IMPLEMENTATION OF IP SLA AND POLICY-BASED ROUTING FOR FAILOVER IN A MULTI-HOMED NETWORK

IMPLEMENTASI IP SLA DAN *POLICY-BASED ROUTING* UNTUK *FAILOVER* PADA JARINGAN *MULTI-HOMED*

Dysan Yusak[1], Wiwin Sulistyo[2]
[1,2]Fakultas Teknologi Informasi Universitas Kristen Satya Wacana
Jl. Diponegoro No.52-60, Kec. Sidorejo, Kota Salatiga, Jawa Tengah
672020180@student.uksw.edu, wiwinsulistyo@uksw.edu,

Abstract *- In the current digital era, stable and uninterrupted internet connectivity is the backbone of modern operations, demanding high availability and reliability. A key challenge is managing the inherent single point of failure associated with a single ISP connection. This study investigates the implementation and effectiveness of IP Service Level Agreement (IP SLA) and Policy-Based Routing (PBR) on a multi-homed network to achieve automated failover optimization. A virtual testbed was constructed using EVE-NG with a Cisco IOS C7200 image, and each test scenario was replicated five times to ensure data consistency. IP SLA was configured for proactive failure detection (parameters: timeout 5000ms, threshold 500ms, request-data-size 100, delay up 10, delay down 5), while PBR was used for VLAN-based traffic steering. The results indicate that this solution successfully steered traffic according to policy and achieved a consistent failover switchover time of 1.07–1.09 seconds (n=5), supported by Wireshark analysis, which documented up to 100% packet loss during the transition. The failure detection time, directly correlated with the IP SLA frequency, varied from 3.96 seconds (5-second frequency) to 45.63 seconds (60-second frequency). During the transition, router CPU load remained low at 3%, indicating high resource efficiency. This research concludes that the combination of IP SLA and PBR is an effective solution for enhancing the resilience and service continuity of multi-homed networks with minimal computational overhead.*

Keywords *- IP SLA, Policy-Based Routing, Failover, Multi-homed Network, Traffic Management*

Abstrak - Jaringan internet telah menjadi fondasi operasi modern, menuntut ketersediaan dan keandalan yang tinggi. Tantangan utama terletak pada pengelolaan *single point of failure* yang melekat pada koneksi tunggal. Penelitian ini mengkaji efektivitas implementasi *IP Service Level Agreement* (IP SLA) dan *Policy-Based Routing* (PBR) pada jaringan *multi-homed* untuk optimalisasi *failover* otomatis. Lingkungan pengujian virtual dibangun menggunakan EVE-NG dengan *image* Cisco IOS C7200, dan setiap skenario direplikasi lima kali. IP SLA dikonfigurasi untuk deteksi proaktif (parameter: *timeout* 5000ms, *threshold* 500ms, *request-data-size* 100, *delay up* 10, *delay down* 5) sementara PBR mengarahkan lalu lintas berbasis VLAN. Hasil penelitian menunjukkan bahwa solusi ini berhasil mengarahkan lalu lintas sesuai kebijakan dengan waktu peralihan 1.07–1.09 detik (*n*=5), didukung oleh analisis *Wireshark* yang mencatat *packet loss* hingga 100% saat transisi. Waktu deteksi kegagalan, yang berkorelasi langsung dengan frekuensi IP SLA, bervariasi dari 3.96 detik (frekuensi 5 detik) hingga 45.63 detik (frekuensi 60 detik). Selama transisi, beban CPU *router* hanya mengalami lonjakan minimal hingga 3%, menunjukkan efisiensi sumber daya yang tinggi. Penelitian ini menyimpulkan bahwa kombinasi IP SLA dan PBR adalah solusi efektif untuk meningkatkan ketahanan dan kontinuitas layanan di lingkungan jaringan *multi-homed* dengan *overhead* komputasi yang minimal.

Kata Kunci - *IP SLA, Policy-Based Routing, Failover*, Jaringan *Multi-homed*, Manajemen trafik

I. PENDAHULUAN

Dalam era digital saat ini, konektivitas internet yang stabil dan tanpa henti menjadi pondasi utama bagi setiap kegiatan, baik bisnis maupun pribadi. Mulai dari operasional bisnis berbasis *cloud*, komunikasi *real-time*, hingga aktivitas sehari-hari seperti *streaming*, semuanya sangat bergantung pada akses internet yang andal [1]. Gangguan sekecil apa pun, bahkan yang berlangsung sesaat, dapat berdampak serius, mulai dari kerugian finansial, reputasi yang rusak, hingga penurunan produktivitas. Untuk meminimalkan risiko tersebut, strategi *multi-homing* yang menggunakan dua atau lebih koneksi internet dari penyedia yang berbeda telah menjadi pendekatan standar untuk menciptakan redundansi dan meningkatkan kapasitas [2].

Tantangan utamanya adalah bagaimana mengelola lalu lintas secara cerdas dan memastikan peralihan layanan (*failover*) yang cepat dan mulus saat salah satu koneksi terganggu. Metode *routing* tradisional, seperti yang digunakan oleh protokol seperti OSPF, sering kali hanya mengandalkan metrik biaya (*cost*) dan cenderung kurang responsif dalam skenario *failover* yang kritis [3]. Implementasi *multi-homing* secara umum juga sering mengandalkan pendekatan yang bervariasi; *Border Gateway Protocol* (BGP), meskipun menjadi standar industri untuk skala besar, memiliki kompleksitas dan waktu konvergensi yang bisa lambat [4]. Sementara itu, protokol redundansi seperti HSRP atau VRRP lebih fokus pada redundansi *first-hop* di dalam jaringan lokal, bukan pada *failover* antar ISP eksternal [5]. Pendekatan deteksi kegagalan seperti *Bidirectional Forwarding Detection* (BFD) menawarkan kecepatan *millisecond*, namun memerlukan dukungan dari *peer* dan terbatas pada deteksi *link* fisik, tidak seperti IP SLA yang menawarkan fleksibilitas pemantauan *end-to-end* [6].

Celah yang belum banyak dieksplorasi adalah mengoptimalkan *trade-off* antara kecepatan deteksi IP SLA dan biaya pemrosesan yang ditimbulkannya. Untuk menjawab kebutuhan ini, penelitian ini mengkaji secara mendalam kombinasi dua teknologi utama: *IP Service Level Agreement* (IP SLA) dan *Policy-Based Routing* (PBR). IP SLA berfungsi sebagai mekanisme deteksi kegagalan proaktif yang dapat secara otomatis memantau ketersediaan dan performa jalur jaringan, menjadikannya pemicu ideal untuk tindakan *failover* [7]. Sementara itu, PBR memungkinkan administrator jaringan untuk mengarahkan lalu lintas berdasarkan aturan spesifik, mengabaikan *routing* standar dan memberikan fleksibilitas tinggi dalam manajemen lalu lintas [8]. Kombinasi kedua teknologi ini berpotensi besar untuk menciptakan sistem *failover* yang sangat cepat, efisien, dan fleksibel, menjamin kontinuitas layanan yang tak terputus.

Penelitian ini secara khusus bertujuan untuk memberikan bukti empiris yang terukur tentang efektivitas solusi IP SLA dan PBR, dengan mengukur pengaruh frekuensi IP SLA terhadap waktu deteksi kegagalan dan biaya CPU router. Selain itu, penelitian ini juga membandingkan stabilitas *failover* berbasis PBR dengan pendekatan lain serta mengamati dampaknya terhadap *packet loss* dan *jitter* selama masa transisi. Berdasarkan celah penelitian di atas, penulis merumuskan dua hipotesis: (1) Peningkatan frekuensi IP SLA akan secara linier mempercepat waktu deteksi kegagalan dan waktu peralihan lalu lintas, namun hal ini akan menyebabkan peningkatan beban CPU pada *router* secara eksponensial. (2) Implementasi *failover* yang terintegrasi antara IP SLA dan PBR akan menghasilkan waktu peralihan lalu lintas yang lebih cepat dan lebih stabil (dengan *packet loss* dan *jitter* yang minimal) dibandingkan dengan solusi *floating static route* tradisional yang hanya mengandalkan deteksi kegagalan rute pasif.

II. SIGNIFIKANSI STUDI

A. Studi Literatur

Dalam konteks *modern*, *multi-homing* sangat penting bagi organisasi yang menuntut ketersediaan layanan tinggi dan kontinuitas akses, terutama pada infrastruktur data *center* dan jaringan perusahaan besar. Jaringan *multi-homed* adalah konfigurasi yang menghubungkan sebuah jaringan ke lebih dari satu penyedia layanan internet (ISP) secara simultan untuk meningkatkan keandalan serta menghilangkan titik kegagalan tunggal (*single point of failure*). Pendekatan *multi-homed* juga memperluas fleksibilitas dan keamanan jaringan terhadap gangguan dan serangan [9]. IP SLA merupakan fitur pada Cisco IOS yang memungkinkan pengawasan performa jaringan secara aktif. Dengan IP SLA, administrator jaringan dapat secara berkala mengukur berbagai parameter kualitas layanan seperti *latency*, *jitter* dan *packet loss*. Salah satu fungsi dasar IP SLA adalah *icmp-echo* (*ping)*, yang berfungsi untuk menguji keterjangkauan jalur (*path reachability*). IP SLA dapat diatur untuk menjalankan pemantauan secara otomatis dalam interval tertentu dan dapat mengirimkan notifikasi atau memicu mekanisme *failover* secara langsung saat mendeteksi anomali, misalnya peningkatan waktu respons di atas ambang batas yang ditetapkan atau kegagalan koneksi pada jalur utama. Dengan demikian, IP SLA berperan penting dalam mendeteksi kegagalan jaringan secara proaktif sehingga gangguan dapat segera diatasi sebelum berdampak negatif pada pengguna akhir [10]. *Policy Based Routing* (PBR) adalah sebuah metode manajemen jaringan yang memungkinkan administrator untuk mengarahkan lalu lintas data melalui jalur yang berbeda berdasarkan aturan spesifik, bukan hanya berdasarkan tujuan seperti *routing* tradisional. Penerapan PBR sangat efektif untuk mencapai *load balance* dan *failover* pada jaringan *multi*-ISP, di mana lalu lintas dari segmen jaringan yang berbeda dapat diarahkan ke ISP yang berbeda untuk menghindari penumpukan dan meningkatkan kualitas layanan. Keunggulan PBR adalah kemampuannya untuk mengalokasikan jalur internet secara spesifik dan fleksibel, meskipun kekurangannya adalah konfigurasinya yang manual dan adanya potensi *overload* jika tidak diatur dengan tepat. Namun, PBR terbukti dapat meningkatkan kualitas layanan internet secara signifikan dengan mengurangi *latency* dan waktu transmisi data [11], [12].

Meskipun terdapat berbagai pendekatan untuk mencapai redundansi dan *failover*, penelitian ini secara spesifik berfokus pada kombinasi IP SLA dan PBR karena pertimbangan praktis dan teknis. Protokol redundansi *first-hop* seperti HSRP atau VRRP efektif untuk menyediakan ketersediaan *gateway* di dalam jaringan lokal, namun tidak dirancang untuk mengelola *failover* antar koneksi ISP eksternal, yang merupakan tujuan utama dari *multi-homing* [13]. Di sisi lain, Bidirectional Forwarding Detection (BFD) menawarkan kecepatan deteksi kegagalan yang superior (dalam hitungan milidetik), namun sangat bergantung pada dukungan dari ISP dan terbatas pada deteksi kegagalan tautan fisik, bukan kualitas layanan *end-to-end* yang dapat dipantau oleh IP SLA [14]. Dengan demikian, IP SLA dan PBR menawarkan solusi yang ringan, mudah diimplementasikan, dan fleksibel untuk mendeteksi serta mengelola *failover* berbasis kebijakan tanpa memerlukan protokol *routing* dinamis yang kompleks. Evaluasi efektivitas solusi *failover* dalam jaringan sangat bergantung pada pengukuran metrik kinerja, termasuk waktu deteksi, waktu peralihan, dan beban CPU *router*. Pengukuran ini dianggap krusial untuk memahami seberapa cepat sistem dapat mengenali kegagalan dan beralih ke jalur cadangan, serta dampak solusi tersebut pada sumber daya perangkat [15].

B. Metode Penelitian

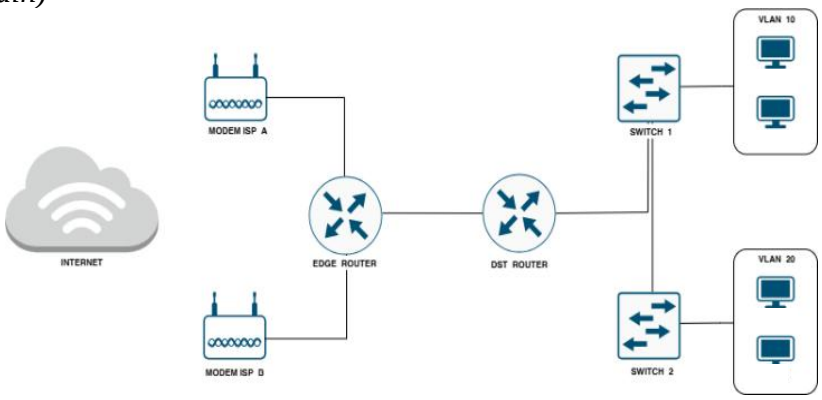
Penelitian ini akan dilakukan dengan menggunakan metode penelitian dari cisco yaitu PPDIOO (*Prepare, Plan, Design, Implement, Operate, Optimize*) untuk menyajikan kerangka kerja sistematis

dalam merancang, mengimplementasikan, menguji, dan mengevaluasi jaringan *multi-homed* berbasis PBR dan IP SLA. Adapun tahapan dari PPDIOO adalah sebagai berikut:

1) *Prepare (Persiapan),* fase ini berfokus pada penyiapan lingkungan pengujian *virtual*. Lab dibangun di EVE-NG, yang berjalan pada *VMware Workstation Player 17* dengan alokasi sumber daya perangkat keras 4 GB RAM dan 2 *core* CPU. Perangkat *virtual* yang digunakan meliputi *Cisco IOS C7200* versi 15.2(4)M12, *Cisco IOL Layer 2 Switch*, dan *Virtual* PC (VPC).

2) *Plan (Perencanaan),* tahap perencanaan mendefinisikan tujuan dan metrik pengujian. Diputuskan untuk menguji IP SLA dengan tiga frekuensi berbeda: 5, 30, dan 60 detik. Parameter IP SLA ditetapkan dengan *timeout* 5000ms, *threshold* 500ms, dan *request-datasize* 100 . Untuk mendukung *failover* dan *failback* yang efisien, parameter *delay* pada *track object* diatur dengan *delay up* 10 dan *delay down* 5. Pengujian konektivitas menggunakan *ping* dengan ukuran paket 100 *byte* pada *interval* 1 detik.

3) *Design (Desain)*



Gambar 1. Desain Awal Topologi Jaringan

Penjelasan topologi yang digunakan digambarkan pada Gambar 1, sebagai berikut:

a. Internet: Disimulasikan sebagai jaringan global yang dihubungkan melalui dua Penyedia Layanan Internet (ISP) yang berbeda.
b. Modem ISP A dan Modem ISP B: Memberikan koneksi terpisah ke EDGE ROUTER.
c. *Edge Router*: Sebagai *border router* yang mengimplementasikan IP SLA dan PBR.
d. *DST Router*: Berfungsi sebagai router distribusi atau *gateway* internal.
e. *Switch* 1 dan *Switch* 2: Kedua *switch* ini digunakan untuk segmentasi jaringan lokal.
f. *Vlan 10* dan *Vlan 20*: Sebagai segmentasi jaringan lokal (VLAN 10 dan VLAN 20).

4) *Implement (Implementasi),* membangun dan mengonfigurasi topologi jaringan virtual di EVE-NG sesuai desain. Pengaturan konektivitas dasar, PBR, IP SLA, *track object*, NAT, dan OSPF diterapkan pada perangkat Cisco. Verifikasi fungsionalitas dasar dilakukan.

5) *Operate (Operasi),* melakukan simulasi kegagalan ISP dengan mematikan *interface Edge_Router*. Data dikumpulkan menggunakan *ping* berkelanjutan, *log router*, *capture Wireshark*, dan pemantauan beban CPU. Pengujian diulang lima kali untuk setiap frekuensi IP SLA. Skenario *failback* juga diuji.

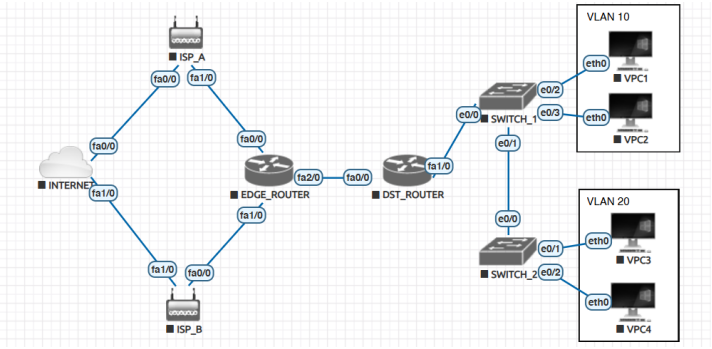
6) *Optimize (Optimasi),* menganalisis, data yang telah terkumpul secara mendalam untuk mengevaluasi kinerja sistem.

III. HASIL DAN PEMBAHSAN

Bagian ini menyajikan hasil implementasi, pengujian, dan pembahasan yang dilakukan dalam penelitian ini.

A. Implementasi Topologi dan Konfigurasi Jaringan

Penulis memulai dengan membangun topologi jaringan virtual di EVE-NG, yang dapat dilihat pada Gambar 2. Alokasi alamat IP untuk semua perangkat (*router*, *switch*, VPCS) telah ditetapkan pada Tabel I, memastikan setiap komponen terhubung dengan baik. Konfigurasi dasar, termasuk OSPF untuk *routing* internal, juga diimplementasikan pada *DST_Router* dan *Edge_Router*, serta segmentasi VLAN pada *switch* untuk mengidentifikasi lalu lintas. *Internet_Router* disimulasikan sebagai *server* eksternal (8.8.8.8) untuk pengujian konektivitas.



Gambar 2. Topologi Jaringan EVE_NG

TABEL I
ALOKASI IP ADDRESS

Perangkat	Interface/VLAN	IP Address/Subnet
Dst_Router	Fa1/0.10	192.168.10.1/24
	Fa1/0.20	192.168.20.1/24
	Fa0/0	192.168.1.2/30
Switch_1	-	-
Switch_2	-	-
Edge_Router	Fa0/0	100.100.100.1/30
	Fa1/0	200.200.200.1/30
	Fa2/0	192.168.1.1/30
ISP-A-Router	Fa0/0	10.10.10.1/30
	Fa1/0	100.100.100.2/30
ISP-B-Router	Fa1/0	10.10.20.1/30
	Fa0/0	200.200.200.2/30
Internet-Router	Loopback0	8.8.8.8/32
	Fa0/0	10.10.10.2/30
	Fa1/0	10.10.20.2/30
VPCS 1, VPC 2	VLAN10	DHCP
VPCS 3, VPC 4	VLAN20	DHCP

1. Konfigurasi IP SLA dan Objek Pelacakan (Track Object)

Pertama, penulis melakukan konfigurasi IP SLA pada perangkat *Edge_Router*. Pertama, menonfigurasi IP SLA 1 dengan perintah `ip sla 1`, kemudian mengatur operasi *icmp-echo* ke

tujuan 8.8.8.8 dengan mengetikkan *icmp-echo 8.8.8.8 source-interface Fa0/0* untuk ISP_A. Frekuensi pengiriman *echo* diatur setiap 5 detik menggunakan perintah *frequency 5*. *Request-data-size 100* digunakan menentukan ukuran *payload* data dari setiap paket *ping* yang dikirim. *Timeout 5000ms* menetapkan batas waktu tunggu respons *ping* hingga 5 detik. *Threshold 500ms*, menandai operasi tersebut sebagai gagal jika RTT dari paket icmp melebihi 500ms maka. Setelah konfigurasi selesai, penulis mengetikan perintah *ip sla schedule 1 life forever start-time now*, dimana IP SLA dijadwalkan aktif secara terus-menerus sejak diterapkan. Setelah IP SLA 1 aktif, mrngonfigurasi *object tracking* menggunakan perintah *track 1 ip sla 1 reachability* dan parameter *delay up 10* dan *delay down 5.* Jika IP SLA 1 gagal menjangkau 8.8.8.8 maka *object track 1* akan dianggap *down*, dan dapat digunakan untuk memicu peralihan rute secara otomatis ke ISP cadangan. Langkah serupa kemudian diulang untuk IP SLA 2, yang juga mengarah ke alamat 8.8.8.8 namun menggunakan antarmuka *FastEthernet0/1* untuk ISP_B, seperti yang dapat dilihat pada Gambar 3.

```
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#ip sla 1
EDGE_ROUTER(config-ip-sla)#icmp-echo 8.8.8.8 source-interface FastEthernet0/0
EDGE_ROUTER(config-ip-sla-echo)#frequency 5
EDGE_ROUTER(config-ip-sla-echo)#request-data-size 100
EDGE_ROUTER(config-ip-sla-echo)#threshold 500
EDGE_ROUTER(config-ip-sla-echo)#timeout 5000
EDGE_ROUTER(config-ip-sla-echo)#exit
EDGE_ROUTER(config)#ip sla schedule 1 life forever start-time now
EDGE_ROUTER(config-ip-sla-echo)#exit
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#ip sla 2
EDGE_ROUTER(config-ip-sla)#icmp-echo 8.8.8.8 source-interface FastEthernet1/0
EDGE_ROUTER(config-ip-sla-echo)#frequency 5
EDGE_ROUTER(config-ip-sla-echo)#request-data-size 100
EDGE_ROUTER(config-ip-sla-echo)#threshold 500
EDGE_ROUTER(config-ip-sla-echo)#timeout 5000
EDGE_ROUTER(config-ip-sla-echo)#exit
EDGE_ROUTER(config)#ip sla schedule 2 life forever start-time now
EDGE_ROUTER(config-ip-sla-echo)#exit
EDGE_ROUTER(config)#track 1 ip sla 1 reachability
EDGE_ROUTER(config-track)#delay up 10
EDGE_ROUTER(config-track)#delay down 5
EDGE_ROUTER(config-track)#exit
EDGE_ROUTER(config)#track 2 ip sla 2 reachability
EDGE_ROUTER(config-track)#delay up 10
EDGE_ROUTER(config-track)#delay down 5
EDGE_ROUTER(config-track)#exit
```

Gambar 3. Konfigurasi IP SLA

2. Access-List dan Route-Map untuk Policy-Based Routing (PBR) dengan Integrasi IP SLA

```
EDGE_ROUTER(config)#access-list 101 permit ip 192.168.10.0 0.0.0.255 any
EDGE_ROUTER(config)#access-list 102 permit ip 192.168.20.0 0.0.0.255 any
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#route-map PBR_TRAFFIC_STEERING permit 10
EDGE_ROUTER(config-route-map)#match ip address 101
EDGE_ROUTER(config-route-map)#set ip next-hop verify-availability 100.100.100.2 1 track 1
EDGE_ROUTER(config-route-map)#set ip default next-hop 200.200.200.2
EDGE_ROUTER(config-route-map)#exit
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#route-map PBR_TRAFFIC_STEERING permit 20
EDGE_ROUTER(config-route-map)#match ip address 102
EDGE_ROUTER(config-route-map)#set ip next-hop verify-availability 200.200.200.2 2 track 2
EDGE_ROUTER(config-route-map)#set ip default next-hop 100.100.100.2
EDGE_ROUTER(config-route-map)#exit
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#route-map PBR_TRAFFIC_STEERING permit 30
EDGE_ROUTER(config-route-map)#exit
```

Gambar 4. Konfigurasi PBR

Pada Gambar 4, Selanjutnya, penulis mengonfigurasi dua buah *access-lis*t yang didefinisikan: *Access-list 101 permit ip 192.168.10.0 0.0.0.255 any untuk* mengidentifikasi lalu lintas yang berasal dari *subnet 192.168.10.0/24* (VLAN 10), dan *Access-list 102 permit ip 192.168.20.0 0.0.0.255 any* untuk *subnet 192.168.20.0/24* (VLAN 20). ACL ini menjadi dasar dalam kebijakan *policy-based routing*.

a. Perintah *route-map PBR_TRAFFIC_STEERING permit 10* mencocokkan IP dari ACL 101 (VLAN 10) dengan perintah *match ip address 101*, lalu menerapkan kebijakan pengalihan ke next-hop 100.100.100.2 dengan perintah *set ip next-hop verify availability 100.100.100.2 1 track 1*, jika *track 1* menunjukkan bahwa koneksi ISP_A masih aktif. Jika tidak aktif, maka trafik akan diarahkan ke *next-hop* alternatif 200.200.200.2 (ISP_B) melalui perintah *set ip default next-hop 200.200.200.2*.
b. *Permit 20* mencocokkan IP dari ACL 102 (VLAN 20), dengan logika sebaliknya: trafik diarahkan ke next-hop 200.200.200.2 (ISP_B) jika *track 2* masih aktif, dan ke *next-hop 100.100.100.2* jika ISP_B *down*.

c. *Permit 30* dikonfigurasi sebagai urutan tambahan untuk menampung kebijakan lainnya di kemudian hari, meskipun belum didefinisikan secara spesifik.

3. Konfigurasi Floating Static Routes

```
EDGE_ROUTER(config)#ip route 0.0.0.0 0.0.0.0 100.100.100.2 track 1
EDGE_ROUTER(config)#ip route 0.0.0.0 0.0.0.0 200.200.200.2 100 track 2
```

Gambar 5. Konfigurasi *Floating Static Routes*

Sebagai pelengkap dari skema *failover* otomatis, dua jalur *default* dengan mekanisme *track* didefinisikan menggunakan perintah *ip route 0.0.0.0 0.0.0.0 100.100.100.2 track 1* dan *ip route 0.0.0.0 0.0.0.0 100.100.100.2 100 track 2* yang di konfigurasi penulis pada Gambar 5.

4. Konfigurasi NAT

```
EDGE_ROUTER(config)#access-list 103 permit ip 192.168.10.0 0.0.0.255 any
EDGE_ROUTER(config)#access-list 103 permit ip 192.168.20.0 0.0.0.255 any
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#route-map ISP_A permit 10
EDGE_ROUTER(config-route-map)#match ip address 103
EDGE_ROUTER(config-route-map)#match interface fastethernet 0/0
EDGE_ROUTER(config-route-map)#exit
EDGE_ROUTER(config)#
EDGE_ROUTER(config)#route-map ISP_B permit 10
EDGE_ROUTER(config-route-map)#match ip address 103
EDGE_ROUTER(config-route-map)#match interface fastethernet 1/0
EDGE_ROUTER(config-route-map)#exit
```

Gambar 6. Konfigurasi NAT

Pada Gambar 6, penulis mengonfigurasi *Network Address Translation* (NAT) pada *Edge_Router* untuk akses internet perangkat internal. Konfigurasi dimulai dengan *access-list 103 permit ip 192.168.10.0 0.0.0.255 any* dan *access-list 103 permit ip 192.168.20.0 0.0.0.255 any* untuk mencocokkan lalu lintas dari VLAN 10 dan VLAN 20. Dua *route-map* terpisah, *ISP_A* dan *ISP_B*, dibuat untuk mengarahkan lalu lintas sesuai dengan antarmuka keluar yang relevan (*FastEthernet0/0* untuk ISP A, *FastEthernet1/0* untuk ISP B). Penerapan NAT *overload* (PAT) dengan *ip nat inside source route-map ISP_A Interface Fa 0/0 Overload* dan *ip nat inside source route-map ISP_B Interface Fa 1/0 Overload* pada masing-masing antarmuka luar memastikan bahwa lalu lintas internal diterjemahkan menggunakan IP publik dari ISP yang aktif.

B. Pengujian Fungsionalitas Dasar Jaringan

Konektivitas dasar diverifikasi dari *Edge_Router*. Perintah *show ip interface brief* mengonfirmasi semua antarmuka *up/up*. Setelah itu penulis melakukan pengujian *ping* ke *DST_Router* (192.168.1.2), *gateway* ISP A (100.100.100.2), *gateway* ISP B (200.200.200.2), dan *Internet_Router* (8.8.8.8) semuanya berhasil, memvalidasi fondasi jaringan.

1. Verifikasi NAT

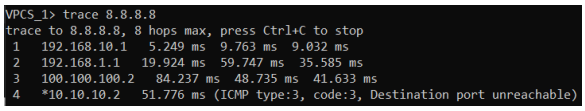
```
EDGE_ROUTER#show ip nat statistics
Total active translations: 23 (0 static, 23 dynamic; 23 extended)
Outside interfaces:
  FastEthernet0/0, FastEthernet1/0
Inside interfaces:
  FastEthernet2/0
Hits: 512  Misses: 0
CEF Translated packets: 444, CEF Punted packets: 30407
Expired translations: 175
Dynamic mappings:
-- Inside Source
[Id: 6] route-map ISP_A interface FastEthernet0/0 refcount 10
[Id: 7] route-map ISP_B interface FastEthernet1/0 refcount 13
nat-limit statistics:
 max entry: max allowed 0, used 0, missed 0
```

Gambar 7. Verifikasi NAT

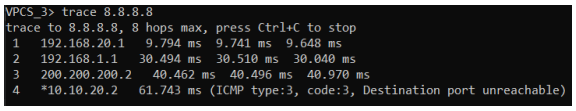
Untuk memverifikasi terjemahan alamat dan statistik NAT, perintah *show ip nat statistics* pada Gambar 7, memberikan *output* yang diperoleh menunjukkan adanya 512 *hits* dan 23 *active translations* (dinamis), serta konfigurasi *dynamic mappings* yang aktif untuk *route-map ISP_A* melalui *interface FastEthernet0/0* dan *ISP_B* melalui *interface FastEthernet1/0*. Hal ini secara jelas memvalidasi bahwa fungsi NAT telah berjalan dengan baik.

2. Verifikasi Arah Lalu Lintas dan Pembahasan Hasil PBR

Penulis memverifikasi implementasi PBR dengan menjalankan perintah *traceroute* dari setiap *Virtual PC* (VPCS) menuju 8.8.8.8.



Gambar 8. (a) *Traceroute* VLAN 10



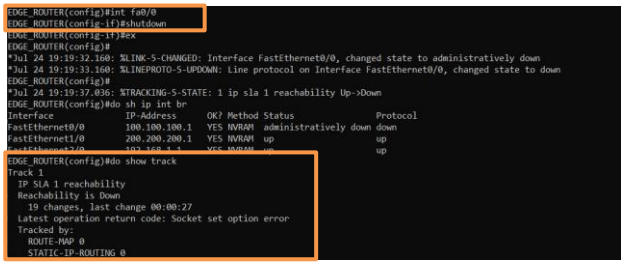
(b) *Traceroute* VLAN 20

Dari VPCS_1 pada Gambar 8 (a), penulis mengetik perintah *trace 8.8.8.8*. *Output* yang diamati menunjukkan *hop* kedua adalah 192.168.1.1 (*Edge_Router*), diikuti oleh *hop* ketiga 100.100.100.2 (ISP A), secara jelas mengonfirmasi lalu lintas diarahkan via ISP A. Dari VPCS_3 pada Gambar 8 (b), secara jelas mengonfirmasi lalu lintas diarahkan *via hop* ketiga 200.200.200.2 (ISP B). Hasil *traceroute* secara definitif mengonfirmasi keberhasilan implementasi PBR.

C. Pengujian dan Analisis Kinerja Failover dan Failback

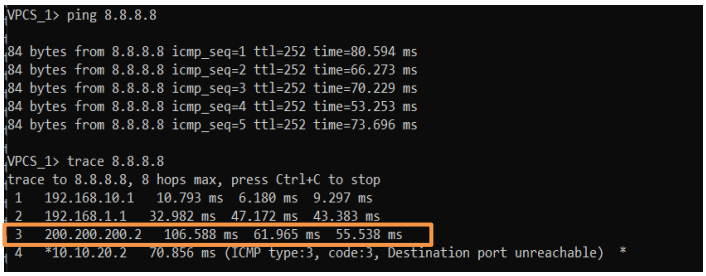
1. Pengujian Failover

1) Mematikan koneksi ISP A



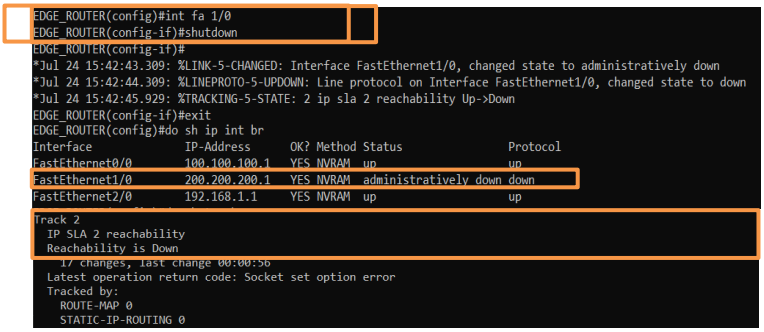
Gambar 9 . Mematikan Koneksi ISP A

Penulis mengetikkan perintah *int fa0/0*, kemudian *shutdown* untuk mematikan interface yang terhubung ke ISP A. Setelah itu melakukan verifikasi *track* dengan mengetikan perintah *do show track* untuk melihat memastikan status *track* 1 sudah *down*, seperti yang dapat dilihat pada Gambar 9.



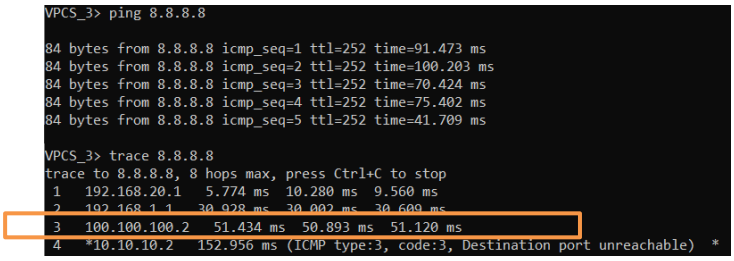
Gambar 10. Uji Coba Konektivitas Internet *Failover* ke ISP B

Pada Gambar 10, Penulis mematikan antarmuka *Fa1/0* yang terhubung ke ISP B. Perintah *show track* mengonfirmasi track 2 berstatus *down*. Pengujian *ping* dan *traceroute* dari VPCS_3 (VLAN 20) menunjukkan pengalihan lalu lintas yang berhasil dari ISP B ke ISP A*Mematikan koneksi ISP B*



Gambar 11 . Mematikan Koneksi ISP B

Penulis mengetikkan perintah *int fa1/0* kemudian *shutdown* untuk mematikan *interface* yang terhubung ke ISP B. Setelah itu melakukan verifikasi *track* dengan mengetikan perintah *do show track* untuk melihat memastikan status *track* 2 sudah *down*, seperti yang dapat dilihat pada Gambar 11.



Gambar 12. Uji Coba Konektivitas Internet *Failover* ke ISP A

Kemudian pada Gambar 12, penulis melakukan uji coba konektivitas internet dengan mengetikkan perintah *ping 8.8.8.8* dari VPCS_3 (yang berada di VLAN 10), dan hasilnya menunjukkan keberhasilan. Hasil *trace* menunjukkan perubahan jalur pada *hop* 3, maka mengindikasikan bahwa manajemen lalu lintas berbasis kebijakan (PBR) telah berhasil mengalihkan lalu lintas VPCS_3 sesuai dengan rute yang diharapkan, memanfaatkan koneksi ISP A.

2. Pemantauan Failover Jaringan

Dalam menganalisis kinerja *failover* secara komprehensif, penulis melakukan pemantauan yang mencakup: waktu deteksi kegagalan, waktu peralihan lalu lintas, dan beban CPU. Pemantauan ini dilakukan dengan mengkonfigurasi frekuensi IP SLA yang berbeda, yaitu 5 detik, 30 detik, 60 detik, serta kualitas layanan *failover*.

TABEL 2
DATA WAKTU DETEKSI DAN PERALIHAN *FAILOVER*

Frekuensi IP SLA (detik)	Waktu Deteksi Kegagalan (detik)	Waktu Peralihan Lalu Lintas (detik)	Beban CPU Saat *Failover* (persen)
5	3.96	1.08	3%
30	21.62	1.07	3%
60	45.63	1.09	3%

1) Pemantauan Waktu Deteksi Kegagalan

Waktu ini diukur dengan membandingkan *timestamp* di *log* sistem *router* saat simulasi kegagalan dimulai (*shutdown* pada *interface* ISP) dengan *timestamp* saat *track object* yang terkait berubah status menjadi *down*. Hasilnya menunjukkan waktu deteksi kegagalan yang bervariasi sesuai dengan frekuensi IP SLA yang diatur: 3.96 detik (frekuensi 5 detik), 21.62 detik (frekuensi 30

detik), dan 45.63 detik (frekuensi 60 detik). Hal ini membuktikan bahwa frekuensi IP SLA secara langsung memengaruhi seberapa cepat sistem mulai bereaksi terhadap kegagalan

2) Pemantauan Waktu Peralihan Lalu Lintas dengan Wireshark

Pengukuran ini dilakukan dengan menganalisis *capture* paket menggunakan *Wireshark*. Penulis menghitung selisih waktu antara *timestamp* paket terakhir yang berhasil dikirim melalui jalur primer dengan *timestamp* paket pertama yang berhasil dikirim melalui jalur cadangan setelah *failover*. Hasilnya menunjukkan bahwa waktu peralihan sangat cepat dan konsisten di setiap frekuensi, yaitu sekitar 1.08 detik (frekuensi 5 detik), 1.07 detik (frekuensi 30 detik), dan 1.09 detik (frekuensi 60 detik). Data ini membuktikan bahwa, terlepas dari waktu deteksi kegagalan, mekanisme PBR mampu mengalihkan lalu lintas secara hampir seketika begitu *failover* dipicu.

3) Pemantauan Beban CPU Router

Pemantauan ini menggunakan perintah *show process cpu* dan *show processes cpu history*. Hasilnya menunjukkan angka yang stabil dan sangat rendah, yaitu konsisten pada 1% untuk kondisi normal dan 3% selama fase transisi *failover*. Meskipun terjadi lonjakan sesaat pada persentase *interrupt* saat *show process* cpu dieksekusi selama transisi, hal ini adalah respons normal *router*. Temuan ini mengonfirmasi bahwa solusi IP SLA dan PBR adalah ringan dan tidak membebani sumber daya *router* secara signifikan, bahkan saat frekuensi pemantauan ditingkatkan.

4) Pemantauan Kualitas Layanan Failover

TABEL 3
DATA KUALITAS LAYANAN FAILOVER

Fase Analisis	Metrik	F= 5	F=30	F=60
Kondisi Normal	*Troughput*	stabil	stabil	stabil
	RTT	54 ms	51 ms	52 ms
	Jitter	15 ms	18 ms	12 ms
	Packet Loss	0 %	0 %	0 %
Selama Transisi (*Failover*)	*RTT*	544 ms	N/A	N/A
	Jitter	N/A	N/A	N/A
	Packet Loss	87.5 %	100%	100 %
Setelah Transisi	*Troughput*	stabil	stabil	stabil
	RTT	54 ms	52 ms	58 ms
	Jitter	16 ms	13 ms	16 ms
	Packet Loss		0 %	0 %

Seperti yang disajikan pada Tabel 3, performa jaringan menunjukkan stabilitas dengan ditandai oleh *throughput* yang konsisten dalam kondisi normal, nilai RTT (*Round-Trip Time*) yang rendah, serta *jitter* yang minimal dan 0% *packet loss*. Namun, saat terjadi kegagalan dan sistem *failover* mulai beraksi, jaringan memasuki fase transisi. Pada momen ini, terjadi lonjakan RTT yang ekstrem dan *packet loss* yang parah, bahkan mencapai 100% pada frekuensi 30 dan 60 detik. Setelah proses transisi selesai, jaringan berhasil memulihkan kinerjanya. *Throughput* kembali pulih, dan *packet loss* kembali menjadi 0%. Meskipun performa jalur cadangan menunjukkan sedikit variasi di mana RTT pada frekuensi 60 detik sedikit lebih tinggi (58ms) dibandingkan frekuensi lainnya secara keseluruhan, performa tersebut masih berada dalam batas yang dapat diterima, membuktikan efektivitas sistem *failover* dalam menjaga ketersediaan layanan.

3. Analisis KinerjaFailover

Analisis kinerja *failover* yang terintegrasi antara IP SLA dan PBR menunjukkan keandalan yang tinggi. Pengujian secara sistematis membuktikan bahwa waktu deteksi kegagalan memiliki ketergantungan langsung pada frekuensi IP SLA. Semakin pendek interval pemantauan yang dikonfigurasi, semakin cepat sistem dapat mengidentifikasi masalah pada jalur utama. Sebagai contoh, waktu deteksi yang tercatat adalah 3.96 detik saat frekuensi IP SLA diatur 5 detik, namun memanjang hingga 45.63 detik pada frekuensi 60 detik. Meskipun waktu deteksi bervariasi, analisis yang lebih mendalam pada waktu peralihan lalu lintas menunjukkan hasil yang sangat konsisten, yaitu hanya dalam kisaran 1.07-1.09 detik. Kecepatan ini membuktikan bahwa begitu IP SLA mendeteksi kegagalan, PBR mampu bereaksi secara hampir seketika. Efisiensi solusi ini juga didukung oleh pemantauan beban CPU *router*, yang tetap stabil pada 1% dalam kondisi normal dan hanya mengalami lonjakan minimal hingga 3% selama fase transisi *failover*, menunjukkan bahwa solusi IP SLA dan PBR adalah konfigurasi yang ringan dan tidak membebani sumber daya.

3. Pengujian dan Analsis Failback

Setelah pengujian *failover* dilakukan, penulis melakukan simulasi failback dengan mengaktifkan kembali antarmuka yang sebelumnya *shutdown*. Perintah *no shutdown* diberikan, yang memicu IP SLA 1 untuk kembali memantau koneksi ke 8.8.8.8. Begitu *track 1* kembali berstatus up, sistem secara otomatis mengembalikan lalu lintas dari VLAN 10 ke jalur utama melalui ISP A, sesuai dengan kebijakan PBR. Hasil pengujian *ping* dan *traceroute* dari VPCS_1 mengonfirmasi keberhasilan ini, menunjukkan bahwa lalu lintas kini kembali melewati next-hop 100.100.100.2. Analisis data menunjukkan bahwa waktu yang dibutuhkan untuk proses *failback* sangat cepat dan konsisten, ditandai hanya dengan lonjakan RTT singkat tanpa adanya RTO. Ini membuktikan bahwa sistem tidak hanya efisien dalam mengalihkan lalu lintas ke jalur cadangan, tetapi juga mampu memulihkannya dengan lancar ke jalur utama begitu koneksi pulih, menjamin ketersediaan jalur secara berkelanjutan.

IV. KESIMPULAN

Sebagai kesimpulan dari pengujian dan pembahasan yang telah dilakukan, implementasi IP SLA yang terintegrasi dengan PBR terbukti menjadi solusi yang andal dan efisien untuk mengoptimalkan *failover* pada jaringan *multi-homed*. Meskipun waktu deteksi kegagalan sangat dipengaruhi oleh frekuensi IP SLA, di mana frekuensi yang lebih rendah menghasilkan waktu deteksi yang lebih cepat. Kinerja PBR dalam mengalihkan lalu lintas terbukti sangat cepat dan konsisten di kisaran 1.07-1.09 detik. Temuan ini menunjukkan bahwa hambatan utama dalam proses *failover* terletak pada kecepatan deteksi, bukan pada mekanisme pengalihan. Solusi ini juga sangat efisien dari segi sumber daya, dengan beban CPU yang hanya mengalami lonjakan minimal hingga 3% selama transisi. Hal ini mengonfirmasi bahwa IP SLA dan PBR dapat diimplementasikan tanpa membebani perangkat keras secara signifikan, bahkan ketika frekuensi pemantauan ditingkatkan. Namun, validitas hasil ini terbatas pada lingkungan lab virtual dan skenario lalu lintas ringan, yang tidak sepenuhnya merepresentasikan kondisi jaringan riil dengan beban trafik tinggi dan variasi jenis lalu lintas. Selain itu, penggunaan ICMP *echo* sebagai satu-satunya metode pemantauan memiliki keterbatasan, karena performanya bisa dipengaruhi oleh kebijakan *rate-limit* di sisi *upstream* atau masalah spesifik pada target. Untuk penelitian lebih lanjut, disarankan untuk menguji sistem ini dengan lalu lintas non-ICMP (seperti HTTP atau VoIP) dan di bawah beban trafik yang lebih berat untuk mengevaluasi dampak RTT dan *packet loss* yang lebih realistis. Menggunakan beberapa target pemantauan atau metode deteksi lain juga dapat meningkatkan keandalan sistem secara keseluruhan, menjadikannya pilihan ideal untuk meningkatkan ketahanan (*resilience*) dan kontinuitas layanan di lingkungan jaringan yang lebih kompleks.

REFERENSI

[1] H. S. Pratama, T. U. Kalsum, dan H. Alamsyah, "The Implementation of Internet-Based Computer Network at SMP Negeri 21 Central Bengkulu," *Jurnal Komputer, Informasi Dan Teknologi*, vol. 1, no. 2, hal. 174–179, 2021.

[2] J. Rahmadani, "Implementasi Sistem Load Balancing dengan Metode Equal Cost Multi Path (ECMP) Interkoneksi Jaringan pada Kantor Dinas Koperasi Palopo," *Peripheral*, vol. 1, no. 2, hal. 81–98, 2025.

[3] K. Finata dan K. Nasution, "Analisis Kinerja Protokol Routing Open Shortest Path First (OSPF) pada Jaringan Universitas Islam Sumatera Utara," *Sudo Jurnal Teknik Informatika*, vol. 3, no. 2, hal. 76–89, 2024. doi: 10.56211/sudo.v3i2.532.

[4] K. Shahid, S. N. Ahmad dan S. T. H. Rizvi, "Optimizing Network Performance: A Comparative Analysis of EIGRP, OSPF, and BGP in IPv6-Based Load-Sharing and Link-Failover Systems," *Future Internet*, vol. 16, no. 9, hal. 339, 2024. doi: 10.3390/fi16090339.

[5] E. K. Silalahi, Y.C. Sitanggang, E. Suryaningsih, dan D. Kiswanto "Implementasi dan Analisis Protokol HSRP dan VRRP dalam Meningkatkan Redudansi Gateway pada Jaringan Virtual," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 2, hal. 1431–1444, 2025. doi: 10.23960/jitet.v13i2.6474

[6] NetworkLessons, "Deteksi Penerusan Dua Arah (BFD)," Network Lessons. Diakses: 19 Juli 2025. [Daring]. Tersedia pada: https://redeyenetworks.com/the-impact-of-network-Downtime-on-business-productivity-and-how-toprevent-it/

[7] Cisco Systems, "IP Addressing Configuration Guide, Cisco IOS XE 17.x," Cisco Systems. Diakses: 19 Juli 2025. [Daring]. Tersedia pada: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/ip-addressing/b-ip-addressing

[8] D. E. Cahyono, "Implementasi Load balance Menggunakan Metode Policy Based Route (PBR) pada Politeknik Sawunggalih aji Kutoarjo," *Jurnal Ekonomi dan Teknik Informatika*, vol. 10, no. 2, hal. 65-70, 2022. doi: 10.37601/jneti.v10i2.217

[9] F. A. S. A. Putra, P.H. Trisnawan dan A. Basuki, "Analisis Perbandingan Kinerja Metode Single Homing dan Multihoming dengan Protokol (BGP)". *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no.3, hal. 1086-1092, 2021.

[10] R. Almakhi, Anton dan F. S. Nugraha, "Implementasi Load Balancing dan Menggunakan IP SLA pada PT Pan Pacific Insurance". *Jurnal Infortech*, vol. 4, no. 2, hal. 98-104, 2022.

[11] L. N. Sahenda, P. a. Wibowo, P. S. D. Puspitasari, "Peningkatan Kualitas Layanan Internet sekolah dengan Metode Policy Based Route (PBR) di SMK PP Negeri 1 Tegalampel Bondowoso". *Indonesian Journal of Community Services*, vol. 4, no. 1, hal. 9-15. 2024. doi: 10.53363/bw.v4i1.242

[12] R. G. Ediyasanto dan T, Hardiani, "Implementasi Policy based routing untuk Load Balancing dan Failover Koneksi Internet di RSA UGM," *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 2, hal. 2930-2936, 2025.

[13] A. I. Cahya, I. R. Widiasari, "Analisis Perbandingan Performansi Metode HSRP Dan VRRP Sebagai Backup Link Koneksi Jaringan". *Jurnal Ilmiah Komputer*, vol. 19, no. 1, hal. 381-390. 2023.

[14] A. B. Hassan dan M. Mansour, "Performance Evaluation of Bidirectional Forwarding Detection (BFD) over the Virtual Router Redundancy Protocol (VRRP)," *in Procedia Computer Science*, vol. 251, no. 1, hal. 1-8. doi: 10.1016/j.procs.2024.11.108

[15] A. Hafid, H. Mukhtar, dan D. Harlian, "Penerapan Failover Network Menggunakan Jaringan VPN dan Jaringan Wireless Point to Point pada Distance Building di PT.Titipan Kilat Riau," *Jurnal Teknologi Terapan*, vol. 9, no. 1, hal. 35-43, 2023.