Dijkstra di sisi Area 0 berkurang drastis sebanding dengan pengurangan ukuran LSDB (Pan et al., 2024). Stabilitas variasi *peak* CPU yang lebih rendah pada skenario perlakuan (rentang hanya 1%) dibandingkan *baseline* (rentang 5%) juga mengindikasikan bahwa *route summarization* tidak hanya menurunkan beban puncak, tetapi juga membuat perilaku ABR lebih deterministik dan dapat diprediksi, yang merupakan keunggulan signifikan untuk keperluan *capacity planning* jaringan produksi.

Adapun parameter *free memory* menunjukkan nilai identik pada kedua skenario, yaitu statis pada 317 MB di seluruh 10 *trial*. Stabilitas ini disebabkan oleh perbedaan alokasi memori antar-skenario yang berada di bawah resolusi pengukuran satuan MB bulat dari perintah RouterOS, serta optimasi manajemen memori MikroTik CHR dalam lingkungan virtualisasi GNS3 (MikroTik, 2026; Sembiring et al., 2025). Meskipun perbedaan memori tidak terukur pada resolusi ini, secara teori pengurangan LSDB sebesar 99,32% tetap berimplikasi pada penghematan alokasi memori struktural yang signifikan dalam implementasi jaringan berskala lebih besar dengan jumlah *prefix* yang lebih tinggi.

Penting untuk ditekankan bahwa signifikansi dari penurunan *peak* CPU ABR dari 11,33% menjadi 8,33% ini harus dipahami dalam konteks beban kerja nyata sebuah router *enterprise*. Pada lingkungan produksi berskala menengah hingga besar, sebuah router sentral atau ABR jarang sekali didedikasikan secara eksklusif hanya untuk memproses protokol *routing* dinamis tunggal. Router sering kali dituntut untuk menjalankan layanan multi-fungsi secara serentak, termasuk pemfilteran paket melalui *firewall*, enkripsi *tunnel Virtual Private Network* (VPN) seperti IPsec, hingga manajemen *bandwidth* berbasis *Quality of Service* (QoS). Jika proses kalkulasi algoritma *Shortest Path First* (SPF) yang intensif memonopoli siklus CPU setiap kali terjadi kepankan tautan (*link flapping*) di area manapun, maka layanan lain akan mengalami degradasi kinerja atau *jitter* yang parah. Oleh karena itu, penghematan rata-rata 3% utilisasi CPU beserta eliminasi lonjakan (*spike*) pemrosesan yang dicapai melalui teknik *route summarization* ini memberikan *headroom* atau ruang napas komputasi yang sangat krusial. Ruang sisa ini memastikan router tetap responsif dalam melayani interupsi perangkat keras maupun interaksi antarmuka administrasi, bahkan pada saat jaringan sedang berada dalam fase transisi topologi yang paling tidak stabil. Terlebih lagi, stabilitas utilisasi ini secara tidak langsung memperpanjang usia pakai (*lifespan*) dari perangkat fisik jika diimplementasikan di luar lingkungan virtual, karena dapat mencegah beban termal yang berlebihan pada prosesor utama (MikroTik, 2025; Pan et al., 2024).

3.3 Analisis Waktu Konvergensi dan Isolasi Kegagalan

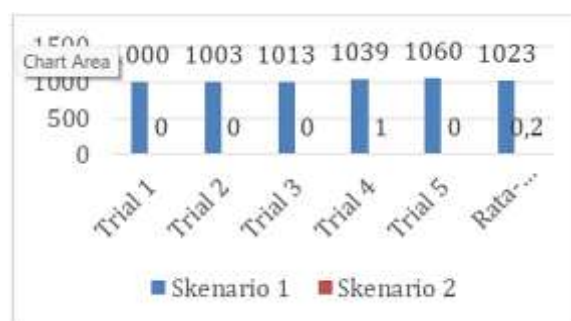
Hasil pengukuran waktu konvergensi *control plane* pada kedua skenario disajikan pada Tabel 5 dan Tabel 6 berikut.

Tabel 5. Data Waktu Konvergensi *Control Plane* Tanpa *Summarization* (Skenario 1)

Percobaan/Trial	T0	T1	Waktu konvergensi (ms)
Trial 1	16409	17409	1000
Trial 2	20734	21737	1003
Trial 3	16749	17807	1013
Trial 4	20130	21169	1039
Trial 5	17484	18543	1060
Rata-rata			1023

Tabel 6. Data Waktu Konvergensi *Control Plane* Dengan *Summarization* (Skenario 2)

Percobaan/Trial	T0	T1	Waktu konvergensi (ms)
Trial 1	38613	38613	0
Trial 2	16891	16891	0
Trial 3	19938	19938	0
Trial 4	30109	30110	1
Trial 5	22852	22852	0
Rata-rata			0,2



Gambar 5. Grafik Perbandingan Waktu Konvergensi *Control Plane* antara Skenario 1 dan Skenario 2



Dari Tabel 5 dapat dilihat bahwa pada skenario *baseline*, waktu konvergensi *control plane* di Area 0 mencapai rata-rata 1.023 ms dengan rentang 1.000–1.060 ms. Durasi ini merepresentasikan total siklus konvergensi OSPF yang dimulai dari penerimaan paket *LS Update* pertama pembawa informasi kegagalan tautan R5–R6, berlanjut pada *flooding* LSA ke seluruh Area 0, hingga terbitnya paket *LS Acknowledge* terakhir yang mengkonfirmasi bahwa seluruh router telah memproses pembaruan topologi secara utuh (Imhoff et al., 2011; Manral et al., 2005). Durasi lebih dari satu detik ini merupakan konsekuensi langsung dari ukuran LSDB sebesar 1.024 entri yang harus dievaluasi ulang oleh algoritma Dijkstra dengan kompleksitas komputasional $O(E \log V)$ (Pan et al., 2024). Temuan ini konsisten dengan penelitian Huda dan Andriyani yang mengonfirmasi bahwa ukuran LSDB berkorelasi positif dengan durasi konvergensi pada topologi OSPF *multi-area* (Huda & Andriyani, 2025). Temuan ini sejalan dengan studi komparatif protokol link-state lainnya yang juga mengonfirmasi keunggulan OSPF dalam kecepatan konvergensi pada topologi jaringan bersegmentasi area menggunakan simulasi GNS3 (Shahid et al., 2024).

Dari Tabel 6 dapat dilihat bahwa pada skenario perlakuan, sebanyak 4 dari 5 *trial* menghasilkan waktu konvergensi 0 ms ($T_0 = T_1$) dan 1 *trial* menghasilkan 1 ms, dengan rata-rata 0,2 ms. Nilai ini bukan berarti konvergensi terjadi secara instan secara fisik, melainkan merupakan manifestasi mekanisme isolasi kegagalan (*fault isolation*) yang merupakan properti fundamental arsitektur OSPF *multi-area* (Moy, 1998; Pan et al., 2024). Ketika tautan R5–R6 putus, informasi kegagalan menyebar sebagai *Router LSA* (Type-1) hanya di dalam Area 1. Karena kegagalan tersebut tidak mengubah *prefix* yang diiklankan ke Area 0, agregat 172.16.0.0/14 masih *reachable* melalui jalur alternatif di Area 1. Akibatnya, hanya terjadi satu paket *LS Update* yang dikirim ke Area 0 sebagai notifikasi perubahan state tautan, tanpa memicu *flooding* massal maupun kalkulasi ulang Dijkstra pada router-router Area 0 termasuk R1. Perbedaan hasil antara skenario 1 dan skenario 2 sebesar 1.022,8 ms ini secara empiris mengonfirmasi bahwa *route summarization* bukan sekadar optimasi administratif, melainkan mekanisme arsitektural yang secara fundamental mengubah perilaku propagasi kegagalan pada *control plane*.

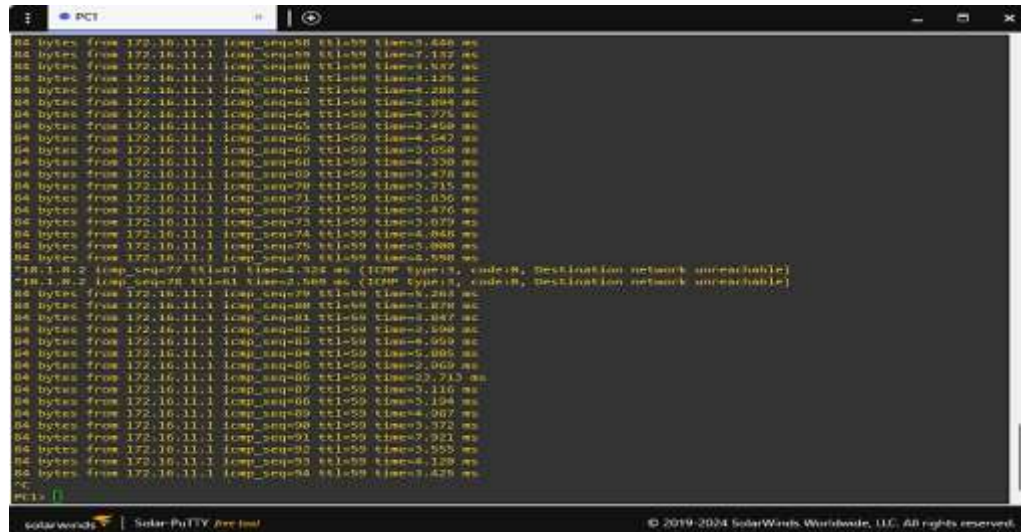
3.4 Pembahasan

Secara keseluruhan, hasil penelitian ini mengonfirmasi secara kuantitatif dan konsisten bahwa *route summarization* pada ABR memberikan dampak transformatif terhadap efisiensi *control plane* OSPF *multi-area*. Reduksi LSA Type-3 sebesar 99,71% dan LSDB sebesar 99,32% yang ditemukan dalam penelitian ini jauh melampaui estimasi teoritis yang umumnya disebutkan dalam literatur jaringan, dan secara empiris memvalidasi efektivitas mekanisme *area range* pada implementasi MikroTik RouterOS CHR (MikroTik, 2026).

Hasil ini memperkuat dan memperluas temuan Yahya et al. yang mengidentifikasi keunggulan topologi *multi-area* dibandingkan *single-area* dalam hal skalabilitas, namun tidak mengukur dampak kuantitatif *route summarization* secara spesifik (Yahya et al., 2024). Dibandingkan dengan Huda dan Andriyani yang fokus pada waktu konvergensi melalui mekanisme *Virtual Link* dan *GRE Tunnel*, penelitian ini menghadirkan perspektif yang berbeda dengan menunjukkan bahwa isolasi kegagalan melalui *route summarization* dapat mereduksi konvergensi *control plane* Area 0 secara hampir sempurna tanpa perlu mekanisme tambahan (Huda & Andriyani, 2025). Sementara itu, validasi platform GNS3 yang dilakukan oleh Shah dan Kang serta Soomro et al. memberikan landasan metodologis yang kuat bahwa data yang diperoleh dalam penelitian ini mencerminkan perilaku protokol yang realistis dan dapat dipercaya (Shah & Kang, 2022; Soomro et al., 2023).

Penting untuk dicatat bahwa meskipun *control plane* Area 0 tidak terinterupsi, transmisi *Ping* dari PC1 (Area 0) menuju R7 (Area 1) tetap mengalami sejumlah respons *Request Time Out* (RTO) pada periode sesaat setelah kegagalan tautan.

```
00 bytes from 172.16.10.1: icmp_seq=1 ttl=64 time=0.100 ms
00 bytes from 172.16.10.1: icmp_seq=2 ttl=64 time=0.110 ms
00 bytes from 172.16.10.1: icmp_seq=3 ttl=64 time=0.120 ms
00 bytes from 172.16.10.1: icmp_seq=4 ttl=64 time=0.130 ms
00 bytes from 172.16.10.1: icmp_seq=5 ttl=64 time=0.140 ms
00 bytes from 172.16.10.1: icmp_seq=6 ttl=64 time=0.150 ms
00 bytes from 172.16.10.1: icmp_seq=7 ttl=64 time=0.160 ms
00 bytes from 172.16.10.1: icmp_seq=8 ttl=64 time=0.170 ms
00 bytes from 172.16.10.1: icmp_seq=9 ttl=64 time=0.180 ms
00 bytes from 172.16.10.1: icmp_seq=10 ttl=64 time=0.190 ms
00 bytes from 172.16.10.1: icmp_seq=11 ttl=64 time=0.200 ms
00 bytes from 172.16.10.1: icmp_seq=12 ttl=64 time=0.210 ms
00 bytes from 172.16.10.1: icmp_seq=13 ttl=64 time=0.220 ms
00 bytes from 172.16.10.1: icmp_seq=14 ttl=64 time=0.230 ms
00 bytes from 172.16.10.1: icmp_seq=15 ttl=64 time=0.240 ms
00 bytes from 172.16.10.1: icmp_seq=16 ttl=64 time=0.250 ms
00 bytes from 172.16.10.1: icmp_seq=17 ttl=64 time=0.260 ms
00 bytes from 172.16.10.1: icmp_seq=18 ttl=64 time=0.270 ms
00 bytes from 172.16.10.1: icmp_seq=19 ttl=64 time=0.280 ms
00 bytes from 172.16.10.1: icmp_seq=20 ttl=64 time=0.290 ms
00 bytes from 172.16.10.1: icmp_seq=21 ttl=64 time=0.300 ms
00 bytes from 172.16.10.1: icmp_seq=22 ttl=64 time=0.310 ms
00 bytes from 172.16.10.1: icmp_seq=23 ttl=64 time=0.320 ms
00 bytes from 172.16.10.1: icmp_seq=24 ttl=64 time=0.330 ms
00 bytes from 172.16.10.1: icmp_seq=25 ttl=64 time=0.340 ms
00 bytes from 172.16.10.1: icmp_seq=26 ttl=64 time=0.350 ms
00 bytes from 172.16.10.1: icmp_seq=27 ttl=64 time=0.360 ms
00 bytes from 172.16.10.1: icmp_seq=28 ttl=64 time=0.370 ms
00 bytes from 172.16.10.1: icmp_seq=29 ttl=64 time=0.380 ms
00 bytes from 172.16.10.1: icmp_seq=30 ttl=64 time=0.390 ms
00 bytes from 172.16.10.1: icmp_seq=31 ttl=64 time=0.400 ms
00 bytes from 172.16.10.1: icmp_seq=32 ttl=64 time=0.410 ms
00 bytes from 172.16.10.1: icmp_seq=33 ttl=64 time=0.420 ms
00 bytes from 172.16.10.1: icmp_seq=34 ttl=64 time=0.430 ms
00 bytes from 172.16.10.1: icmp_seq=35 ttl=64 time=0.440 ms
00 bytes from 172.16.10.1: icmp_seq=36 ttl=64 time=0.450 ms
00 bytes from 172.16.10.1: icmp_seq=37 ttl=64 time=0.460 ms
00 bytes from 172.16.10.1: icmp_seq=38 ttl=64 time=0.470 ms
00 bytes from 172.16.10.1: icmp_seq=39 ttl=64 time=0.480 ms
00 bytes from 172.16.10.1: icmp_seq=40 ttl=64 time=0.490 ms
00 bytes from 172.16.10.1: icmp_seq=41 ttl=64 time=0.500 ms
00 bytes from 172.16.10.1: icmp_seq=42 ttl=64 time=0.510 ms
00 bytes from 172.16.10.1: icmp_seq=43 ttl=64 time=0.520 ms
00 bytes from 172.16.10.1: icmp_seq=44 ttl=64 time=0.530 ms
00 bytes from 172.16.10.1: icmp_seq=45 ttl=64 time=0.540 ms
00 bytes from 172.16.10.1: icmp_seq=46 ttl=64 time=0.550 ms
00 bytes from 172.16.10.1: icmp_seq=47 ttl=64 time=0.560 ms
00 bytes from 172.16.10.1: icmp_seq=48 ttl=64 time=0.570 ms
00 bytes from 172.16.10.1: icmp_seq=49 ttl=64 time=0.580 ms
00 bytes from 172.16.10.1: icmp_seq=50 ttl=64 time=0.590 ms
00 bytes from 172.16.10.1: icmp_seq=51 ttl=64 time=0.600 ms
00 bytes from 172.16.10.1: icmp_seq=52 ttl=64 time=0.610 ms
00 bytes from 172.16.10.1: icmp_seq=53 ttl=64 time=0.620 ms
00 bytes from 172.16.10.1: icmp_seq=54 ttl=64 time=0.630 ms
00 bytes from 172.16.10.1: icmp_seq=55 ttl=64 time=0.640 ms
00 bytes from 172.16.10.1: icmp_seq=56 ttl=64 time=0.650 ms
00 bytes from 172.16.10.1: icmp_seq=57 ttl=64 time=0.660 ms
00 bytes from 172.16.10.1: icmp_seq=58 ttl=64 time=0.670 ms
00 bytes from 172.16.10.1: icmp_seq=59 ttl=64 time=0.680 ms
00 bytes from 172.16.10.1: icmp_seq=60 ttl=64 time=0.690 ms
00 bytes from 172.16.10.1: icmp_seq=61 ttl=64 time=0.700 ms
00 bytes from 172.16.10.1: icmp_seq=62 ttl=64 time=0.710 ms
00 bytes from 172.16.10.1: icmp_seq=63 ttl=64 time=0.720 ms
00 bytes from 172.16.10.1: icmp_seq=64 ttl=64 time=0.730 ms
00 bytes from 172.16.10.1: icmp_seq=65 ttl=64 time=0.740 ms
00 bytes from 172.16.10.1: icmp_seq=66 ttl=64 time=0.750 ms
00 bytes from 172.16.10.1: icmp_seq=67 ttl=64 time=0.760 ms
00 bytes from 172.16.10.1: icmp_seq=68 ttl=64 time=0.770 ms
00 bytes from 172.16.10.1: icmp_seq=69 ttl=64 time=0.780 ms
00 bytes from 172.16.10.1: icmp_seq=70 ttl=64 time=0.790 ms
00 bytes from 172.16.10.1: icmp_seq=71 ttl=64 time=0.800 ms
00 bytes from 172.16.10.1: icmp_seq=72 ttl=64 time=0.810 ms
00 bytes from 172.16.10.1: icmp_seq=73 ttl=64 time=0.820 ms
00 bytes from 172.16.10.1: icmp_seq=74 ttl=64 time=0.830 ms
00 bytes from 172.16.10.1: icmp_seq=75 ttl=64 time=0.840 ms
00 bytes from 172.16.10.1: icmp_seq=76 ttl=64 time=0.850 ms
00 bytes from 172.16.10.1: icmp_seq=77 ttl=64 time=0.860 ms
00 bytes from 172.16.10.1: icmp_seq=78 ttl=64 time=0.870 ms
00 bytes from 172.16.10.1: icmp_seq=79 ttl=64 time=0.880 ms
00 bytes from 172.16.10.1: icmp_seq=80 ttl=64 time=0.890 ms
00 bytes from 172.16.10.1: icmp_seq=81 ttl=64 time=0.900 ms
00 bytes from 172.16.10.1: icmp_seq=82 ttl=64 time=0.910 ms
00 bytes from 172.16.10.1: icmp_seq=83 ttl=64 time=0.920 ms
00 bytes from 172.16.10.1: icmp_seq=84 ttl=64 time=0.930 ms
00 bytes from 172.16.10.1: icmp_seq=85 ttl=64 time=0.940 ms
00 bytes from 172.16.10.1: icmp_seq=86 ttl=64 time=0.950 ms
00 bytes from 172.16.10.1: icmp_seq=87 ttl=64 time=0.960 ms
00 bytes from 172.16.10.1: icmp_seq=88 ttl=64 time=0.970 ms
00 bytes from 172.16.10.1: icmp_seq=89 ttl=64 time=0.980 ms
00 bytes from 172.16.10.1: icmp_seq=90 ttl=64 time=0.990 ms
00 bytes from 172.16.10.1: icmp_seq=91 ttl=64 time=1.000 ms
00 bytes from 172.16.10.1: icmp_seq=92 ttl=64 time=1.010 ms
00 bytes from 172.16.10.1: icmp_seq=93 ttl=64 time=1.020 ms
00 bytes from 172.16.10.1: icmp_seq=94 ttl=64 time=1.030 ms
00 bytes from 172.16.10.1: icmp_seq=95 ttl=64 time=1.040 ms
00 bytes from 172.16.10.1: icmp_seq=96 ttl=64 time=1.050 ms
00 bytes from 172.16.10.1: icmp_seq=97 ttl=64 time=1.060 ms
00 bytes from 172.16.10.1: icmp_seq=98 ttl=64 time=1.070 ms
00 bytes from 172.16.10.1: icmp_seq=99 ttl=64 time=1.080 ms
00 bytes from 172.16.10.1: icmp_seq=100 ttl=64 time=1.090 ms
00 bytes from 172.16.10.1: icmp_seq=101 ttl=64 time=1.100 ms
00 bytes from 172.16.10.1: icmp_seq=102 ttl=64 time=1.110 ms
00 bytes from 172.16.10.1: icmp_seq=103 ttl=64 time=1.120 ms
00 bytes from 172.16.10.1: icmp_seq=104 ttl=64 time=1.130 ms
00 bytes from 172.16.10.1: icmp_seq=105 ttl=64 time=1.140 ms
00 bytes from 172.16.10.1: icmp_seq=106 ttl=64 time=1.150 ms
00 bytes from 172.16.10.1: icmp_seq=107 ttl=64 time=1.160 ms
00 bytes from 172.16.10.1: icmp_seq=108 ttl=64 time=1.170 ms
00 bytes from 172.16.10.1: icmp_seq=109 ttl=64 time=1.180 ms
00 bytes from 172.16.10.1: icmp_seq=110 ttl=64 time=1.190 ms
00 bytes from 172.16.10.1: icmp_seq=111 ttl=64 time=1.200 ms
00 bytes from 172.16.10.1: icmp_seq=112 ttl=64 time=1.210 ms
00 bytes from 172.16.10.1: icmp_seq=113 ttl=64 time=1.220 ms
00 bytes from 172.16.10.1: icmp_seq=114 ttl=64 time=1.230 ms
00 bytes from 172.16.10.1: icmp_seq=115 ttl=64 time=1.240 ms
00 bytes from 172.16.10.1: icmp_seq=116 ttl=64 time=1.250 ms
00 bytes from 172.16.10.1: icmp_seq=117 ttl=64 time=1.260 ms
00 bytes from 172.16.10.1: icmp_seq=118 ttl=64 time=1.270 ms
00 bytes from 172.16.10.1: icmp_seq=119 ttl=64 time=1.280 ms
00 bytes from 172.16.10.1: icmp_seq=120 ttl=64 time=1.290 ms
00 bytes from 172.16.10.1: icmp_seq=121 ttl=64 time=1.300 ms
00 bytes from 172.16.10.1: icmp_seq=122 ttl=64 time=1.310 ms
00 bytes from 172.16.10.1: icmp_seq=123 ttl=64 time=1.320 ms
00 bytes from 172.16.10.1: icmp_seq=124 ttl=64 time=1.330 ms
00 bytes from 172.16.10.1: icmp_seq=125 ttl=64 time=1.340 ms
00 bytes from 172.16.10.1: icmp_seq=126 ttl=64 time=1.350 ms
00 bytes from 172.16.10.1: icmp_seq=127 ttl=64 time=1.360 ms
00 bytes from 172.16.10.1: icmp_seq=128 ttl=64 time=1.370 ms
00 bytes from 172.16.10.1: icmp_seq=129 ttl=64 time=1.380 ms
00 bytes from 172.16.10.1: icmp_seq=130 ttl=64 time=1.390 ms
00 bytes from 172.16.10.1: icmp_seq=131 ttl=64 time=1.400 ms
00 bytes from 172.16.10.1: icmp_seq=132 ttl=64 time=1.410 ms
00 bytes from 172.16.10.1: icmp_seq=133 ttl=64 time=1.420 ms
00 bytes from 172.16.10.1: icmp_seq=134 ttl=64 time=1.430 ms
00 bytes from 172.16.10.1: icmp_seq=135 ttl=64 time=1.440 ms
00 bytes from 172.16.10.1: icmp_seq=136 ttl=64 time=1.450 ms
00 bytes from 172.16.10.1: icmp_seq=137 ttl=64 time=1.460 ms
00 bytes from 172.16.10.1: icmp_seq=138 ttl=64 time=1.470 ms
00 bytes from 172.16.10.1: icmp_seq=139 ttl=64 time=1.480 ms
00 bytes from 172.16.10.1: icmp_seq=140 ttl=64 time=1.490 ms
00 bytes from 172.16.10.1: icmp_seq=141 ttl=64 time=1.500 ms
00 bytes from 172.16.10.1: icmp_seq=142 ttl=64 time=1.510 ms
00 bytes from 172.16.10.1: icmp_seq=143 ttl=64 time=1.520 ms
00 bytes from 172.16.10.1: icmp_seq=144 ttl=64 time=1.530 ms
00 bytes from 172.16.10.1: icmp_seq=145 ttl=64 time=1.540 ms
00 bytes from 172.16.10.1: icmp_seq=146 ttl=64 time=1.550 ms
00 bytes from 172.16.10.1: icmp_seq=147 ttl=64 time=1.560 ms
00 bytes from 172.16.10.1: icmp_seq=148 ttl=64 time=1.570 ms
00 bytes from 172.16.10.1: icmp_seq=149 ttl=64 time=1.580 ms
00 bytes from 172.16.10.1: icmp_seq=150 ttl=64 time=1.590 ms
00 bytes from 172.16.10.1: icmp_seq=151 ttl=64 time=1.600 ms
00 bytes from 172.16.10.1: icmp_seq=152 ttl=64 time=1.610 ms
00 bytes from 172.16.10.1: icmp_seq=153 ttl=64 time=1.620 ms
00 bytes from 172.16.10.1: icmp_seq=154 ttl=64 time=1.630 ms
00 bytes from 172.16.10.1: icmp_seq=155 ttl=64 time=1.640 ms
00 bytes from 172.16.10.1: icmp_seq=156 ttl=64 time=1.650 ms
00 bytes from 172.16.10.1: icmp_seq=157 ttl=64 time=1.660 ms
00 bytes from 172.16.10.1: icmp_seq=158 ttl=64 time=1.670 ms
00 bytes from 172.16.10.1: icmp_seq=159 ttl=64 time=1.680 ms
00 bytes from 172.16.10.1: icmp_seq=160 ttl=64 time=1.690 ms
00 bytes from 172.16.10.1: icmp_seq=161 ttl=64 time=1.700 ms
00 bytes from 172.16.10.1: icmp_seq=162 ttl=64 time=1.710 ms
00 bytes from 172.16.10.1: icmp_seq=163 ttl=64 time=1.720 ms
00 bytes from 172.16.10.1: icmp_seq=164 ttl=64 time=1.730 ms
00 bytes from 172.16.10.1: icmp_seq=165 ttl=64 time=1.740 ms
00 bytes from 172.16.10.1: icmp_seq=166 ttl=64 time=1.750 ms
00 bytes from 172.16.10.1: icmp_seq=167 ttl=64 time=1.760 ms
00 bytes from 172.16.10.1: icmp_seq=168 ttl=64 time=1.770 ms
00 bytes from 172.16.10.1: icmp_seq=169 ttl=64 time=1.780 ms
00 bytes from 172.16.10.1: icmp_seq=170 ttl=64 time=1.790 ms
00 bytes from 172.16.10.1: icmp_seq=171 ttl=64 time=1.800 ms
00 bytes from 172.16.10.1: icmp_seq=172 ttl=64 time=1.810 ms
00 bytes from 172.16.10.1: icmp_seq=173 ttl=64 time=1.820 ms
00 bytes from 172.16.10.1: icmp_seq=174 ttl=64 time=1.830 ms
00 bytes from 172.16.10.1: icmp_seq=175 ttl=64 time=1.840 ms
00 bytes from 172.16.10.1: icmp_seq=176 ttl=64 time=1.850 ms
00 bytes from 172.16.10.1: icmp_seq=177 ttl=64 time=1.860 ms
00 bytes from 172.16.10.1: icmp_seq=178 ttl=64 time=1.870 ms
00 bytes from 172.16.10.1: icmp_seq=179 ttl=64 time=1.880 ms
00 bytes from 172.16.10.1: icmp_seq=180 ttl=64 time=1.890 ms
00 bytes from 172.16.10.1: icmp_seq=181 ttl=64 time=1.900 ms
00 bytes from 172.16.10.1: icmp_seq=182 ttl=64 time=1.910 ms
00 bytes from 172.16.10.1: icmp_seq=183 ttl=64 time=1.920 ms
00 bytes from 172.16.10.1: icmp_seq=184 ttl=64 time=1.930 ms
00 bytes from 172.16.10.1: icmp_seq=185 ttl=64 time=1.940 ms
00 bytes from 172.16.10.1: icmp_seq=186 ttl=64 time=1.950 ms
00 bytes from 172.16.10.1: icmp_seq=187 ttl=64 time=1.960 ms
00 bytes from 172.16.10.1: icmp_seq=188 ttl=64 time=1.970 ms
00 bytes from 172.16.10.1: icmp_seq=189 ttl=64 time=1.980 ms
00 bytes from 172.16.10.1: icmp_seq=190 ttl=64 time=1.990 ms
00 bytes from 172.16.10.1: icmp_seq=191 ttl=64 time=2.000 ms
00 bytes from 172.16.10.1: icmp_seq=192 ttl=64 time=2.010 ms
00 bytes from 172.16.10.1: icmp_seq=193 ttl=64 time=2.020 ms
00 bytes from 172.16.10.1: icmp_seq=194 ttl=64 time=2.030 ms
00 bytes from 172.16.10.1: icmp_seq=195 ttl=64 time=2.040 ms
00 bytes from 172.16.10.1: icmp_seq=196 ttl=64 time=2.050 ms
00 bytes from 172.16.10.1: icmp_seq=197 ttl=64 time=2.060 ms
00 bytes from 172.16.10.1: icmp_seq=198 ttl=64 time=2.070 ms
00 bytes from 172.16.10.1: icmp_seq=199 ttl=64 time=2.080 ms
00 bytes from 172.16.10.1: icmp_seq=200 ttl=64 time=2.090 ms
00 bytes from 172.16.10.1: icmp_seq=201 ttl=64 time=2.100 ms
00 bytes from 172.16.10.1: icmp_seq=202 ttl=64 time=2.110 ms
00 bytes from 172.16.10.1: icmp_seq=203 ttl=64 time=2.120 ms
00 bytes from 172.16.10.1: icmp_seq=204 ttl=64 time=2.130 ms
00 bytes from 172.16.10.1: icmp_seq=205 ttl=64 time=2.140 ms
00 bytes from 172.16.10.1: icmp_seq=206 ttl=64 time=2.150 ms
00 bytes from 172.16.10.1: icmp_seq=207 ttl=64 time=2.160 ms
00 bytes from 172.16.10.1: icmp_seq=208 ttl=64 time=2.170 ms
00 bytes from 172.16.10.1: icmp_seq=209 ttl=64 time=2.180 ms
00 bytes from 172.16.10.1: icmp_seq=210 ttl=64 time=2.190 ms
00 bytes from 172.16.10.1: icmp_seq=211 ttl=64 time=2.200 ms
00 bytes from 172.16.10.1: icmp_seq=212 ttl=64 time=2.210 ms
00 bytes from 172.16.10.1: icmp_seq=213 ttl=64 time=2.220 ms
00 bytes from 172.16.10.1: icmp_seq=214 ttl=64 time=2.230 ms
00 bytes from 172.16.10.1: icmp_seq=215 ttl=64 time=2.240 ms
00 bytes from 172.16.10.1: icmp_seq=216 ttl=64 time=2.250 ms
00 bytes from 172.16.10.1: icmp_seq=217 ttl=64 time=2.260 ms
00 bytes from 172.16.10.1: icmp_seq=218 ttl=64 time=2.270 ms
00 bytes from 172.16.10.1: icmp_seq=219 ttl=64 time=2.280 ms
00 bytes from 172.16.10.1: icmp_seq=220 ttl=64 time=2.290 ms
00 bytes from 172.16.10.1: icmp_seq=221 ttl=64 time=2.300 ms
00 bytes from 172.16.10.1: icmp_seq=222 ttl=64 time=2.310 ms
00 bytes from 172.16.10.1: icmp_seq=223 ttl=64 time=2.320 ms
00 bytes from 172.16.10.1: icmp_seq=224 ttl=64 time=2.330 ms
00 bytes from 172.16.10.1: icmp_seq=225 ttl=64 time=2.340 ms
00 bytes from 172.16.10.1: icmp_seq=226 ttl=64 time=2.350 ms
00 bytes from 172.16.10.1: icmp_seq=227 ttl=64 time=2.360 ms
00 bytes from 172.16.10.1: icmp_seq=228 ttl=64 time=2.370 ms
00 bytes from 172.16.10.1: icmp_seq=229 ttl=64 time=2.380 ms
00 bytes from 172.16.10.1: icmp_seq=230 ttl=64 time=2.390 ms
00 bytes from 172.16.10.1: icmp_seq=231 ttl=64 time=2.400 ms
00 bytes from 172.16.10.1: icmp_seq=232 ttl=64 time=2.410 ms
00 bytes from 172.16.10.1: icmp_seq=233 ttl=64 time=2.420 ms
00 bytes from 172.16.10.1: icmp_seq=234 ttl=64 time=2.430 ms
00 bytes from 172.16.10.1: icmp_seq=235 ttl=64 time=2.440 ms
00 bytes from 172.16.10.1: icmp_seq=236 ttl=64 time=2.450 ms
00 bytes from 172.16.10.1: icmp_seq=237 ttl=64 time=2.460 ms
00 bytes from 172.16.10.1: icmp_seq=238 ttl=64 time=2.470 ms
00 bytes from 172.16.10.1: icmp_seq=239 ttl=64 time=2.480 ms
00 bytes from 172.16.10.1: icmp_seq=240 ttl=64 time=2.490 ms
00 bytes from 172.16.10.1: icmp_seq=241 ttl=64 time=2.500 ms
00 bytes from 172.16.10.1: icmp_seq=242 ttl=64 time=2.510 ms
00 bytes from 172.16.10.1: icmp_seq=243 ttl=64 time=2.520 ms
00 bytes from 172.16.10.1: icmp_seq=244 ttl=64 time=2.530 ms
00 bytes from 172.16.10.1: icmp_seq=245 ttl=64 time=2.540 ms
00 bytes from 172.16.10.1: icmp_seq=246 ttl=64 time=2.550 ms
00 bytes from 172.16.10.1: icmp_seq=247 ttl=64 time=2.560 ms
00 bytes from 172.16.10.1: icmp_seq=248 ttl=64 time=2.570 ms
00 bytes from 172.16.10.1: icmp_seq=249 ttl=64 time=2.580 ms
00 bytes from 172.16.10.1: icmp_seq=250 ttl=64 time=2.590 ms
00 bytes from 172.16.10.1: icmp_seq=251 ttl=64 time=2.600 ms
00 bytes from 172.16.10.1: icmp_seq=252 ttl=64 time=2.610 ms
00 bytes from 172.16.10.1: icmp_seq=253 ttl=64 time=2.620 ms
00 bytes from 172.16.10.1: icmp_seq=254 ttl=64 time=2.630 ms
00 bytes from 172.16.10.1: icmp_seq=255 ttl=64 time=2.640 ms
00 bytes from 172.16.10.1: icmp_seq=256 ttl=64 time=2.650 ms
00 bytes from 172.16.10.1: icmp_seq=257 ttl=64 time=2.660 ms
00 bytes from 172.16.10.1: icmp_seq=258 ttl=64 time=2.670 ms
00 bytes from 172.16.10.1: icmp_seq=259 ttl=64 time=2.680 ms
00 bytes from 172.16.10.1: icmp_seq=260 ttl=64 time=2.690 ms
00 bytes from 172.16.10.1: icmp_seq=261 ttl=64 time=2.700 ms
00 bytes from 172.16.10.1: icmp_seq=262 ttl=64 time=2.710 ms
00 bytes from 172.16.10.1: icmp_seq=263 ttl=64 time=2.720 ms
00 bytes from 172.16.10.1: icmp_seq=264 ttl=64 time=2.730 ms
00 bytes from 172.16.10.1: icmp_seq=265 ttl=64 time=2.740 ms
00 bytes from 172.16.10.1: icmp_seq=266 ttl=64 time=2.750 ms
00 bytes from 172.16.10.1: icmp_seq=267 ttl=64 time=2.760 ms
00 bytes from 172.16.10.1: icmp_seq=268 ttl=64 time=2.770 ms
00 bytes from 172.16.10.1: icmp_seq=269 ttl=64 time=2.780 ms
00 bytes from 172.16.10.1: icmp_seq=270 ttl=64 time=2.790 ms
00 bytes from 172.16.10.1: icmp_seq=271 ttl=64 time=2.800 ms
00 bytes from 172.16.10.1: icmp_seq=272 ttl=64 time=2.810 ms
00 bytes from 172.16.10.1: icmp_seq=273 ttl=64 time=2.820 ms
00 bytes from 172.16.10.1: icmp_seq=274 ttl=64 time=2.830 ms
00 bytes from 172.16.10.1: icmp_seq=275 ttl=64 time=2.840 ms
00 bytes from 172.16.10.1: icmp_seq=276 ttl=64 time=2.850 ms
00 bytes from 172.16.10.1: icmp_seq=277 ttl=64 time=2.860 ms
00 bytes from 172.16.10.1: icmp_seq=278 ttl=64 time=2.870 ms
00 bytes from 172.16.10.1: icmp_seq=279 ttl=64 time=2.880 ms
00 bytes from 172.16.10.1: icmp_seq=280 ttl=64 time=2.890 ms
00 bytes from 172.16.10.1: icmp_seq=281 ttl=64 time=2.900 ms
00 bytes from 172.16.10.1: icmp_seq=282 ttl=64 time=2.910 ms
00 bytes from 172.16.10.1: icmp_seq=283 ttl=64 time=2.920 ms
00 bytes from 172.16.10.1: icmp_seq=284 ttl=64 time=2.930 ms
00 bytes from 172.16.10.1: icmp_seq=285 ttl=64 time=2.940 ms
00 bytes from 172.16.10.1: icmp_seq=286 ttl=64 time=2.950 ms
00 bytes from 172.16.10.1: icmp_seq=287 ttl=64 time=2.960 ms
00 bytes from 172.16.10.1: icmp_seq=288 ttl=64 time=2.970 ms
00 bytes from 172.16.10.1: icmp_seq=289 ttl=64 time=2.980 ms
00 bytes from 172.16.10.1: icmp_seq=290 ttl=64 time=2.990 ms
00 bytes from 172.16.10.1: icmp_seq=291 ttl=64 time=3.000 ms
00 bytes from 172.16.10.1: icmp_seq=292 ttl=64 time=3.010 ms
00 bytes from 172.16.10.1: icmp_seq=293 ttl=64 time=3.020 ms
00 bytes from 172.16.10.1: icmp_seq=294 ttl=64 time=3.030 ms
00 bytes from 172.16.10.1: icmp_seq=295 ttl=64 time=3.040 ms
00 bytes from 172.16.10.1: icmp_seq=296 ttl=64 time
```



Gambar 7. Hasil Ping dari PC1 ke R7 pada Skenario Dengan Summarization

Gangguan *data plane* yang teramati pada Gambar 6 dan Gambar 7 ini tidak merepresentasikan kegagalan *route summarization* maupun konvergensi Area 0 yang lambat. Ketika tautan R5–R6 putus, paket *Ping* dari PC1 tetap diteruskan dengan benar oleh seluruh router di Area 0 menuju ABR, namun router-router di Area 1 (khususnya R5 dan tetangganya) sedang berada dalam periode kalkulasi ulang SPF intra-area sehingga tabel *routing* lokal mereka berada dalam kondisi tidak stabil. Selama periode transien ini paket yang tiba di Area 1 tidak dapat diteruskan dengan benar dan dikembalikan dengan pesan *unreachable* (Pan et al., 2024). Dengan demikian, RTO yang teramati adalah fenomena konvergensi *data-plane* lokal di Area 1 (*intra-area convergence delay*), yang secara konseptual terpisah dan independen dari konvergensi *control-plane* Area 0. Fakta bahwa PC1 menerima respons *Ping* yang sukses baik sebelum maupun setelah periode RTO mengkonfirmasi bahwa *backbone* Area 0 beroperasi penuh tanpa interupsi selama keseluruhan kejadian kegagalan berlangsung. Temuan ini secara empiris mengkonfirmasi proposisi teoritis bahwa OSPF *multi-area* dengan *route summarization* memberikan isolasi kegagalan yang efektif pada lapisan *control plane* Area 0, namun tidak mengubah durasi pemulihan intrinsik intra-area di area sumber kegagalan (Moy, 1998; Pan et al., 2024).

Pemisahan konseptual antara stabilitas *control plane* dan gangguan sesaat pada *data plane* ini merupakan indikator keberhasilan dari desain jaringan hierarkis. Dalam arsitektur OSPF tanpa agregasi rute, kegagalan tautan tunggal di tepi jaringan akan memicu efek domino berupa gelombang kejut pembaruan LSA yang memaksa seluruh router di dalam domain OSPF, dari ujung ke ujung, untuk menunda transmisi *data plane* mereka demi menyelesaikan kalkulasi ulang *control plane*. Sebaliknya, intervensi *route summarization* sukses membendung gelombang kejut tersebut hanya pada batas-batas Area 1. Peristiwa *Request Time Out* (RTO) sebanyak kurang lebih 4 hingga 5 paket ICMP yang terekam pada uji coba mewakili batas latensi fisik perutean intra-area yang tidak dapat dieliminasi secara murni oleh protokol OSPF, melainkan membutuhkan penyesuaian pada level perangkat keras seperti optimasi waktu sinkronisasi tabel *routing* ke tabel penerusan (*FIB update delay*). Dengan demikian, data empiris dari simulasi GNS3 ini memberikan bukti otentik bahwa konfigurasi *area range* pada MikroTik RouterOS bukan sekadar fitur sekunder yang bersifat opsional. Konfigurasi tersebut merupakan komponen arsitektural fundamental yang wajib diaktifkan sebagai *best practice* oleh para administrator jaringan. Langkah ini sangat esensial guna mencegah meluasnya domain kegagalan (*failure domain*) dan menjamin skalabilitas ketahanan arsitektur *enterprise* secara berkelanjutan seiring dengan bertambahnya *node* jaringan dan kompleksitas layanan di masa depan (Pan et al., 2024).

4. KESIMPULAN

Penelitian ini membuktikan secara kuantitatif bahwa penerapan *route summarization* pada ABR dalam lingkungan OSPF multi-area berbasis GNS3 dan MikroTik RouterOS CHR menghasilkan dampak yang sangat signifikan terhadap efisiensi *control plane*. Dari sisi reduksi LSA, LSA Type-3 tereduksi 99,71% dari 1.020 menjadi 3 entri, ukuran LSDB berkurang 99,32% dari 1.024 menjadi 7 entri, dan entri *routing table* turun 99,22% dari 1.026 menjadi 8 entri. Reduksi dramatis ini membuktikan bahwa ABR yang dikonfigurasi dengan perintah *area range* secara fundamental mengubah perannya dari penerus informasi individual menjadi pengagregasi topologi yang menyembunyikan kompleksitas internal area sumber dari seluruh router di area tujuan. Sementara itu, LSA Type-1 yang tetap identik di kedua skenario mengkonfirmasi bahwa mekanisme isolasi antar-area OSPF berjalan sesuai spesifikasi RFC 2328 (Moy, 1998). Dari sisi beban komputasi, *peak* CPU ABR menurun 26,4% secara relatif dari rata-rata 11,33% menjadi 8,33%, disertai stabilitas variasi yang jauh lebih baik, dari rentang 5% pada *baseline* menjadi hanya 1% pada kondisi perlakuan. Stabilitas ini mengindikasikan bahwa *route summarization* tidak hanya menurunkan beban puncak, tetapi juga membuat perilaku ABR lebih deterministik, yang merupakan keunggulan penting untuk keperluan *capacity planning* jaringan skala produksi. Adapun parameter *free memory* yang statis pada 317 MB di seluruh pengujian menunjukkan bahwa



perbedaan alokasi memori antar-skenario berada di bawah resolusi pengukuran platform virtualisasi yang digunakan, meskipun secara teori penghematan memori struktural tetap terjadi seiring berkurangnya entri LSDB. Lebih lanjut dari sisi konvergensi, isolasi kegagalan yang sangat efektif pada *control plane* Area 0 dicapai dengan rata-rata waktu konvergensi hanya 0,2 ms pada skenario perlakuan, dibandingkan 1.023 ms pada kondisi *baseline*. Perbedaan sebesar 1.022,8 ms ini mengkonfirmasi secara empiris bahwa *route summarization* merupakan mekanisme arsitektural yang secara fundamental mengubah perilaku propagasi kegagalan pada *control plane*, bukan sekadar optimasi administratif. Gangguan *data plane* berupa *Request Time Out* yang teramati pada periode transien pasca-kegagalan tautan merupakan fenomena konvergensi intra-area lokal di Area 1 yang terpisah dan independen dari konvergensi *control plane* Area 0, sehingga tidak mengurangi validitas hasil isolasi kegagalan yang dicapai. Berdasarkan seluruh temuan tersebut, disarankan agar penelitian lanjutan mengkaji dampak *route summarization* pada topologi yang lebih kompleks dengan jumlah ABR lebih dari satu, mengevaluasi perilaku protokol pada skenario kegagalan ganda (*multi-link failure*), serta memperluas pengujian ke platform perangkat keras fisik untuk memvalidasi konsistensi hasil simulasi GNS3 dengan kondisi jaringan produksi nyata.

REFERENCES

- Ari Hidayatulloh, B., & Septiadi, J. (2025). Desain Dan Analisis Kinerja Routing Protocol Ospfv3 : Single Area Vs Multi Area Pada Mikrotik Router-OS V6 Menggunakan Gns3. *Syntax : Journal of Software Engineering, Computer Science and Information Technology*, 6(1), 100–106. <https://doi.org/10.46576/syntax.v6i1.6820>
- Aulia, R., Risiko Liza, & Haida Dafitri. (2024). Analisis Routing Loop dalam Open Shortest Path First (OSPF) Routing Menggunakan Teknik Spanning Tree di Jaringan Multi Area. *Hello World Jurnal Ilmu Komputer*, 2(4), 158–168. <https://doi.org/10.56211/helloworld.v2i4.419>
- Eislam, C. M. R., Mulyana, D. I., & Akbar, Y. (2023). Implementasi Jaringan Failover Menggunakan OSPF Dengan Algoritme Dijkstra. *Progresif: Jurnal Ilmiah Komputer*, 19(1), 93. <https://doi.org/10.35889/progresif.v19i1.980>
- Huda, M., & Andriyani, W. (2025). Multi-Area OSPF Analysis Using Virtual Link and GRE Tunnel. *Eduvest - Journal of Universal Studies*, 5(2), 1804–1819. <https://doi.org/10.59188/eduvest.v5i2.1710>
- Imhoff, B., Poretsky, S., & Michielsen, K. (2011). Terminology for Benchmarking Link-State IGP Data-Plane Route Convergence. In *Rfc 6412*. <https://datatracker.ietf.org/doc/html/rfc6412>
- Irwani, I., Hafni, H., & Iqbal, M. (2025). Implementasi Dynamic Routing OSPF Menggunakan Mikrotik Dengan Metode Tunneling untuk Optimasi Konektivitas Jaringan Internet. *Jurnal Minfo Polgan*, 14(2), 3218–3229. <https://doi.org/10.33395/jmp.v14i2.15696>
- Manral, V., White, R., & Shaikh, A. (2005). RFC 4061 - Benchmarking Basic OSPF Single Router Control Plane Convergence. In *Ietf Rfc* (Vol. 1). <https://datatracker.ietf.org/doc/html/rfc4061>
- Maulana, A. (2024). Penerapan Protokol Routing Ospf Pada Router Mikrotik dan Analisa Kinerja Menggunakan Traffic Generator Tool (Studi Kasus : Munzir Kitchen Jakarta). *CONTEN : Computer and Network Technology*, 4(2), 135–146. <https://doi.org/10.31294/conten.v4i2.6185>
- MikroTik. (2025). *OSPF - RouterOS - MikroTik Documentation*. Help.Mikrotik.Com. <https://help.mikrotik.com/docs/spaces/ROS/pages/9863229/OSPF>
- MikroTik. (2026). *Mikrotik Cloud Hosted Router (CHR) Manual*. Help.Mikrotik.Com. <https://help.mikrotik.com/docs/spaces/ROS/pages/18350234/Cloud+Hosted+Router+CHR>
- Moy, J. (1998). *RFC2328 - OSPF Version 2*. Ascend Communications, Inc. <http://www.ietf.org/rfc/rfc2328.txt>
- Pan, C., Lu, H., Shi, H., Wang, Y., & Qin, L. (2024). Inverse Coupled Simulated Annealing for Enhanced OSPF Convergence in IoT Networks. *Electronics (Switzerland)*, 13(22), 4332. <https://doi.org/10.3390/electronics13224332>
- Pratama, A., Faisal, I., & Handoko, D. (2024). Analisis Perbandingan Kinerja Routing Ospf Multi Area IP Versi 4 dan Routing OSPF V3 IP Versi 6 Pada Mikrotik OS Simulasi GNS3. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 3(1), 42–48. <https://doi.org/10.70340/jirsi.v3i1.108>
- Putra, A. P., & Darwis, M. (2025). Perancangan Jaringan Sistem Smart Home berbasis IoT menggunakan Cisco Packet Tracer dengan Metode Waterfall. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 7(3), 410–418. <https://doi.org/10.47233/jteksis.v7i3.2015>
- Putra, F. P. E., Mufidah, K., Ilhamsyah, R. M., Efendy, S. A., & Barokah, S. N. R. (2024). Tinjauan Performa RouterOS Mikrotik dalam Jaringan Internet: Analisis Kinerja dan Kelayakan. *Digital Transformation Technology*, 3(2), 903–910. <https://doi.org/10.47709/digitech.v3i2.3446>
- Sembiring, F. G., Ashillah, S., Nasution, A. K., & Kiswanto, D. (2025). Analisis Kinerja Routing Dinamis Pada Jaringan Virtual Menggunakan Mikrotik Chr. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3). <https://doi.org/10.23960/jitet.v13i3.6516>
- Shah, A. N., & Kang, S. (2022). A Practical Analysis Report of Rip, EIGRP, And an OSPF Dynamic Routing Protocol using the Network Simulator Tool GNS-3. *International Journal of Distributed and Parallel Systems*, 13(5), 1–17. <https://doi.org/10.5121/ijdps.2022.13501>
- Shahid, K., Ahmad, S. N., & Rizvi, S. T. H. (2024). Optimizing Network Performance: A Comparative Analysis of EIGRP, OSPF, and BGP in IPv6-Based Load-Sharing and Link-Failover Systems. *Future Internet*, 16(9). <https://doi.org/10.3390/fi16090339>



- Soomro, A. M., Naeem, A. Bin, Ayub, F., Senapati, B., Ghafoor, M. I., & Bari, N. (2023). Performance Evaluation of Routing Protocol OSPF with GNS3. *Journal of Computing and Biomedical Informatics*, 5(1), 174–182. <https://doi.org/10.56979/501/2023>
- Wirya, M. G. S., Wawan, & Mahmudin. (2025). Perkembangan Teknologi Jaringan Komputer Dari PAN, LAN, MAN, WAN Hingga 5G. *JRIIN: Jurnal Riset Informatika Dan Inovasi*, 3(7), 1544–1548. <https://jurnalmahasiswa.com/index.php/jriin>
- Yahya, A., Erawan, A., Supriatna, D., Luthfia, G. M., Anwar, J., Hastuti, R. D., & Thooyibah, T. (2024). Komparasi Routing OSPF pada Single Area dan Multi Area Menggunakan Aplikasi Cisco Packet Tracer. *Digital Transformation Technology*, 4(1), 323–329. <https://doi.org/10.47709/digitech.v4i1.4031>