

Analisis Performa *Open Shortest Path First Load Balancing* dengan Metode *Cost Manipulation*

Performance Analysis of *Open Shortest Path First Load Balancing* with *Cost Manipulation Method*

Mochamad Wahyudi¹, Firmansyah²

¹Universitas Bina Sarana Informatika, Indonesia

²Universitas Nusa Mandiri, Indonesia

Informasi Artikel

Genesis Artikel:

Diterima, 21 April 2022

Direvisi, 09 Mei 2022

Disetujui, 16 Juni 2022

Kata Kunci:

Cost Manipulation

Load Balancing

Open Short Path First

Packet Loss

Keywords:

Cost Manipulation

Load Balancing

Open Short Path First

Packet Loss

ABSTRAK

Quality of Service (QoS) di dalam sebuah layanan jaringan menjadi faktor terpenting untuk memastikan kapasitas transfer paket data. Salah satunya pemilihan protokol *routing* yang akan digunakan. *Routing Protokol Open Short Path First (OSPF)* menggunakan metode *Cost Manipulation* mampu menjadi sebuah alternatif solusi untuk memastikan QoS di dalam layanan jaringan dikarenakan metode *Cost Manipulation* mampu memilih jalan terbaik menuju *network* tujuan tanpa mempertimbangkan kembali metrik yang seharusnya, baik *shortest path* ke *network* tujuan ataupun *bandwidth*-nya. Hasil pengujian *traceroute* sebelum pengimplementasian *OSPF cost manipulation* didapatkan hanya menggunakan 1 (satu) *single line* saja dan *packet loss* yang didapatkan saat terjadinya *link failure* dengan pengiriman 907 *packet* data adalah 1,4 *packet loss*. Sedangkan setelah pengimplementasi *OSPF cost manipulation* dapat menggunakan 2 (dua) *dual line* sebagai *load balancing* dan *packet loss* yang didapatkan menurun dengan hasil rata-rata sebesar 0,6 *packet loss* dan pengimplementasian *cost manipulation* mampu membagi transfer paket data dengan sama rata.

ABSTRACT

Quality of Service (QoS) in a network service is the most important factor to ensure data packet transfer capacity. One of them is the routing protocol that will be used. The Open Short Path First (OSPF) Routing Protokol using the Cost Manipulation method can be an alternative solution to ensure QoS in the network because the Cost Manipulation method is able to choose the best path to the destination network without reconsidering the metrics that should be, either the shortest path to the destination network or bandwidth -his. The results of the traceroute test before the implementation of OSPF cost manipulation were obtained using only 1 (one) line and the packet loss obtained during a link failure with sending 907 packets of data was 1.4 packet loss. Meanwhile, after implementing OSPF cost manipulation, you can use two dual lines as load balancing and the packet loss obtained decreases with an average result of 0.6 packet loss and implementation cost manipulation is able to divide data packet transfers equally.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Penulis Korespondensi:

Firmansyah,

Program Studi Sistem Informasi,

Universitas Nusa Mandiri,

Email: firmansyah.fmy@nusamandiri.ac.id

1. PENDAHULUAN

Kualitas performa layanan jaringan menjadi faktor utama terhadap kinerja lalu lintas layanan. Untuk membangun layanan jaringan yang baik tak luput dari protokol routing yang digunakan. Protokol routing menentukan bagaimana router dapat saling berkomunikasi satu sama lain, menyebarkan informasi yang memungkinkan mereka untuk memilih rute yang akan digunakan [1, 2]. Dalam protokol routing *Open Short Path First* (OSPF) menggunakan *cost* dan *bandwidth* sebagai metrik yang digunakan dalam jaringan untuk melihat jalur rute di kecepatan cepat, *throughput* terbaik, lebih sedikit *jitter* dan *packet loss* secara berurutan untuk mengumpulkan informasi dalam topologi jaringan [3]. Kegagalan transfer paket data dalam suatu jaringan menjadi sebuah ancaman yang besar, baik disebabkan oleh kegagalan *link* maupun beban *traffic* yang padat [4–7]. *Load Balancing* merupakan sebuah teknik yang digunakan untuk melakukan distribusi beban *traffic* pada dua ataupun lebih jalur konektivitas secara seimbang, dengan tujuan mengoptimalkan jaringan, memaksimalkan *throughput*, serta memperkecil terjadinya *delay* pada jaringan dan meminimalisir terjadinya *overload* terhadap salah satu jalur koneksi [8–13].

Pada hasil penelitian sebelumnya didapatkan, Penerapan *cost manipulation* pada jaringan *Software Defined Networking* (SDN) memiliki pengaruh terhadap kinerja jaringan. Perbandingan skenario jaringan SDN dan konvensional pada dua parameter (QoS dan waktu konvergensi) menggunakan *cost manipulation* [8]. Hasil yang didapatkan *protocol routing* melakukan pemilihan jalur terbaik berdasarkan *cost* dan *bandwidth* untuk menuju target. Terjadi perbedaan antara penelitian yang dilakukan sebelumnya dengan penelitian ini, dimana *cost* dan alokasi *bandwidth* bukanlah menjadi faktor utama didalam pemilihan jalur terbaik. Sedangkan pada penelitian lainnya didapatkan OSPF memiliki rata-rata nilai *cost* yang hampir sama. Tanpa melakukan *cost manipulation* jika terjadi *link failure* membutuhkan waktu 5 hingga 15 detik untuk melakukan *failover interface* [9]. Sedikit perbedaan hasil penelitian sebelumnya dengan penelitian ini dimana pada *routing protokol* OSPF menggunakan *cost manipulation* jika terjadi *link failure* membutuhkan waktu 5,4ms. Serta pada penelitian lainnya, didapatkan hasil *Routing protokol* MPLS OSPF menghasilkan kinerja baik pada parameter *delay* dengan nilai *delay* terbesar yang didapat 11 milidetik dan memiliki penurunan *throughput* pada saat dilakukan gangguan *link failure* [10]. Pengujian kinerja *routing protokol* OSPF dikategorikan sangat baik setelah melakukan beberapa pengiriman paket dan didapatkan *packet loss* yang didapatkan dengan rata-rata 2,09% [11]. Perbedaan dengan penelitian ini, jika terjadi *link failure* maka *packet loss* yang didapatkan sebesar 3 *packet loss* dengan rata-rata *packet loss* yang terjadi sebesar 0,6 *packet*. Pengujian lainnya adalah pengujian lama waktu melakukan konvergensi kembali ketika terjadi perubahan jaringan atau *link* utama jalur paket putus, pengujian ini berdasarkan *client* adalah *streaming video* didapatkan protokol OSPF menggunakan *cost* sebagai metriknya [14]. Perbedaan signifikan pada penelitian ini terdapat pada penerapan metode yang digunakan. Dimana protokol routing OSPF pada dasarnya menggunakan *cost* dan *bandwidth* sebagai matrik untuk menentukan jalur terbaik didalam suatu layanan jaringan. Namun, pada penelitian ini *routing protokol* OSPF dengan menggunakan *cost manipulation* tidak mempertimbangkan *cost* sebagai matriknya didalam pemilihan jalurnya.

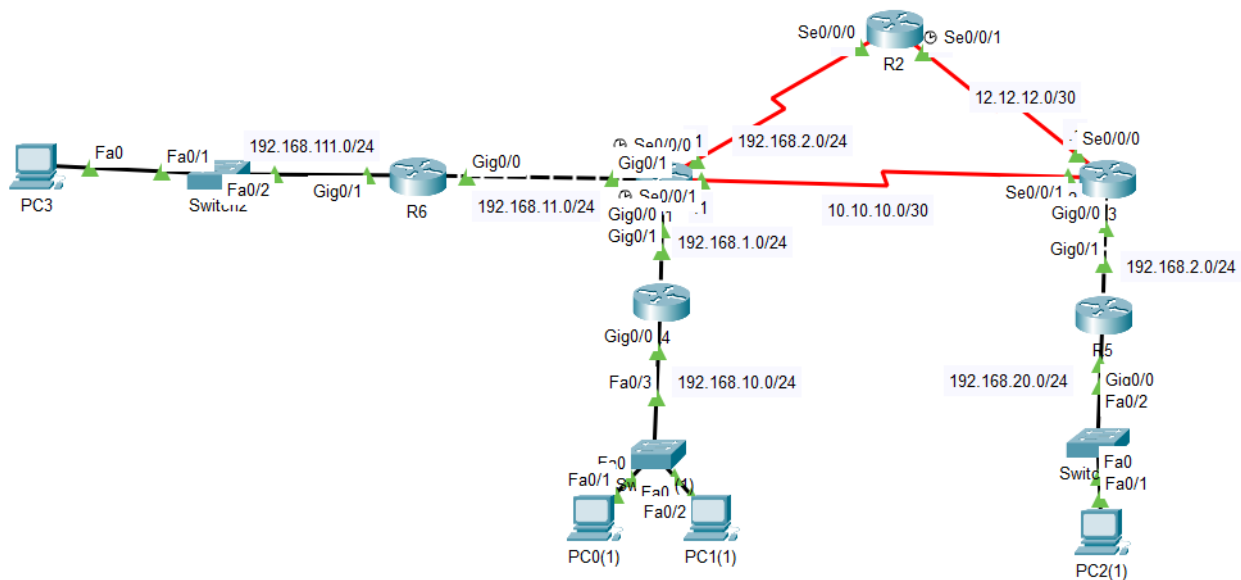
Dengan menggunakan OSPF *Cost manipulation* mampu melakukan *dual link* untuk memaksimalkan layanan *load balancing* yang mampu membagi beban jaringan sama rata sehingga mengurangi beban pada salah satu router, serta *load balancing* mampu melakukan *update table routing* secara otomatis dan mampu menjaga stabilnya layanan jaringan [7, 14]. *Cost manipulation* merupakan salah satu solusi untuk dapat menjaga layanan jaringan tetap stabil. Pengujian ini membandingkan hasil parameter QoS berdasarkan jumlah *cost* yang dilewatkan paket data ke *server*. Pengujian sebelumnya adalah pengujian lama waktu router melakukan konvergensi kembali ketika terjadi perubahan jaringan atau *link* utama jalur paket putus, pengujian ini berdasarkan *client* adalah *streaming video* didapatkan protokol OSPF menggunakan *cost* sebagai metriknya. Sedikit berbeda dengan penelitian tersebut penelitian ini protokol OSPF tidak mempertimbangkan *cost* sebagai matrik. Metode routing MPLS OSPF menunjukkan hasil yang baik dalam menangani gangguan, dengan nilai parameter QoS yang tergolong dalam kategori baik. Nilai *throughput* didapat mengalami penurunan besar nilai pada kondisi gangguan *link failure* [10] [15]. Sedangkan pada penelitian ini metode yang digunakan untuk menjaga QoS layanan jaringan menggunakan protokol routing OSPF dengan melakukan *cost manipulation*. Beberapa penelitian lainnya routing protokol OSPF menggunakan metode *Access Control List* dan tanpa *Access Control List*. Pengimplementasian *cost manipulation load balancing* didalam protokol OSPF diharapkan dapat melakukan *backup* secara otomatis jika terdapat salah satu *interface* mengalami *link failure* serta pengimplementasian metode *Cost Manipulation* diharapkan mampu memilih jalan terbaik ke *network* tujuan tanpa mempertimbangkan kembali metrik yang seharusnya, hal ini yang menjadikan landasan penelitian analisis performa *Open Shortest Path First Load Balancing* dengan metode *cost manipulation*.

Penelitian ini menyajikan empat bagian, yaitu: pendahuluan, bagian ini menjelaskan latar belakang masalah, kesenjangan penelitian, dan tujuan penelitian. bagian kedua adalah metode penelitian pada bagian ini peneliti menjelaskan bagaimana cara mengumpulkan informasi atau data dan melakukan pengolahan data terhadap data yang telah diperoleh. bagian ketiga Hasil dan Analisis pada tahap ini menjelaskan hasil pengujian dan validasi data menggunakan Metode *Cost Manipulation*, dan bagian ke Empat adalah kesimpulan pada bagian ini menjelaskan kesimpulan penelitian mengenai hasil penelitian menggunakan *Cost Manipulation* dan saran untuk penelitian selanjutnya.

2. METODE PENELITIAN

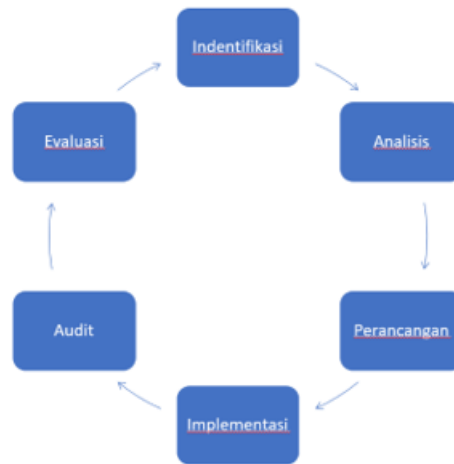
Dalam penelitian analisis performa *open shortest path first load balancing* dengan metode *cost manipulation* penulis menggunakan bantuan software simulasi Cisco *Packet Tracer* untuk membuat simulasi jaringan berjalan yang dijalankan secara virtualisasi namun tidak merubah dan mengurangi seperti fitur aslinya. Pada penelitian analisis performa *open shortest path first load balancing* dengan metode *cost manipulation* peneliti menggunakan 3 buah device router yang diimplementasikan menggunakan protokol routing OSPF. Penelitian ini bersifat instrumental dan mengumpulkan semua data dengan menghasilkan data secara deskriptif dan menampilkan hasil *load balancing* sebelum maupun sesudah pengimplementasian OSPF *cost manipulation*.

Terlihat pada gambar 1 merupakan skema jaringan yang digunakan pada penelitian Analisis Performa *Open Shortest Path First Load Balancing* dengan metode *cost manipulation*. Pengujian dilakukan untuk mendapatkan hasil Analisa kinerja *load balancing* menggunakan metode *cost manipulation*. Jika sebelumnya tanpa adanya *load balancing* protokol routing ospf akan menggunakan jalur terbaiknya dari PC0 menuju PC2 melalui R1 menuju R3. Namun, setelah pengimplementasian *load balancing* PC0 menuju PC2 dapat melalui jalur yang berbeda yaitu dari R1 menuju R2 lalu menuju R3 dengan mempertimbangkan layanan jaringan dibagi dengan sama rata.



Gambar 1. Skema Jaringan

Metode penelitian yang digunakan dalam penelitian ini menggunakan *The Security Policy Development Life Cycle (SPDLC)*, yang memiliki enam (6) tahapan, yaitu: Identifikasi, digunakan untuk melakukan pengidentifikasian terhadap permasalahan keamanan didalam jaringan komputer. Analisis, pada tahapan ini penulis melakukan percobaan untuk mengetahui resiko dan ancaman didalam keamanan jaringan sebelum dan sesudah pengimplementasian *load balancing* dengan metode *cost manipulation*. Perancangan, tahapan ini penulis melakukan perancangan jaringan *load balancing* dengan metode *cost manipulation*. Implementasi, penulis melakukan konfigurasi *load balancing* dengan metode *cost manipulation*, dan melakukan tahapan uji konektivitas terhadap layanan jaringan. Audit, digunakan untuk melakukan pemeriksaan terhadap layanan jaringan yang telah diimplementasikan dengan mempertimbangkan sebelum dan setelah pengimplementasian *load balancing* dengan metode *cost manipulation*. Evaluasi, melakukan evaluasi layanan jaringan yang telah diterapkan untuk mendapatkan hasil dari kualitas jaringan.



Gambar 2. Metode Penelitian

3. HASIL DAN ANALISIS

Skenario pengujian analisis performa *open shortest path first load balancing* dengan metode *cost manipulation* dapat dilakukan dengan cara melakukan uji konektivitas jaringan khususnya jaringan yang berada pada R1 menuju R3. Pengujian dilakukan dengan menggunakan protokol ICMP untuk mengetahui *hop count* yang dilalui sebelum dan sesudah pengimplementasian *open shortest path first load balancing* dengan metode *cost manipulation* serta melihat *packet loss* saat terjadinya *link failure*.

Tabel 1. Spesifikasi IP Address

Device	Interface	IP Address
R1	G0/0	192.168.1.1/24
	Se0/0/0	11.11.11.1/30
	Se0/0/1	10.10.10.1/30
R2	Se0/0/0	11.11.11.2/30
	Se0/0/1	12.12.12.2/30
R3	G0/0	192.168.2.124
	Se0/0/0	12.12.12.1/30
	Se0/0/1	10.10.10.2/30
PC0	NIC	192.168.1.2/24
PC1	NIC	192.168.1.3/24
PC2	NIC	192.168.2.2/24

Berdasarkan Tabel 1, dapat dijelaskan didalam jaringan terdapat tiga (3) *device* router dan tiga (3) *device* PC. Setiap router menggunakan *interface* serial 0/0/0 dan serial 0/0/1 untuk saling berkomunikasi antara R1 dengan R2 maupun dengan R3, sedangkan *interface* Gigabit digunakan untuk menggabungkan dengan jaringan lokal yang terdapat pada R1 dan R3. Konfigurasi OSPF Setelah mengetahui spesifikasi IP Address yang terdapat pada Tabel 1 dan topologi jaringan yang digunakan pada gambar 1 langkah selanjutnya ialah melakukan konfigurasi OSPF terhadap R1, R2 maupun R3. Pengimplementasian routing OSPF terhadap ketiga *device* router menggunakan OSPF ID-1 dan menggunakan *single area* yaitu area 10.

```

R1(config)#routerospf1
R1(config-router)#network10.10.10.0.0.3area10
R1(config-router)#network11.11.11.0.0.3area10
R1(config-router)#network192.168.1.0.0.255area10
  
```

```
R2(config)#routerospf1
R2(config-router)#network11.11.11.0.0.3area10
R2(config-router)#network12.12.12.0.0.3area10

R3(config)#routerospf1
R3(config-router)#network10.10.10.0.0.3area10
R3(config-router)#network12.12.12.0.0.3area10
R3(config-router)#network192.168.2.0.0.255area10
```

Terlihat pada gambar 2 merupakan *cost* yang didapatkan terhadap *interface* s0/0/0 sebesar 64 dan *interface* s0/0/1 memiliki *cost* yang sama besar yaitu sebesar 64 pada R1. Nilai *cost* yang sama didapatkan pula pada R2 dan R3 yang berhubungan langsung dengan router.

```
GigabitEthernet0/0 is up, line protocol is up
Internet address is 192.168.1.1/24, Area 10
Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 192.168.1.1, Interface address 192.168.1.1
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Index 1/1, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)
Serial0/0/0 is up, line protocol is up
Internet address is 11.11.11.1/30, Area 10
Process ID 1, Router ID 192.168.1.1, Network Type POINT-TO-POINT, Cost: 64
Transmit Delay is 1 sec, State POINT-TO-POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Index 2/2, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 12.12.12.2
Suppress hello for 0 neighbor(s)
Serial0/0/1 is up, line protocol is up
Internet address is 10.10.10.1/30, Area 10
Process ID 1, Router ID 192.168.1.1, Network Type POINT-TO-POINT, Cost: 64
Transmit Delay is 1 sec, State POINT-TO-POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:04
Index 3/3, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 192.168.2.1
Suppress hello for 0 neighbor(s)
```

Gambar 3. Show IP OSPF Interface R1

Konfigurasi OSPF *load balancing cost manipulation* pengimplementasian OSPF *load balancing cost manipulation* diharapkan mampu untuk mengoptimalkan *Quality of Service* didalam layanan jaringan dengan melakukan pembagian beban layanan jaringan dengan sama rata. Dimana jika PC0 akan melakukan *transfer* paket data menuju PC2 dapat menggunakan layanan pada R2 ataupun R3. Untuk pengimplementasian OSPF *load balancing cost manipulation* dapat menggunakan perintah.

```
R1(config)#ints0/0/0
R1(config-if)#ipospf cost1
R1(config-if)#ints0/0/1
R1(config-if)#ipospf cost2
```

```
R2(config)#ints0/0/0
R2(config-if)#ipospf cost1
```

```
R2(config-if)#ints0/0/1
R2(config-if)#ipospfcost1
```

```
R3(config)#ints0/0/0
R3(config-if)#ipospfcost1
R3(config-if)#ints0/0/1
R3(config-if)#ipospfcost2
```

Dijelaskan pada tabel 2 merupakan nilai *cost* yang didapatkan pada masing-masing *interface* router, baik pada R1, R2 maupun R3. Dengan memilikinya nilai *cost* yang sama antar router, maka pengiriman paket data dari PC0 yang terdapat pada R1 menuju PC2 yang terdapat pada R3 hanya dapat berjalan secara satu arah tanpa dapat melewati R2. Hal ini nantinya akan berakibat terjadinya *overload* lalu lintas jaringan dikarenakan tidak dibaginya beban layanan jaringan dengan sama rata. Konfigurasi OSPF *load balancing cost manipulation* pengimplementasian *ospf load balancing cost manipulation* diharapkan mampu untuk mengoptimalkan *Quality of Service* didalam layanan jaringan dengan melakukan pembagian beban layanan jaringan dengan sama rata. Dimana jika PC0 akan melakukan transfer paket data menuju PC2 dapat menggunakan layanan pada R2 ataupun R3. Untuk pengimplementasian OSPF *load balancing cost manipulation* dapat menggunakan perintah.

Tabel 2. Spesifikasi OSPF Cost

Device	Interface	Cost
R1	G0/0	1
	Se0/0/0	64
	Se0/0/1	64
R2	Se0/0/0	64
	Se0/0/1	64
R3	G0/0	1
	Se0/0/0	64
	Se0/0/1	64

Terlihat pada gambar 4 merupakan *table routing* didalam R1, terlihat R1 terdapat 5 rute yang terhubung secara langsung dengan R1 dalam membentuk *table routing*-nya. Hal ini akan berakibat terjadinya *overload* lalu lintas jaringan dikarenakan tidak dibaginya beban layanan jaringan dengan sama rata. Dengan menggunakan metode *cost manipulation* yang telah dilakukan terhadap R1.

```

I   10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   10.10.10.0/30 is directly connected, Serial0/0/1
L   10.10.10.1/32 is directly connected, Serial0/0/1
L   11.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   11.11.11.0/30 is directly connected, Serial0/0/0
L   11.11.11.1/32 is directly connected, Serial0/0/0
O   12.0.0.0/30 is subnetted, 1 subnets
    12.12.12.0/30 [110/128] via 11.11.11.2, 00:29:20, Serial0/0/0
    [110/128] via 10.10.10.2, 00:29:20, Serial0/0/1
O   192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.1.0/24 is directly connected, GigabitEthernet0/0
L   192.168.1.1/32 is directly connected, GigabitEthernet0/0
O   192.168.2.0/24 [110/65] via 10.10.10.2, 00:29:20, Serial0/0/1

```

Gambar 4. Tabel Routing R1

dijelaskan pada gambar ??, jika sebelumnya *cost* yang didapatkan pada R1 sebesar 64 *cost* kini berubah menjadi 1 *cost* untuk *interface* s0/0/0 dan 2 *cost* untuk *interface* s0/0/1. Hal ini dikarenakan telah dilakukannya pengimplentasian *cost manipulation* didalam *protokol routing* OSPF. Dengan melakukan *cost manipulation* diharapkan layanan jaringan dapat dibagi dengan sama rata dari R1 menuju R3 (*load balancing*) dikarenakan memiliki nilai *cost* yang sama.

```

Serial0/0/0 is up, line protocol is up
Internet address is 11.11.11.1/30, Area 10
Process ID 1, Router ID 192.168.1.1, Network Type POINT-TO-POINT, Cost: 1
Transmit Delay is 1 sec, State POINT-TO-POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:06
Index 2/2, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 12.12.12.2
Suppress hello for 0 neighbor(s)
Serial0/0/1 is up, line protocol is up
Internet address is 10.10.10.1/30, Area 10
Process ID 1, Router ID 192.168.1.1, Network Type POINT-TO-POINT, Cost: 2
Transmit Delay is 1 sec, State POINT-TO-POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:04
Index 3/3, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 192.168.2.1
Suppress hello for 0 neighbor(s)
    
```

Gambar 5. Show IP OSPF Interface R1 Manipulation

Tabel 3 merupakan hasil pengimplementasian *cost manipulation* secara keseluruhan pada R1, R2 dan R3. Nantinya PC0 yang berada dibawah R4 melalui R1 akan melakukan transfer paket data dapat melalui R2 terlebih dahulu ataupun langsung menuju R3 untuk menuju PC2 yang berada pada R3. Uji konektifitas OSPF tanpa *load balancing* yang pertama kali dilakukan adalah melakukan *tracert router* dari PC0 yang terdapat pada R1 menuju PC2 yang berada pada R3. *Tracert* dilakukan untuk mendapatkan *hops count* yang dilalui antara PC0 dengan PC2 terlihat pada gambar 6.

Tabel 3. Spesifikasi OSPF Cost Manipulation

Device	Interface	Cost
R1	G0/0	1
	Se0/0/0	1
	Se0/0/1	2
R2	Se0/0/0	1
	Se0/0/1	1
R3	G0/0	1
	Se0/0/0	1
	Se0/0/1	2

dijelaskan pada gambar 6 merupakan hasil *hop count* yang didapat adalah PC0 melalui 192.168.1.1 yang digunakan sebagai alokasi alamat *gateway* didalam jaringan PC0 lalu menuju ke IP Address 10.10.10.2 yang merupakan alokasi IP Address pada R3 dan menuju ke 192.168.2.2 yang merupakan alamat tujuan.

```

C:\>tracert 192.168.2.2

Tracing route to 192.168.2.2 over a maximum of 30 hops:

  1  0 ms    0 ms    0 ms    192.168.10.1
  2  0 ms    0 ms    0 ms    192.168.1.1
  3  1 ms    1 ms    1 ms    10.10.10.2
  4  3 ms    1 ms    3 ms    12.12.12.1

Trace complete.
    
```

Gambar 6. Tracert Tanpa Load Balancing PC0

Tabel 4 menjelaskan total *cost* yang digunakan saat terjadinya sebuah proses *transfer* paket data dari PC0 menuju PC2. Nilai

cost yang didapatkan dari PC0 menuju PC2 jika melewati R1 dan R3 hanya sebesar 194 *cost* hal ini berbeda jika menggunakan layanan pada R1-R2-R3 dengan menggunakan *cost* yang jauh lebih besar yaitu 258 *cost*. Maka dari itu hasil *tracert route* yang didapatkan pada tabel 4 melalui IP 10.10.10.2 yang berada pada R3 dengan *cost* yang lebih kecil hal ini sejalan dengan cara kerja *routing protokol* OSPF yang bekerja berdasarkan *matrix* dengan mempertimbangkan *cost* dan *bandwidth* didalam layanan jaringan. Selanjutnya peneliti melakukan uji konektifitas *link failure* OSPF tanpa *load balancing* pengujian dilakukan dengan menguji konektifitas layanan jaringan jika terjadinya *link failure* didalam layanan jaringan. Skenario pengujian dilakukan dengan cara melakukan pemutusan akses dari R1 menuju R3 untuk mendapatkan hasil analisa jika terjadi *link failure* saat proses *transfer* paket data didalam layanan jaringan.

Tabel 4. Total Cost PC0 ke PC2 Sebelum *Laod Balancing*

Source	Cost R1		Cost R3		Destination
	G0/0	Se0/0/1	Se0/0/1	G0/0	
PC0	1	64	1	1	PC2

Source	Cost R1		Cost R2		Cost R3		Destination
	G0/0	Se0/0/1	Se0/0/0	Se0/0/1	Se0/0/1	G0/0	
PC0	1	64	64	64	1	1	PC2

Terlihat pada tabel 5 merupakan hasil uji konektifitas *link failure* OSPF tanpa adanya *load balancing*. Dari total 5 (lima) kali pengujian sebesar 907 packet send didapatkan hasil *packet receiver* sebesar 900 packet dengan mengalami *packet loss* sebesar 7 *packet loss* saat terjadinya *link failure* dengan rata-rata *packet loss* yang terjadi sebesar 1,4 *packet* dan dengan *time average* sebesar 5,4ms. Sedangkan pada penelitian lainnya didapatkan pengimplementasian OSPF membutuhkan waktu untuk melakukan *failover* sebesar 5 hingga 15 detik [14]. Langkah selanjutnya melakukan uji konektifitas OSPF dengan *load balancing cost manipulation* dengan tujuan untuk mendapatkan hasil analisa kinerja *cost manipulation*.

Tabel 5. *Tracert* PC0 ke PC2 Melalui 10.10.10.2

Percobaan	Packet Send	Packet Receiver	Packet Loss	Time Min	Time Max	Time Average
1	37	36	1	1ms	22ms	7ms
2	52	50	2	1ms	18ms	7ms
3	103	102	1	1ms	19ms	7ms
4	203	201	2	1ms	20ms	3ms
5	512	511	1	1ms	29ms	3ms
Total	907	900	7	1ms	108ms	27ms
Rata-rata			1,4 Packet	1ms	21,6ms	5,4ms

dijelaskan pada gambar 7 merupakan hasil *tracert* yang dilakukan pada R1 menuju alamat 192.168.2.1 yang terdapat pada R3, *tracert* pertama kali *hops count* yang dilalui yaitu 10.10.10.2 yang berada pada R3 dan *tracert route* selanjutnya lalu lintas akan dialihkan ke 11.11.11.2 yang berada pada R2.

```

R1#tracert 192.168.2.1
Type escape sequence to abort.
Tracing the route to 192.168.2.1

 0  10.10.10.2    14 msec    0 msec    1 msec
R1#tracert 192.168.2.1
Type escape sequence to abort.
Tracing the route to 192.168.2.1

 0  11.11.11.2    0 msec    1 msec    1 msec

```

Gambar 7. *Tracert* Load Balancing PC0

Tabel 6 menjelaskan hasil dari sebuah analisa perbandingan OSPF sebelum dan sesudah menggunakan *load balancing cost manipulation*. Hasil yang didapatkan sebelum adanya *cost manipulation* saat terjadi pengiriman paket data dari PC0 menuju P2 jika menggunakan jalur R1;R3 memiliki *cost* 194 dan setelah pengimplementasian *cost manipulation* maka *cost* yang didapatkan dari 194 menjadi 6 *cost*. Sedangkan *cost* yang didapatkan jika menggunakan jalur R1;R2;R3 sebelum menggunakan *cost manipulation*

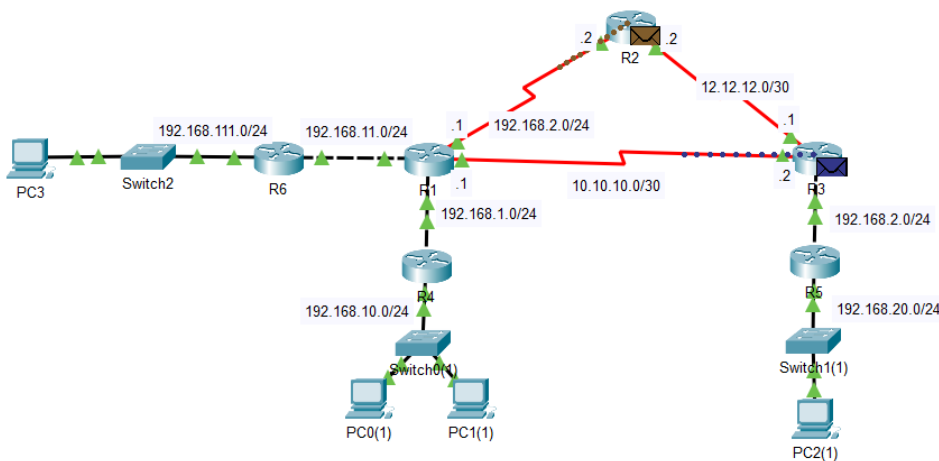
sebesar 258 dan setelah menggunakan *cost manipulation cost* menjadi 6 *cost*. Jadi setelah pengimplementasian *cost manipulation* PC0 menuju PC2 baik menggunakan jalur R1;R3 ataupun R1;R2;R3 sama-sama mendapatkan *cost* sebesar 6 *cost* hal ini dapat memicu terjadinya *load balancing* dengan melakukan *cost manipulation* didalam jaringan protokol OSPF.

Tabel 6. Total Cost PC0 ke PC2 Setelah Load Balancing

Analisa	Source	Cost R1		Cost R3		Destination
		G0/0	Se0/0/1	Se0/0/1	G0/0	
Sebelum	PC0	1	64	1	1	PC2
Sesudah	PC0	1	2	1	1	PC2

Analisa	Source	Cost R1		Cost R2		Cost R3		Destination
		G0/0	Se0/0/1	Se0/0/0	Se0/0/1	G0/0	G0/0	
Sebelum	PC0	1	64	64	1	1	1	PC2
Sesudah	PC0	1	1	1	1	1	1	PC2

Terlihat pada gambar 8 merupakan hasil pengiriman paket data menggunakan *software* simulasi *cisco packet tracer* dari PC0 menuju PC2 yang melakukan pengiriman paket data sebanyak 2 (dua) kali secara bersamaan dengan tujuan untuk mendapatkan *traffic* lalu lintas yang akan dilalui. *Traffic* lalu lintas jaringan yang didapatkan melakukan pembagian *traffic* dengan sama rata antara R1 menuju ke R3, ataupun dari R1 menuju R2 dan menuju ke R3. Analisa ini menjawab bahwa telah suksesnya pengimplementasian *cost manipulation* pada jaringan OSPF yang mampu melakukan *load balancing* didalam proses *transfer* paket layanan data didalam jaringan.



Gambar 8. Traffic Load Balancing

Hasil yang didapatkan saat terjadinya *link failure* setelah pengimplementasian *cost manipulation* dapat dilihat pada tabel 7. Dari total 5 (lima) kali pengujian sebesar 907 *packet send* didapatkan hasil *packet receiver* sebesar 903 *packet* hasil yang lebih baik jika sebelum adanya *load balancing* dengan hasil *packet receiver* hanya sebesar 900 *packet*, serta dengan mengalami *packet loss* hanya sebesar 3 *packet loss* saat terjadinya *link failure* dengan rata-rata *packet loss* yang terjadi sebesar 0,6 *packet* dengan hasil yang lebih baik dibandingkan tanpa adanya *cost manipulation* dengan hasil *packet loss* rata-rata sebesar 1,4 *packet* dan dengan *time average* sebesar 4,2ms.

Tabel 7. Tracert PC0 ke PC2

Percobaan	Packet Send	Packet Receiver	Packet Loss	Time Min	Time Max	Time Average
1	37	36	1	1ms	22ms	7ms
2	52	51	1	1ms	18ms	7ms
3	103	102	0	1ms	19ms	3ms
4	203	203	0	1ms	11ms	2ms
5	512	511	1	1ms	17ms	2ms
Total	907	903	3	1ms	87ms	21ms
Rata-rata			0,6 Packet	1ms	17,4ms	4,2ms

4. KESIMPULAN

Pengimplementasian OSPF *cost manipulation* dapat mengurangi *packet loss* saat terjadinya *link failure*. Sebelum pengimplementasian *cost manipulation* didapatkan rata-rata sebesar 1,4 *packet loss* yang terjadi sedangkan setelah pengimplementasian ospf *cost manipulation packet loss* yang didapatkan menurun menjadi rata-rata sebesar 0,6 *packet*. Serta pengimplementasian OSPF *cost manipulation* mampu meningkatkan layanan jaringan terlihat dari hasil *packet receiver* sebelum pengimplementasian sebesar 900 berbanding 903 *packet* dari total 907 *packet send* yang dilakukan. Saran untuk penelitian selanjutnya dalam pengembangan dan pengimplementasian routing protokol OSPF *cost manipulation* dapat mempertimbangkan efisiensi dan efektifitas lalu lintas layanan jaringan. Serta dapat memanfaatkan protokol routing OSPFv3 dengan menggunakan IP Address versi 6.

UCAPAN TERIMA KASIH

Bagian ucapan terimakasih adalah opsional. Sumber dana penelitian dapat diletakkan di sini.

REFERENSI

- [1] M. Waseel and J. Arora, "Improvement of Convergence Delay in OSPF by Using Backup Path Technique," *Dogo Rangsang Research Journal*, vol. 11, no. 01, pp. 70–74, 2021.
- [2] A. Manzoor, M. Hussain, and S. Mehrban, "Performance Analysis and Route Optimization: Redistribution Between EIGRP, OSPF & BGP Routing Protocols," *Computer Standards and Interfaces*, vol. 68, no. 1, p. 103391, 2019.
- [3] M. Kassim, M. Ismail, M. I. Yusof, and M. A. Abdullah, "Time Based Traffic Policing and Shaping Algorithms on Campus Network Internet Traffic," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2-3, pp. 135–140, 2017.
- [4] N. F. Nadhirah Norazlan, R. A. Rahman, M. Kassim, and A. R. Mahmud, "Virtual Private Network Load Balancing Using OSPF Routing," in *ISCAIE 2020 - IEEE 10th Symposium on Computer Applications and Industrial Electronics*, 2020, pp. 164–169.
- [5] R. A. Purnama and F. Firmansyah, "Redundancy Gateway Menggunakan Metode Failover dan Load Sharing Gateway," *Indonesian Journal of Computer Science*, vol. 9, no. 1, pp. 22–31, 2020.
- [6] Firmansyah, S. Dewi, and R. a. Purnama, "Quality of Service Gateway Load Balancing Protocol Message Digest Algorithm 5 Authentication untuk Peningkatan Kualitas Jaringan," *Jurnal Teknik Informatika*, vol. 5, no. November, pp. 45–50, 2020.
- [7] Firmansyah, M. Wahyudi, R. Adi Purnama, and L. Pujiastuti, "Performance Analysis of Routing Enhanced Interior Gateway Routing Protocol Load Balancing for IPv6," *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019*, vol. 6, no. 4, pp. 4–9, 2019.
- [8] R. Adrian, A. Dahlan, and K. Anam, "OSPF Cost Impact Analysis on SDN Network," in *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017*, 2018, pp. 198–201.
- [9] A. Fiade, M. A. Agustian, and S. U. Masrurroh, "Analysis of Failover Link System Performance in OSPF, EIGRP, RIPV2 Routing Protocol with BGP," in *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*, 2019, pp. 1–7.
- [10] A. Budiman, A. Sucipto, and A. R. Dian, "Analisis Quality of Service Routing MPLS OSPF Terhadap Gangguan Link Failure," *Techno.Com*, vol. 20, no. 1, pp. 28–37, 2021.
- [11] T. M. Diansyah, D. Handoko, I. Faisal, A. Yuniarti, K. Chiuloto, and R. Liza, "Design Analysis of OSPF (Open Shortest Path First) Routing by Calculating Packet Loss of Network WAN (Wide Area Network)," *Journal of Physics: Conference Series*, vol. 1361, no. 1, pp. 1–8, 2019.
- [12] D. A. Figueroa, "Experience on a Course Using Packet Tracer and Cisco Real Devices for Practices IP Routing and IP Addressing," *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, vol. 1, no. 13, pp. 1–10, 2021.
- [13] A. Hidayat, N. Widiyasono, and ..., "Analisis Efektifitas Open Shortest Path First (OSPF) dengan dan Tanpa Menggunakan Access Control List (ACL)," *SAIS—Scientific Articles of ...*, vol. 4, no. 1, pp. 83–88, 2021.

-
- [14] N. A. Putra, V. Riyanto, G. Wijaya, and N. Herlinawati, "Firewall Design Using Access Control List Method as Data Filtering," *Jurnal Mantik*, vol. 5, no. 3, pp. 1684–1693, 2021.
- [15] L. Z. A. Mardedi and K. Marzuki, "Network Rancang Bangun Jaringan Komputer LAN Berdasarkan Perbandingan Kinerja Routing Protokol EIGRP dan Routing Protokol OSPF," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 202–210, 2019.

