

Implementasi Jaringan *Failover* Menggunakan OSPF Dengan Algoritme *Dijkstra*

Candra Milad Ridha Eislam^{1*}, Dadang Iskandar Mulyana², Yuma Akbar³

Program Studi Teknik Informatika, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika,
 Jakarta, Indonesia

*Email Corresponding Author: candramilad@gmail.com

Abstract

Companies usually implement OSPF (Open Shortest Path First) routing on a single link as a communication path to the Internet. This is very vulnerable when there is a problem with the link. The problem that is often encountered is when the link is down. To keep the internet connection running, it is necessary to further develop the network communication process by implementing failover. Failover utilizes two links, the first as the main link and the second link as a backup. This paper will simulate the implementation of a failover network using OSPF in the EVE-NG simulation. The results when testing OSPF use the cost value to determine the link selection. The link with the lowest cost value will be used as the main line, while the link with the higher cost value is used as the backup line. When the main connection is interrupted, the connection will switch to the backup. The ping results obtained in the test when the line changed were 21ms; and it only happened two times unreachable, while for browsing traffic there were no problems on the network.

Keywords: Internet; failover network; Open Shortest Path First; Dijkstra's Algorithm

Abstrak

Perusahaan biasanya mengimplementasikan routing OSPF (Open Shortest Path First) pada satu *single link* sebagai jalur komunikasi ke Internet. Hal ini sangat rentan ketika terjadi permasalahan pada *link*. Permasalahan yang sering ditemui adalah ketika *link down*. Untuk menjaga koneksi internet agar tetap berjalan perlu adanya pengembangan lebih lanjut pada proses komunikasi jaringan dengan menerapkan *failover*. *Failover* memanfaatkan dua buah *link*, pertama sebagai *main link* dan *link* kedua sebagai *backup*. Paper ini akan mensimulasikan implementasi jaringan *failover* menggunakan OSPF pada simulasi EVE-NG. Hasilnya pada saat pengujian OSPF menggunakan nilai *cost* untuk menentukan pemilihan linknya. *Link* dengan nilai *cost* terkecil akan digunakan sebagai jalur utama, sedangkan *link* dengan nilai *cost* yang lebih tinggi digunakan sebagai jalur *backup*. Ketika koneksi utama terganggu maka koneksi akan berpindah ke *backup*. Hasil *ping* yang didapat dalam uji coba ketika jalur berpindah adalah 21ms; dan hanya terjadi dua kali *unreachable*, sedangkan untuk trafik *browsing* tidak mengalami kendala pada jaringan.

Keywords: Internet; Jaringan failover; Open Shortest Path First; Algoritme Dijkstra

1. Pendahuluan

Internet merupakan hal yang paling dibutuhkan. Ketersediaan terhadap informasi yang cepat, mudah dan tepat akan mengakibatkan kinerja pada sebuah jaringan komputer menjadi meningkat dan akhirnya menyebabkan jaringan komputer terjadi terganggu jika tidak bisa mengendalikan peningkatan jaringan komputer tersebut [1]. Untuk dapat berkomunikasi dengan baik antara jaringan lokal perlu adanya mekanisme *routing* yang handal agar komunikasi dapat tetap berjalan dengan baik.

Salah satu *routing* yang digunakan adalah OSPF, yang menggunakan algoritme *Dijkstra*. Dengan algoritme ini OSPF dapat memilih jalur utama, sehingga jika terjadi gangguan pada jalur utama, maka jalur cadangan akan mengambil alih fungsi jalur utama [2]. Algoritme OSPF telah meluas digunakan dalam merancang routing jaringan internet. Musril [3] telah menguji *Open Shortest Path First* (OSPF) Untuk Menentukan Jalur Terbaik Dalam Jaringan

Internet. Hasil uji yang telah dilakukan menunjukkan protokol routing OSPF dapat memberikan pilihan jalur terbaik (*best path*) dalam mengantarkan paket data. Uji coba protokol routing OSPF dalam mengoptimalkan jalur pengantaran paket data juga telah dilakukan oleh Polii [4]. Penelitian tersebut bertujuan untuk mempercepat waktu konvergensi pada protokol OSPF setelah inialisasi *router* berdasarkan studi topologi jaringan. Hasil pengujian menunjukkan terjadi percepatan konvergensi selama 2,4detik, Sehingga dapat disimpulkan terjadi peningkatan rerata kecepatan waktu konvergensi sebesar 26,95%

Paper kami menyajikan model implementasi jaringan *failover* pada OSPF sehingga, yang diharapkan dapat memberikan solusi permasalahan pada koneksi Internet, dengan menyediakan link tambahan sebagai jalur *backup*, sehingga ketika terjadi permasalahan pada *link*, koneksi akan berpindah dan menggunakan koneksi *backup*.

2. Tinjauan Pustaka

Terdapat banyak *routing protocol* yang digunakan dalam jaringan saat ini, tetapi semuanya mempunyai tujuan yang sama yaitu secara umum untuk pemilihan jalur di jaringan sesuai penelitian dari [5] yang membahas mengenai analisis cara operasi dari berbagai *routing protocol* pada jaringan local yang disimulasikan menggunakan aplikasi simulasi *Cisco Packet Tracer*.

Routing merupakan salah satu komponen penting dari infrastruktur jaringan di internet saat ini [6] namun kinerja *routing* pada *router* semakin membebani kinerja *router*, oleh karena itu peneliti melakukan perancangan *routing* OSPF untuk IP jaringan internet dan melakukan evaluasi kinerjanya dengan melakukan perbandingan antara kinerja VLAN pada OSPF dengan pengelolaan VPLS melalui MPLS dengan hasil pengelolaan VLAN pada OSPF memiliki kinerja terbaik yaitu mendapatkan nilai throughput hingga 59,6%, *packet loss* hingga 0,32%, *jitter* dan delay sebesar 4,44 ms.

Kemudian pada penelitian yang dilakukan oleh [7] membahas mengenai evaluasi performa *routing protocol* antara RIP, OSPF dan IGRP dengan hasil bahwa OSPF memberikan kinerja terbaik pada pengujian beberapa *traffic* seperti VoIP, Video, FTP dan HTTP dalam hal rata-rata delay pada *interface router* dengan nilai 0.285 msec sedangkan urutan kedua adalah IGRP dengan nilai rata-rata delay 0.375 msec dan RIP dengan nilai delay adalah 0.5 dari penelitian ini disimpulkan bahwa OSPF merupakan salah satu *routing protocol* dengan performa dan hasil yang terbaik.

OSPF merupakan salah satu *routing protocol* yang mempunyai nilai *latency* terendah hal ini dibuktikan dengan penelitian yang dilakukan oleh [8] membahas mengenai performa dari RIP, EIGRP dan OSPF yang diuji pada komunikasi nirkabel untuk target yang diukur dalam penelitian tersebut adalah *voice*, HTTP dan *traffic video* baik yang dikirim maupun diterima oleh *router*, serta rata-rata *latency* dan nilai throughput dengan hasil OSPF menunjukan nilai *latency* terendah untuk pengujian *voice*, HTTP dan *traffic video* dengan rata-rata nilai *latency* 5 ms. Kemudian dari segi *throughput* yang dihasilkan OSPF lebih baik dibandingkan dengan *protocol routing* lain, dengan hasil ini OSPF lebih cocok diterapkan untuk jaringan nirkabel.

Dynamic Routing OSPF merupakan salah satu jenis *routing protocol* yang dapat membangun topologi jaringan secara mandiri dengan memilih jalur terbaik hal ini yang dibahas oleh [9] dengan mengimplementasikan *dynamic routing* OSPF di SMKN2 Teluk Kuantan untuk menggantikan *routing static* pada jaringan di sekolah tersebut. Hal ini dikarenakan *routing static* memungkinkan terjadi kerusakan atau trouble secara terpisah pada jaringan salah satu permasalahan pada *routing static* adalah perbaikan yang memakan waktu lama karena administrator harus melakukan update entri *routing* sehingga menghambat kegiatan sekolah ketika akan membutuhkan akses internet. Dengan diimplementasikannya *dynamic routing* OSPF, entri *routing* dapat diupdate secara dinamis sehingga administrator jaringan tidak perlu lagi menambahkan entri *routing* secara manual.

OSPF merupakan *routing protocol* yang menggunakan *routing link-state* yang memiliki kelebihan dapat menghasilkan banyak jalur (*best path*) ke sebuah tujuan network. Pada penelitian [10] OSPF diimplementasikan sebagai pengganti dari *static routing* pada jaringan SMK Budi Mulia Tangerang hal ini dikarenakan OSPF dapat mempermudah perubahan *network* dan proses *maintenance* karena informasi *routing* tetap dijaga sehingga dapat *up to date*.

OSPF juga menjadi solusi untuk meminimalkan *downtime* pada jaringan seperti yang telah dilakukan oleh peneliti [11] dengan mengimplementasikan OSPF sebagai pengganti *static routing* pada Univeritas Merdeka Malang. Hasil penelitian menunjukan bahwa OSPF dapat

melakukan pemilihan jalur secara otomatis oleh *router*, sehingga dapat memudahkan administrator dalam mengelola jaringannya. Ketika terjadi perubahan pada jaringan administrator hanya perlu melakukan penambahan pada *router* yang berada pada satu segmen jaringan tersebut kemudian informasi akan dikirimkan secara otomatis oleh *router* ke *router* lain dalam waktu 8,23 detik. Apabila terjadi kegagalan *link*, maka *router* secara otomatis dapat mengalihkan jalur routing dalam waktu 3 detik sesuai dengan simulasi yang telah dilakukan.

Menerapkan routing OSPF untuk keamanan jaringan terutama untuk mencegah terjadinya *hacking* dan membangun jaringan terpusat di Dinas Komunikasi dan Informatika Depok yang telah diteliti oleh peneliti pada [12] dengan menerapkan OSPF routing, *Demilitarized Zone* dan *Firewall* menggunakan perangkat jaringan Mikrotik *RouterBoard*. OSPF menjadi salah satu solusi pengganti *static routing* dikarenakan OSPF dapat membuat jalannya jaringan data menjadi lebih baik dan lebih cepat, serta administrator dapat mengelola jaringan lebih baik dan lebih mudah.

OSPF merupakan *protocol routing* yang menggunakan Algoritme link-state untuk membangun dan menghitung jalur terbaik ke semua tujuan yang diketahui [13] pada penelitian ini peneliti melakukan Analisa performa OSPF pada komunikasi jaringan antara kantor pusat dan cabang dengan hasil yang menunjukan bahwa OSPF mengirimkan setiap database ke masing-masing *router* dengan nilai transmisi data maksimal 1 ms dan pengiriman data berhasil mencapai 100%.

Peneliti pada [14] melakukan penelitian pada pembuatan jaringan baru dimana routing OSPF diterapkan pada jalur utama dan menerapkan *redistribution* pada setiap *router* yang terhubung ke masing-masing client. *Redistribution* ini merupakan metode routing untuk meneruskan informasi routing ke routing lain agar dapat saling menukarkan informasi routing table masing-masing. Hasil penelitian ini, berdasarkan pada penjumlahan nilai metric dijkstra dengan *metric redistribute*, maka jalur terbaik (*best path*) ditentukan berdasarkan nilai metric yang lebih kecil dan *redistribution* dapat menghubungkan *routing dynamic* OSPF dengan *routing protocol* lain yang terhubung ke masing-masing client.

Penggunaan *dynamic routing* OSPF dan BGP dengan metode pembebanan kinerja VRRP, HSRP dan GLBP sebagai jalur cadangan yang dilakukan oleh peneliti [15] dengan hasil merekomendasikan *dynamic routing* OSPF dengan metode pembebanan VRRP memiliki perbandingan waktu 1 detik ketika terjadi kegagalan pada jalur internet. Hal ini dibuktikan dengan nilai QoS sebesar 3,96 dengan kategori sangat memuaskan, lebih baik dibandingkan lainnya.

OSPF dapat membuat dan menemukan topologi jaringan secara lengkap seperti yang telah diteliti [16] dengan hasil OSPF dapat dengan cepat melakukan konvergensi dengan mengirimkan atau menerima *Link State Update* (LSU) pada *router* dan melakukan Analisa LSA pada *router* sehingga didapatkan topologi jaringan yang lengkap.

Pada penelitian [17] salah satu metode yang digunakan untuk meningkatkan kecepatan data dengan menentukan cara pengiriman data yang lebih cepat dan terbaik di jaringan adalah dengan menggunakan *routing protocol* OSPF dan EIGRP. Hasil dari penelitian ini adalah *throughput* yang didapat sebesar 45,75 bps, delay sebesar 5,91 ms, jitter sebesar 1,477 ms, dan loss paket sebesar 0%. Sedangkan nilai *cost* pengiriman paket data pada OSPF adalah 781,25 kbps dan nilai metric pada EIGRP adalah 20 kbps. Dengan nilai ini kedua *routing protocol* dapat diimplemetasikan untuk meningkatkan kinerja jaringan menjadi semakin tinggi.

OSPF dapat mengirimkan informasi perutean didalam *router* yang terhubung kedalam satu autonomous system (AS), permasalahan yang sering timbul dari semakin besar dan luasnya jaringan computer adalah semakin lama waktu untuk memperbarui table routing pada *protocol* OSPF. Pada penelitian ini [18] peneliti mencoba melakukan pengoptimalan kinerja OSPF dengan melakukan perubahan nilai hello interval dengan hasil menunjukan bahwa perubahan hello interval dapat mempengaruhi waktu konvergensi dan pemilihan DR dan BDR dengan hasil hello interval yang digunakan adalah 1 detik sedangkan dengan waktu konvergensi lebih kecil dari 11,7 detik.

Metode yang dapat dikembangkan untuk membagi beban kerja (*load balance*) pada 2 ISP (*Internet Service Provider*) adalah dengan menggunakan ECMP (*Equal Cost Multi Path*) dan mengabungkan dengan *protocol routing* OSPF maupun BGP, pada penelitian ini [19] peneliti melakukan perbandingan performa kinerja antara ECMP yang menggunakan OSPF dan ECMP yang menggunakan BGP dengan hasil *throughput* pada ECMP yang menggunakan

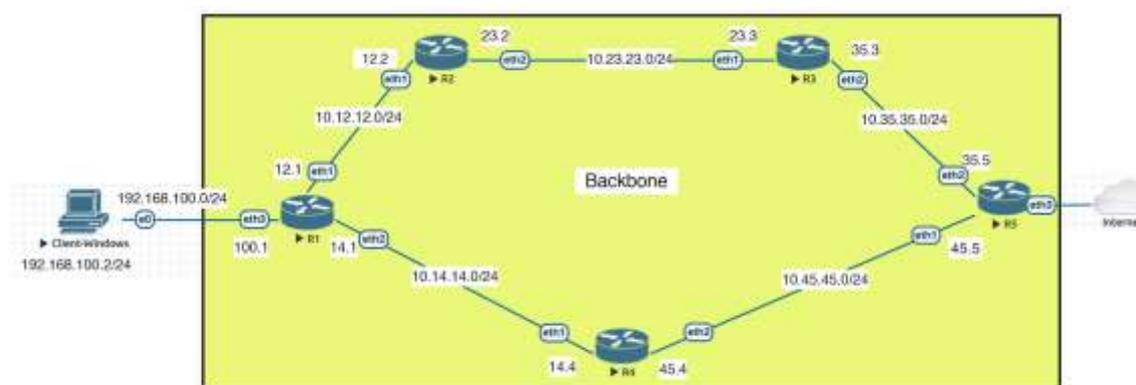
OSPF adalah 764.13 bps hal ini memiliki performansi yang lebih rendah dibandingkan dengan yang menggunakan BGP 818.1 bps) sedangkan untuk nilai delay saat pengiriman data TCP dan UDP OSPF mengalami peningkatan dengan nilai 85.61 ms dibandingkan dengan BGP yang mempunyai nilai 89.23 ms.

State of the art penelitian ini adalah penggunaan *dynamic routing* pada jaringan failover untuk memastikan ketersediaan link, ketika link utama mati maka *traffic* data dari jaringan akan diarahkan ke *link* cadangan.

3. Metodologi

3.1 Model/Topologi Jaringan Yang Dikembangkan

Penelitian ini menggunakan sistem yang dibangun dengan simulasi jaringan virtual EVE-NG dimana terdapat 5 router dan 1 client. Setiap *router* saling terhubung melalui *routing* OSPF dan menjalankan *failover*, ada salah satu router (R1) yang terhubung ke *client* dan 1 *router* (R5) yang terhubung ke Internet. *Router* yang digunakan untuk simulasi adalah Mikrotik CHR. Sedangkan untuk clientnya menggunakan Windows 7. PC *client* akan digunakan untuk melakukan pengujian pada penelitian ini. Berikut merupakan rancangan topologi jaringan yang akan digunakan dalam penelitian ini:



Gambar 1. Topologi Jaringan OSPF *Failover*

Pada Gambar 1, peneliti membuat design topologi menggunakan 5 router yang saling terhubung, kemudian salah satu router terhubung ke *client* dan satu router lain terhubung ke Internet. Semua router dikonfigurasi *dynamic routing* OSPF dengan menggunakan konfigurasi area *backbone* supaya dapat saling berkomunikasi satu sama lain. Sehingga ketika *client* mengirimkan request ke arah internet, permintaan dari client akan dikirimkan melalui salah satu jalur. Pemilihan jalur dilakukan oleh OSPF dengan memperhatikan parameter nilai *cost*. Kemudian setelah ditentukan jalur utama, peneliti akan mencoba mematikan link atau jalur utama tersebut sehingga permintaan dari client ketika menuju ke Internet akan dialihkan menggunakan jalur *backup*.

3.2 Alur Implementasi dari Rancangan Topologi Jaringan

Adapun pembahasan umum dari masing-masing tahapan diatas adalah sebagai berikut:

1. Tahapan dimulai dengan instalasi EVE-NG pada sebuah Server.
2. Selanjutnya peneliti membuat topologi dengan memasukan image MikroTik CHR dan Windows 7 sebagai PC client.
3. Melakukan konfigurasi *Addressing* di setiap router.
4. Setelah melakukan konfigurasi IP address dilakukan konfigurasi OSPF menggunakan area *backbone* pada semua router.
5. Melakukan konfigurasi OSPF redistribute default *route* pada R5, untuk mendistribusikan konfigurasi default *route* ke semua router di OSPF.
6. Untuk *failover* nilai *cost* nya berbeda.
7. Setelah semua router saling terhubung, maka dilakukan pengujian dan analisis.
8. Hasil dari pengujian dan analisis akan ditarik kesimpulan dan telah selesai dilaksanakan.

Untuk menjalankan pengujian, peneliti mencoba mendesain topologi jaringan *redundancy* menggunakan aplikasi simulasi jaringan EVE-NG ada beberapa hal yang dibutuhkan dalam proses pengujian adalah sebagai berikut:

1. Aplikasi Simulasi Jaringan EVE-NG digunakan untuk mendesain topologi jaringan dan memetakan router dan PC pada simulator secara visual.
2. Aplikasi *Wireshark* digunakan untuk menganalisa traffic paket OSPF.
3. Command Prompt digunakan untuk pengetesan traffic ICMP berupa ping dan *traceroute* untuk mengetahui jalur mana yang dipilih.
4. OSPF metric yaitu menggunakan nilai *cost* pada *routing table*.
5. Tabel *addressing* digunakan untuk memetakan alokasi IP *address* pada setiap *route* dan PC pada topologi EVE-NG, sebagai berikut

Tabel 1. *Addressing*

No	Nama Perangkat	Port	IP Address	Gateway	Keterangan
1	R1	Ether1	10.12.12.1/24	10.12.12.2	Link ke R2
		Ether2	10.14.14.1/24	10.14.14.4	Link ke R4
		Ether3	192.168.100.1/24	N/A	Link to client
2	R2	Ether1	10.12.12.2/24	N/A	Link to R1
		Ether2	10.23.23.2/24	10.23.23.3	Link to R3
3	R3	Ether1	10.23.23.3/24	N/A	Link to R2
		Ether2	10.35.35.3/24	10.35.35.5	Link to R5
4	R4	Ether1	10.14.14.4/24	N/A	Link to R1
		Ether2	10.45.45.4/24	10.45.45.5	Link to R5
5	R5	Ether1	10.45.45.5/24	N/A	Link to R4
		Ether2	10.35.35.5/24	N/A	Link to R3
		Ether3	DHCP	DHCP	Link to Internet
6	Client-Windows	Ethernet0	192.168.100.2/24	192.168.100.1	Link to R1

4. Hasil dan Pembahasan

4.1 Mengaktifkan Routing OSPF dengan area *backbone* pada router

```
[admin@R1] > routing ospf network print
Flags: X - disabled, I - invalid
# NETWORK AREA
0 10.12.12.0/24 backbone
1 10.14.14.0/24 backbone
2 192.168.100.0/24 backbone
[admin@R1] >
```

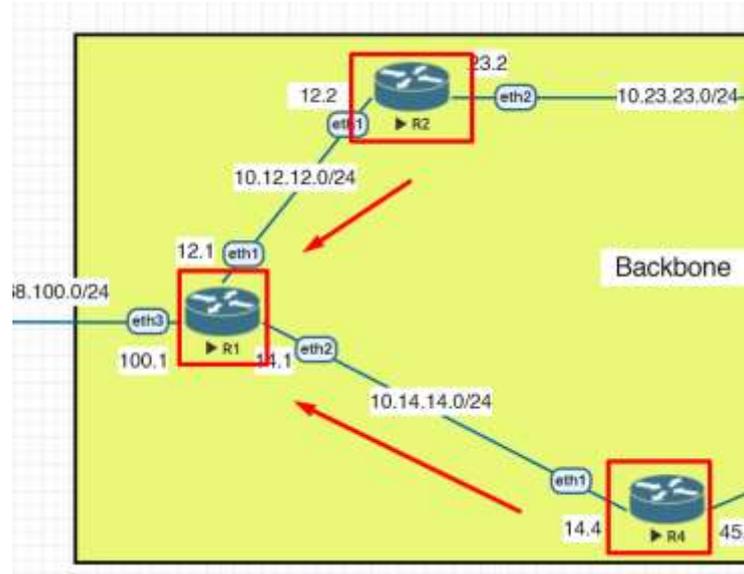
Gambar 2. Mengadvertise *Connected Network* kedalam OSPF pada Mikrotik

Untuk menghubungkan semua router didalam jaringan, perlu menambahkan atau melakukan advertise semua network yang terkoneksi secara langsung (*connected network*) kedalam OSPF *Network* dengan menggunakan area *backbone*. Kemudian untuk memastikan apakah router OSPF sudah terkoneksi dengan lawan, dapat dilihat informasinya melalui OSPF *Neighbor*.

```
[admin@R1] > routing ospf neighbor print
0 Instance=default router-id=10.14.14.4 address=10.14.14.4 interface=ether2 priority=1 dr-address=10.14.14.4
backup-dr-address=10.14.14.1 state="Full" state-changes=5 ls-retransmits=0 ls-requests=0 db-summaries=0
adjacency=2h19m50s
1 Instance=default router-id=10.12.12.2 address=10.12.12.2 interface=ether1 priority=1 dr-address=10.12.12.1
backup-dr-address=10.12.12.2 state="Full" state-changes=5 ls-retransmits=0 ls-requests=0 db-summaries=0
adjacency=3h14m18s
[admin@R1] >
```

Gambar 3. OSPF *Neighbor*

Secara visual informasi yang terlihat pada OSPF *Neighbor* pada R1 diatas akan sesuai dengan gambaran topologi fisik, dimana R1 terkoneksi secara langsung ke R2 (10.12.12.2) dan ke R4 (10.14.14.4).



Gambar 4. Topologi OSPF *Neighbor*

4.2 Mengaktifkan *Redistribute Default Route* pada OSPF

```
[admin@R5] > routing ospf instance print
Flags: X - disabled, * - default
0 * name="default" router-id=0.0.0.0 distribute-default=always-as-type-1
  redistribute-connected=no redistribute-static=no redistribute-rip=no
  redistribute-bgp=no redistribute-other-ospf=no metric-default=1
  metric-connected=20 metric-static=20 metric-rip=20 metric-bgp=auto
  metric-other-ospf=auto in-filter=ospf-in out-filter=ospf-out
[admin@R5] >
```

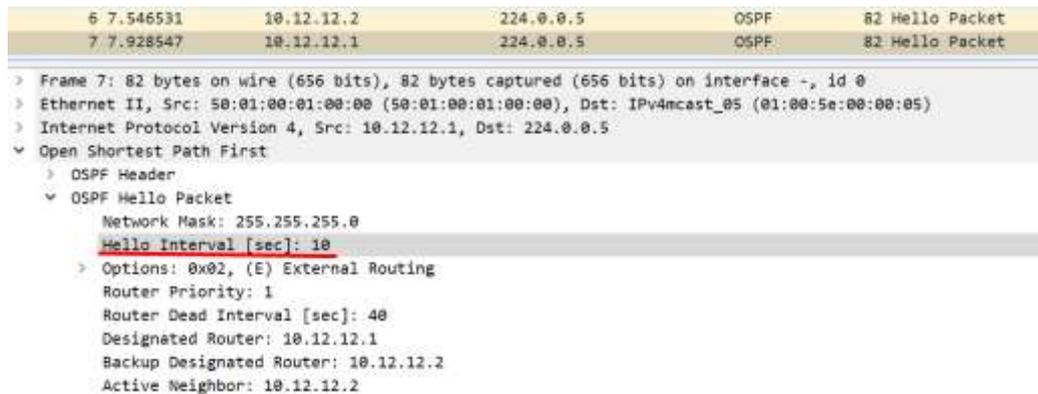
Gambar 5. *Redistribute default route* kedalam OSPF

Pada R5 dilakukan konfigurasi *redistribute default route* kedalam OSPF melalui OSPF instance, dengan menggunakan parameter “always-as-type-1” yang artinya R5 selain akan melakukan distribusi informasi route juga akan memberikan parameter nilai *metric cost*nya kesemua router dengan kombinasi nilai *metric internal* (OSPF) dijumlah dengan nilai *metric external (default route)* sehingga semua router akan mendapat informasi default route didalam routing tablenya dan secara otomatis juga akan mendapat informasi nilai *metric cost* yang digunakan ketika menuju ke *default route*. Informasi *default route* ini akan dianggap berasal dari OSPF ketika dikirim ke semua router di jaringan.

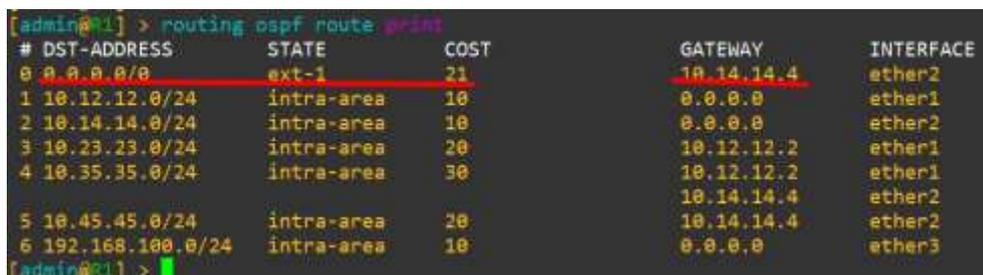
4.3 Hasil Akhir Pengujian

OSPF merupakan salah satu dynamic routing yang masuk ke *category interior gateway protocol*, secara otomatis dapat mendistribusikan informasi *network* yang telah diadvertise ke semua router didalam jaringan. Setelah semua router saling terhubung, Langkah pertama yang peneliti lakukan adalah mengecek bagaimana komunikasi OSPF menggunakan aplikasi *wireshark* untuk melihat lalu lintas data di jaringan.

Pada gambar 6, dapat peneliti simpulkan bahwa OSPF akan mengirimkan hello packet sebagai paket pembuka dalam interval 10 detik melalui IP multicast 224.0.0.5



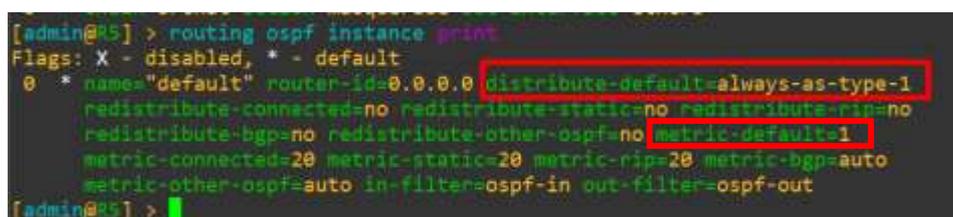
Gambar 6. Tampilan Wireshark untuk melihat traffic OSPF



Gambar 7. OSPF Routing Tabel pada R1

Terlihat pada Gambar 7 bahwa informasi *default route* sudah diperoleh R1 beserta informasi state “ext-1”, informasi ini didapat dari R5 yang telah melakukan *redistribute default route* sebelumnya. Informasi default route akan dilewatkan melalui *gateway* 10.14.14.4 *gateway* ini merupakan alamat IP dari R4, artinya lalu lintas data dari client ketika akan menuju ke internet akan dilewatkan melalui R4 karena nilai *cost*-nya lebih kecil dibandingkan yang lainnya.

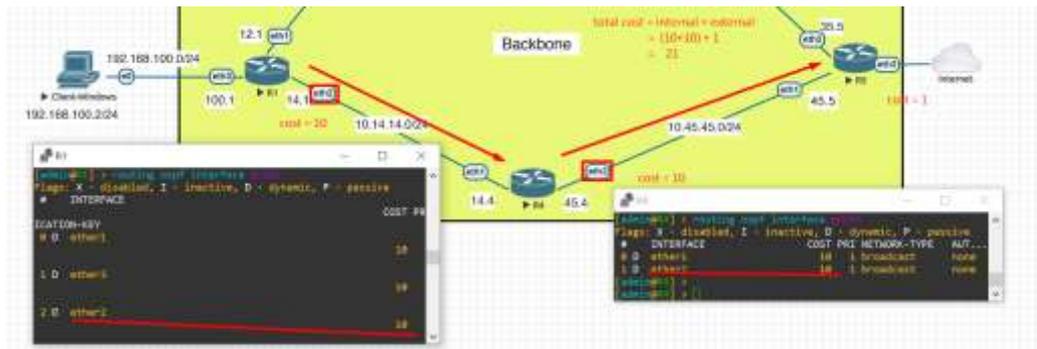
Nilai *cost* yang diperoleh oleh default route pada R1 sebesar 21, dikarenakan pada R5 melakukan *redistribute default route* menggunakan opsi “always-as-type-1” yang artinya OSPF akan melakukan penjumlahan nilai *cost* di setiap *out-interface router* dengan nilai *metric* dari *default route* itu sendiri. Untuk nilai *metric* dari *default route* adalah 1, hal ini dapat terlihat dari informasi nilai *metric* pada OSPF Instance pada Gambar 8.



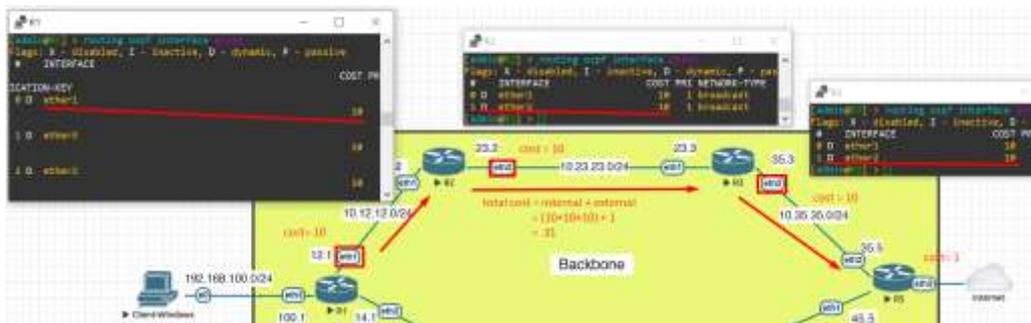
Gambar 8. Nilai metric default route

Sedangkan untuk melihat nilai *metric internal* atau nilai *cost* disetiap *router* dapat dilihat pada OSPF Interface. Secara umum nilai *cost* untuk semua interface di router mikrotik adalah 10. Nilai *cost* akan dijumlahkan oleh *router* di setiap *out-interfacenya*.

Pada gambar 9, dapat peneliti simpulkan bahwa nilai *cost* disetiap *out-interface router* akan dijumlahkan dengan nilai *metric* pada *default route*. Nilai *cost* pada *out-interface* dari R1 ke R4 adalah 10 + nilai *cost* pada *out-interface* dari R4 ke R5 adalah 10 + nilai *metric default route* adalah 1, sehingga nilai total *cost*-nya adalah 10+10+1 = 21.



Gambar 9. Perhitungan nilai cost



Gambar 10. Perhitungan nilai cost

Gambar 10 menjelaskan perhitungan cost pada link lain yaitu ketika permintaan traffic data dikirimkan melalui jalur atas R1 → R2 → R3 → R5, maka jumlah nilai costnya ditiap out-interfacenya adalah nilai cost dari R1 ke R2 adalah 10 + nilai cost dari R2 ke R3 adalah 10 + nilai cost dari R3 ke R5 adalah 10 + nilai metric default route adalah 1, sehingga nilai total costnya adalah 10+10+10+1 = 31. Nilai cost ini lebih besar dibandingkan dengan nilai cost yang melewati jalur bawah R1 → R4 → R5 sehingga permintaan traffic data ke internet akan dikirimkan lewat jalur bawah R1 → R4 → R5 karena sebagai jalur utama sedangkan untuk jalur atas R1 → R2 → R3 → R5 akan digunakan sebagai jalur cadangan. Jalur cadangan ini baru akan digunakan ketika terjadi permasalahan pada jalur utama seperti *link down*. Hal inilah yang disebut sebagai *Failover*.

```
[admin@R1] > tool traceroute google.com src-address=192.168.100.1
# ADDRESS          LOSS SENT  LAST      AVG      BEST     WORST
1 10.14.14.4         0% 2 1.7ms    2.3     1.7     2.8
2 10.45.45.5         0% 2 3.4ms    3.2     3       3.4
3 172.23.0.1         0% 2 3.8ms    3.8     3.7     3.8
4 192.168.100.1      0% 2 4.5ms    4.3     4       4.5
5 10.91.32.1         0% 2 7.6ms    8       7.6     8.4
6 103.75.97.150     0% 2 6.1ms    6.2     6.1     6.2
7 103.83.6.5         0% 2 6.7ms    6.9     6.7     7
8 103.83.6.18       0% 2 18ms     18.2    18      18.4
9 72.14.194.82      0% 2 19.5ms   19.9    19.5    20.2
10 209.85.255.97     0% 2 19ms     19      19      19
11 74.125.242.35     0% 2 20.9ms   24.7    20.9    28.5
12 72.14.232.222    0% 2 22.1ms   22.6    22.1    23
13 209.85.250.141   0% 2 19.7ms   26.7    19.7    33.6
14 142.251.52.25     0% 2 20ms     19.8    19.5    20
15                  100% 2 timeout
16                  100% 2 timeout
17                  100% 2 timeout
18                  100% 2 timeout
19                  100% 1 timeout
- [Q quit|D dump|C-z pause]
```

Gambar 11. Pengujian traceroute dari R1 ke Internet

Hasil *traceroute* pada R1 ketika menuju ke Internet akan dilewatkan melalui *gateway* 10.14.14.4 (R4) kemudian berikutnya akan dikirimkan melalui *gateway* 10.45.45.5 (R5) sampai ke Internet seperti pada gambar 11.

```
C:\Users\ovi>tracert -d google.com

Tracing route to google.com [142.251.12.139]
over a maximum of 30 hops:

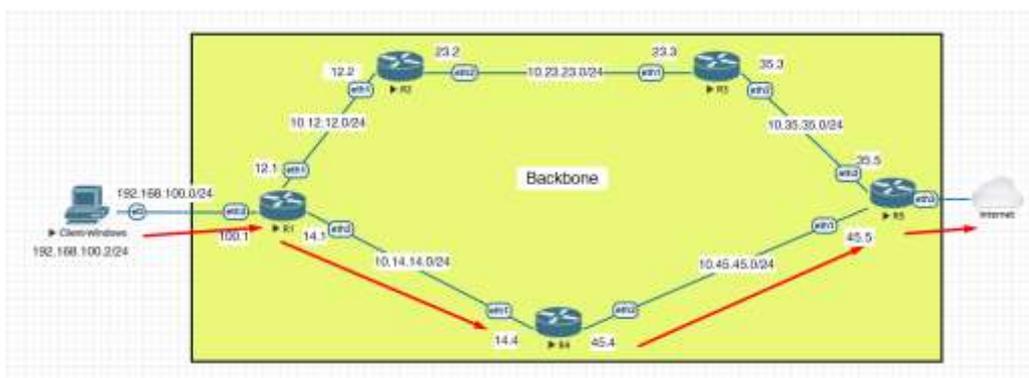
  0  0 ms  0 ms  0 ms  192.168.100.1 ✓
  1  2 ms  1 ms  1 ms  10.14.14.4 ✓
  2  4 ms  2 ms  2 ms  10.14.14.4 ✓
  3  5 ms  3 ms  4 ms  10.45.45.5 ✓
  4  6 ms  4 ms  4 ms  172.23.0.1
  5  8 ms  6 ms  5 ms  192.168.100.1
  6  13 ms  9 ms  10 ms  10.91.32.1
  7  11 ms  7 ms  7 ms  103.75.97.146
  8  10 ms  8 ms  7 ms  103.83.6.5
  9  23 ms  22 ms  22 ms  103.83.6.2
 10  23 ms  21 ms  21 ms  72.14.194.82
 11  24 ms  22 ms  22 ms  108.170.254.225
 12  27 ms  22 ms  24 ms  108.170.254.226
 13  23 ms  23 ms  53 ms  142.251.229.217
 14  23 ms  21 ms  21 ms  142.251.231.202
 15  22 ms  21 ms  21 ms  142.251.52.193
 16  *      *      *      Request timed out.
 17  *      *      *      Request timed out.
 18  *      *      *      Request timed out.
 19  *      *      *      Request timed out.
 20  *      *      *      Request timed out.
 21  *      *      *      Request timed out.
 22  *      *      *      Request timed out.
 23  *      *      *      Request timed out.
 24  *      *      *      Request timed out.
 25  25 ms  21 ms  21 ms  142.251.12.139

Trace complete.

C:\Users\ovi>
```

Gambar 12. Pengujian traceroute dari client ke Internet

Hasil traceroute pada gambar 12 dapat peneliti simpulkan bahwa permintaan *traffic* dari *client* ke internet juga akan dilewatkan melalui jalur R1 (192.168.100.1) → R4 (10.14.14.4) → R5 (10.45.45.5) → Internet.



Gambar 13. Tampilan visual link utama

Berikutnya peneliti akan mencoba simulasi untuk memutuskan jalur utama, seolah-olah *link down* atau sedang terjadi permasalahan pada jalur utama. Hasilnya adalah sebanyak 25 paket ICMP ping yang dikirimkan oleh *client* menuju ke Internet (google.com) hanya akan terjadi *unreachable* sebanyak 2 kali sebelum ping kembali reply dengan rata-rata nilai *latency* 21ms dan nilai paket *loss* adalah 0% terlihat seperti pada gambar 14.

```

C:\Users\ovi>ping google.com -t

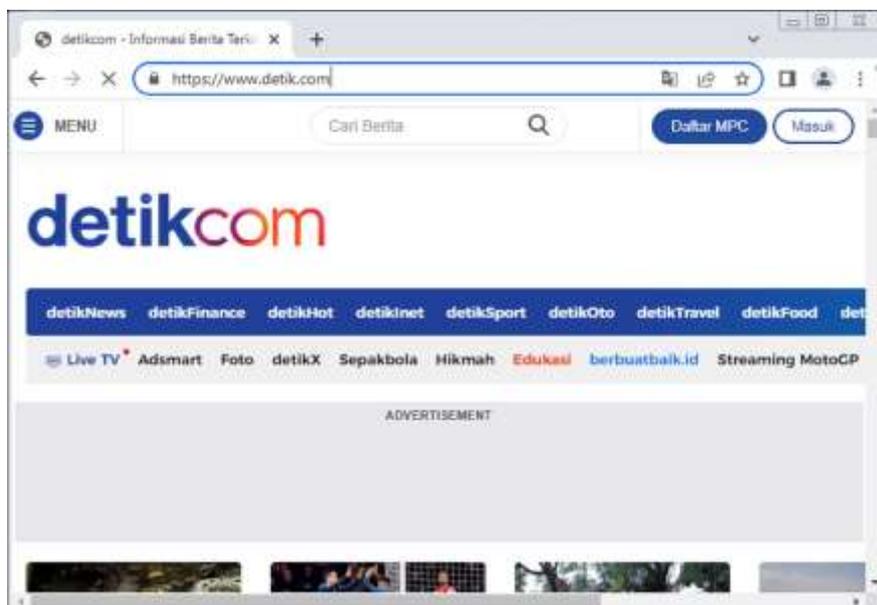
Pinging google.com [142.251.10.113] with 32 bytes of data:
Reply from 142.251.10.113: bytes=32 time=20ms TTL=54
Reply from 142.251.10.113: bytes=32 time=21ms TTL=54
Reply from 142.251.10.113: bytes=32 time=22ms TTL=54
Reply from 142.251.10.113: bytes=32 time=22ms TTL=54
Reply from 142.251.10.113: bytes=32 time=20ms TTL=54
Reply from 142.251.10.113: bytes=32 time=20ms TTL=54
Reply from 192.168.100.1: Destination net unreachable.
Reply from 192.168.100.1: Destination net unreachable.
Reply from 142.251.10.113: bytes=32 time=22ms TTL=53
Reply from 142.251.10.113: bytes=32 time=23ms TTL=53
Reply from 142.251.10.113: bytes=32 time=23ms TTL=53
Reply from 142.251.10.113: bytes=32 time=21ms TTL=53
Reply from 142.251.10.113: bytes=32 time=22ms TTL=53
Reply from 142.251.10.113: bytes=32 time=23ms TTL=53
Reply from 142.251.10.113: bytes=32 time=21ms TTL=53
Reply from 142.251.10.113: bytes=32 time=22ms TTL=53
Reply from 142.251.10.113: bytes=32 time=25ms TTL=53
Reply from 142.251.10.113: bytes=32 time=22ms TTL=53
Reply from 142.251.10.113: bytes=32 time=22ms TTL=53
Reply from 142.251.10.113: bytes=32 time=24ms TTL=53

Ping statistics for 142.251.10.113:
    Packets: Sent = 25, Received = 25, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 20ms, Maximum = 25ms, Average = 21ms
Control-C
^C
C:\Users\ovi>

```

Gambar 14. Hasil ping dari *Client* ketika jalur utama putus

Peneliti juga melakukan pengujian traffic browsing HTTPS dari *client* menuju ke Internet dengan hasil browsing tetap aman ketika sedang terjadi perpindahan jalur dari utama ke jalur cadangan.



Gambar 15. Pengujian akses browser dari client

Setelah dilakukan pengujian, terbukti bahwa OSPF akan melakukan kalkulasi link untuk memilih jalur utama berdasarkan nilai total *cost* terkecil. Dengan melakukan penjumlahan di tiap *out-interface router* yang dilalui. Ketika jalur utama mengalami gangguan maka permintaan traffic data dari client ketika menuju ke internet akan dipindah ke jalur cadangan. Hal ini sebagai

pengembangan dari penelitian yang telah dilakukan oleh [6] yang mengimplementasikan OSPF sebagai pengganti *static route*.

5. Simpulan

Implementasi OSPF pada jaringan *failover* dapat dibangun dengan baik serta ketika terjadi permasalahan pada link utama, maka permintaan dari *client* ketika menuju ke internet akan dilewatkan melalui jalur cadangan. OSPF menentukan jalur utama berdasarkan nilai total *cost* terkecil, nilai total *cost* didapat dari penjumlahan *cost* di tiap *out-interface router*. Pada saat melakukan pengujian ping dengan beban rata-rata ketika jalur berpindah hasil yang didapat adalah 21ms dan hanya 2 kali *unreachable* dan tidak mengalami kendala pada saat client melakukan browsing.

Beberapa kekurangan yang masih memerlukan rencanan pengembangan di penelitian selanjutnya adalah:

1. Menerapkan keamanan pada *router-router* OSPF berupa autentikasi antar link, sehingga dapat mengamankan komunikasi antar router di OSPF, pada penelitian ini belum menerapkan autentikasi pada *router-router* OSPF.
2. Perluanya diterapkan konsep multi area untuk membagi jumlah router sehingga dapat mengurangi *resource* CPU maupun RAM di setiap *router* yang berada pada area tersebut.
3. Ada beberapa *dynamic routing protocol* lain nya yang bisa digunakan salah satu nya adalah BGP (*Border Gateway Protocol*), RIP (*Routing Information Protocol*), EIGRP (*Enhanced Interior Gateway Protocol*) yang bisa digunakan sebagai alternatif OSPF.
4. Lebih baik menggunakan Loadbalance pada jaringan OSPF karena kedua jalur dapat digunakan secara bersamaan, dan ketika salah satu jalur mengalami gangguan, *traffic* data akan dilarikan ke jalur lainnya.

Daftar Referensi

- [1] R. Gatra and B. Sugiantoro, "Analysis of Backbone UIN Sunan Kalijaga Network Development in Yogyakarta Using Comparison of Static Routing Protocol and OSPF Dynamic Routing," *Proc. Int. Conf. Sci. Eng. (ICSE-UIN-SUKA 2021)*, vol. 211, pp. 130–137, 2022, doi: 10.2991/aer.k.211222.021.
- [2] O. K. Sulaiman, "Analisis Perbandingan Penggunaan Metric Cost dan Bandwidth Pada Routing Protocol OSPF," *Sink. J. dan Penelit. Tek. Inform.*, vol. 1, no. April 2017, pp. 7–12, 2017, [Online]. Available: <http://jurnal.polgan.ac.id/index.php/sinkron/article/view/28>
- [3] H.A. Musril, "Penerapan Open Shortest Path First (Ospf) Untuk Menentukan Jalur Terbaik Dalam Jaringan". *Jurnal Elektro dan Telekomunikasi Terapan (e-Journal)*, vol. 4, no. 1, pp. 421-421, 2017.
- [4] A.S.C. Polii, "Simulasi Kinerja Waktu Konvergensi Protokol Routing Open Shortest Path First (OSPF) Pada Jaringan Kampus. Tesis, Program Pascasarjana Universitas Hasanuddin Makassar, 2013.
- [5] Khaing Khaing Wai, "Analysis of RIP, EIGRP, and OSPF Routing Protocols in a Network," *Int. J. Trend Sci. Res. Dev.*, vol. 3, no. 5, pp. 2484–2487, 2019, doi: <https://doi.org/10.31142/Dijtsrd27928>.
- [6] I. Nurhaida, D. Ramayanti, and I. Nur, "Performance Comparison based on Open Shortest Path First (OSPF) Routing Algorithm for IP Internet Networks," *Commun. Appl. Electron.*, vol. 7, no. 31, pp. 12–25, 2019, doi: 10.5120/cae2019652838.
- [7] A. Najim, R. Sami, and S. Abbas, "Performance Evaluation of Common Routing Protocols RIP , OSPF , IGRP Evaluación de rendimiento de protocolos de," *Rev. Aus 25*, no. August, pp. 106–112, 2019, doi: 10.4206/aus.2019.n25.9.
- [8] M. H. Kabir, M. A. Kabir, M. S. Islam, M. G. Mortuza, and M. Mohiuddin, "Performance Analysis of Mesh Based Enterprise Network Using RIP, EIGRP and OSPF Routing Protocols," *Eng. Proc.*, no. November, p. 47, 2022, doi: 10.3390/ecsa-8-11285.
- [9] M. Handika, H. Herwin, D. Haryono, and R. Muzawi, "Implementation of Dynamic Routing With Ospf (Open Shortest Path First) Technique in Local Area Network (Lan) At Smkn 2 Teluk Kuantan," *J. Teknol. Dan Open Source*, vol. 4, no. 1, pp. 65–70, 2021, doi: 10.36378/jtos.v4i1.1382.
- [10] R. D. Marcus, E. P. Wibowo, and M. Muksin, "Penerapan Open Shortest Path Fisrt (OSPF) untuk Membangun Jaringan Berskala Besar Berbasis Mikrotik," *Briliant J. Ris. dan Konseptual*, vol. 3, no. 3, p. 319, 2018, doi: 10.28926/briliant.v3i3.204.

-
- [11] E. G. W. P. S. D. Krisnaningsih, "Penerapan Ospf Routing, De-Militarized Zone, Dan Firewall Pada Mikrotik Routerboardtm Dinas Komunikasi Dan Informatika Depok," *J. Sist. Inf.*, vol. Volume.2, J. Sist. Inf., pp. 59–67, 2015, [Online]. Available: <http://hdl.handle.net/11311/551109>
- [12] N. Hayati and R. T. K. Sari, "Configuration and Analysis Link-State Open Shortest Path First Routing Performance Multi-Area," *SMARTICS J.*, vol. 7, no. 1, pp. 14–20, 2021.
- [13] A. P. N. Permana and R. Firmansyah, "Distribusi Jaringan Menggunakan Routing Ospf Dengan Metode Redistribution," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 1, pp. 519–532, 2018, doi: 10.24176/simet.v9i1.2030.
- [14] D. I. Mudhoep, Linawati, and Oka Saputra, "Kombinasi Protokol Routing OSPF dan BGP dengan VRRP, HSRP, dan GLBP," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 1, pp. 1–10, 2021, doi: 10.22146/jnteti.v10i1.942.
- [15] Y. Jiang, P. Guo, and M. Guo, "A Network Topology Discovery Model Based on OSPF," *Atl. Press*, vol. 3, no. Jimec 2018, pp. 283–286, 2018, doi: 10.2991/jimec-18.2018.60.
- [16] H. Nurwarsito and A. R. Sindunata, "Optimization of hello interval in OSPF routing protocol performance on mesh network topology," *EECCIS 2020 - 2020 10th Electr. Power, Electron. Commun. Control. Informatics Semin.*, pp. 222–225, 2020, doi: 10.1109/EECCIS49483.2020.9263434.
- [17] A. Triwerdaya, D. Trianto Nugrahadi, M. I. Mazdadi, I. Budiman, and A. R. Arrahimi, "Journal of Data Science and Software Engineering Implementation of Load Balance ECMP Between BGP and OSPF Using Dual Connection (Aji) | 110 Implementation Of Load Balance Equal Cost Multi Path (Ecmp) Between Routing Protocol Border Gateway Protocol (BGP)," *J. Data Sci. Softw. Eng.*, vol. 01, no. 2, pp. 110–118, 2020.
- [18] P. Perdana and D. V. S. Y. Sakti, "Implementasi Dynamic Routing OSPF dengan Metode Single Area Pada SMK Budi Mulia Tangerang," *J. IDEALIS*, vol. 2, no. 1, pp. 407–414, 2019, [Online]. Available: <http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/article/view/1434/728>
- [19] C. B. Waluyo, "Analisis kinerja Routing OSPF dan EIGRP dengan Teknik Redistribution," *Conf. Senat. STT Adisutjipto Yogyakarta*, vol. 6, pp. 167–176, 2020, doi: 10.28989/senatik.v6i0.428.