

Analisa QoS (Quality of Service) Zeroshell pada Mekanisme Load Balancing dan Failover

Praja Risnaldy, Indri Neforawati

Program Studi Teknik Multimedia dan Jaringan

Jurusan Teknik Informatika dan Komputer

Politeknik Negeri Jakarta

Depok, Indonesia

praja.risnaldy.tik16@mhs.wpnj.ac.id, indri.neforawati@tik.pnj.ac.id

Diterima : 15 April 2020. Disetujui: 22 Mei 2020. Dipublikasikan: 26 Mei 2020

Abstract – The bandwidth requirements are increasing along with the number of internet users are increasing. The problem that is often faced by users in accessing the internet is ISPs (Internet Service Provider) are often down and internet connections tend to be slow. So, it is necessary to hold a multi-connection using two different ISPs by utilizing load balancing and failover. Load balancing and failover are implemented in zeroshell 3.9.3 using the weight round-robin method. Testing is done with an internet speed tester, downloading videos from YouTube, and disconnecting one of the connection lines. Testing with Internet speed tester is to get the throughput value of systems with load balancing and systems without load balancing. Video download testing is done to get the QoS value, namely throughput, packet loss, delay, and jitter. Testing by disconnecting one of the connection lines to get the function of failover as well as calculating the delay value of the connection line transfer. The result is that load balancing and failover systems can function on zeroshell, systems that implement load balancing have higher throughput values compared to ISP 1 and ISP 2 which do not implement load balancing systems, the value of throughput, packet loss, and delay has a "very good" TIPHON category with index 4, jitter value has a "good" TIPHON category with index 3, and delay value of failover system has a "bad" TIPHON category with index 1.

Keywords: Failover; Load Balancing; QoS; Zeroshell

Abstrak – Kebutuhan bandwidth semakin hari semakin meningkat seiring bertambahnya jumlah pengguna internet. Masalah yang sering dihadapi pengguna dalam mengakses internet adalah ISP yang sering down dan koneksi internet yang cenderung lambat. Sehingga, perlu diadakan multikoneksi menggunakan dua ISP yang berbeda dengan pemanfaatan load balancing dan failover. Load balancing dan failover diimplementasikan pada zeroshell 3.9.3 dengan metode weight round-robin. Pengujian dilakukan dengan internet speed tester, download video dari YouTube, serta memutuskan salah satu jalur koneksi. Pengujian dengan internet speed tester untuk mendapatkan nilai throughput dari sistem load balancing dan sistem tanpa load balancing. Pengujian download video dilakukan untuk mendapatkan nilai QoS, yaitu throughput, packet loss, delay, dan jitter. Pengujian dengan memutuskan salah satu jalur koneksi untuk mendapatkan fungsi dari failover sekaligus menghitung nilai delay dari perpindahan jalur koneksi. Hasil penelitian ini adalah sistem load balancing dan failover dapat berfungsi pada zeroshell, sistem yang menerapkan load balancing memiliki nilai throughput yang lebih tinggi dibandingkan dengan ISP 1 dan ISP 2 yang tidak menerapkan sistem load balancing, nilai QoS throughput, packet loss, dan delay memiliki kategori TIPHON “sangat bagus” dengan indeks 4, nilai QoS jitter memiliki kategori TIPHON “bagus” dengan indeks 3, serta nilai delay sistem failover memiliki kategori TIPHON “Jelek” dengan indeks 1.

Kata Kunci: Failover; Load Balancing; QoS; Zeroshell

I. PENDAHULUAN

Internet menjadi tempat bagi setiap pengguna untuk saling terhubung dalam jaringan yang sangat luas [1]. Untuk bisa menggunakan internet dibutuhkan ISP (*Internet Service Provider*) sebagai penyedia layanan internet, baik untuk sambungan lokal maupun internasional. Sambungan lokal menyediakan interkoneksi antar ISP di Indonesia agar dapat saling terhubung satu sama lain, sedangkan sambungan internasional menyediakan *bandwidth* yang dapat digunakan untuk terhubung ke *backbone* internasional [2].

Kebutuhan *bandwidth* semakin hari semakin meningkat seiring bertambahnya jumlah pengguna internet. Tahun 2005 sampai tahun 2019 jumlah pengguna internet di dunia selalu bertambah setiap tahun [3]. Jumlah pengguna di Asia pada pertengahan tahun 2019 sebanyak 55% dari total seluruh pengguna internet di dunia yang berjumlah kurang lebih 4,5 miliar [4]. Peningkatan jumlah pengguna internet tidak didukung dengan mutu jaringan internet yang sebanding [5]. Masalah yang sering dihadapi pengguna untuk mengakses internet adalah ISP yang sering *down* dan koneksi internet yang cenderung lambat. Sehingga, perlu diadakan multi koneksi menggunakan dua jalur ISP yang

berbeda untuk dapat digunakan secara bersamaan agar didapat *bandwidth* yang besar demi memenuhi kebutuhan internet yang besar dengan memanfaatkan mekanisme *load balancing* dan *failover* [6].

Load balancing berfungsi agar trafik dapat berjalan lebih baik, memaksimalkan *throughput*, memperkecil *delay*, dan menghindari *overload* pada salah satu jalur koneksi [7]. *Failover* berfungsi agar jalur koneksi tetap tersedia jika terjadi gangguan pada jalur koneksi utama. Implementasi *load balancing* dan *failover* diterapkan pada zeroshell yang merupakan routerOS berbasis *linux*. Zeroshell digunakan karena memiliki beberapa fitur, seperti *load balancing* dan *failover* serta dalam menjalankan zeroshell dibutuhkan spesifikasi *hardware* yang kecil, *minimum requirements hardware* [8].

Penelitian mengenai *load balancing* dan *failover* sudah cukup banyak dilakukan, seperti penelitian mengenai implementasi *load balancing* dua ISP menggunakan mikrotik [9], implementasi *high availability server* menggunakan metode *load balancing* dan *failover* [10], implementasi *load balancing* dan *failover* berbasis mikrotik *router* [6], dan penelitian mengenai aplikasi *load balancing* dengan metode Nth [7].

Penelitian-penelitian tersebut membahas tentang implementasi *load balancing* dan *failover*. Penelitian ini juga membahas hal yang sama, namun diimplementasikan pada zeroshell sekaligus dilakukan analisis QoS (*Quality of Service*) yang akan disesuaikan dengan standar TIPHON. Tujuan penelitian ini adalah mengevaluasi kualitas jaringan pada *load balancing* dan *failover* dengan analisis QoS pada Zeroshell.

II. TINJAUAN PUSTAKA

A. Load Balancing

Load Balancing berfungsi agar trafik dapat berjalan lebih baik, memaksimalkan *throughput*, memperkecil *delay*, dan menghindari *overload* pada salah satu jalur koneksi [7]. *Throughput*, *performance*, *scalability*, waktu respon, pemanfaatan sumber daya, dan *fault tolerance* adalah beberapa parameter pengukuran yang dapat digunakan untuk mengevaluasi *load balancing* [11].

Hingga saat ini, terdapat berbagai macam algoritma untuk *load balancing*, seperti *round-robin*, *least-connection*, dan *least loaded*. *Round-robin* merupakan algoritma *load balancing* yang dilakukan dengan memberi giliran masing-masing *switch* atau *router* secara berurutan dan sirkular, sehingga tidak mementingkan beban *traffic* dari setiap *switch* atau *router*. *Least connection* bekerja dengan memilih *switch* atau *router* dengan koneksi *outgoing* (keluar) yang paling rendah. Sedangkan *least loaded* akan

memilih *switch* atau *router* dengan beban kerja yang paling rendah [1].

B. Failover

Failover merupakan kemampuan sistem untuk dapat berpindah *gateway* secara manual maupun otomatis jika salah satu *gateway* mengalami masalah sehingga dapat menjadi *backup* untuk sistem yang mengalami masalah dalam mengakses internet [8].

C. Zeroshell

Zeroshell adalah distribusi berbasis Linux yang didedikasikan untuk implementasi *router* dan *firewall* yang sepenuhnya dapat diatur melalui antarmuka *web*. Fitur yang disediakan antara lain *load balancing* dan *failover* untuk beberapa koneksi internet, VPN *Site to Site* dan VPN *Host to Site*, *captive portal* untuk *hotspot*, *firewall*, QoS, autentikasi dan akuntansi RADIUS, hingga pelacakan, dan pencatatan koneksi jaringan [12].

Zeroshell memiliki fitur untuk menyeimbangkan trafik yang dilalui, *NetBalancer*. *NetBalancer* adalah fitur dari zeroshell yang dapat memanfaatkan beberapa *gateway* internet menjadi seimbang, fitur ini disebut juga dengan *load balancing*. *Load balancing* pada zeroshell menggunakan metode *weight round robin* [8].

Metode *weight round robin* didasari oleh *round-robin scheduling*. *Weight round robin* merupakan teknik penjadwalan yang dapat diterapkan dalam berbagai bidang untuk pemakaian sumber daya bersama-sama pada sebuah komputer atau jaringan. Zeroshell ditujukan untuk *server* dan perangkat *embedded* agar dapat membuat layanan jaringan menjadi lebih baik [8].

D. Parameter QoS

Parameter QoS yang digunakan:

1. Throughput

Throughput adalah kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses, diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [13].

TABEL I. THROUGHPUT VERSI TIPHON

Kategori	Throughput (bps)	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	< 25	1

2. Delay (latency)

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti, atau waktu proses yang lama [13].

TABEL II. DELAY VERSI TIPHON

Kategori	Delay (ms)	Indeks
Sangat Bagus	< 150	4
Bagus	150 s/d 300	3
Sedang	300 s/d 450	2
Jelek	< 450	1

3. Packet loss

Packet loss merupakan persentase hilangnya paket saat pengiriman data [13].

TABEL III. PACKET LOSS VERSI TIPHON

Kategori	Packet Loss (%)	Indeks
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

4. Jitter

Jitter diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket di akhir perjalanan *jitter*. *Jitter* lazimnya disebut variasi *delay* [13].

TABEL IV. JITTER VERSI TIPHON

Kategori	Jitter (ms)	Indeks
Sangat Bagus	0	4
Bagus	0 s/d 75	3
Sedang	75 s/d 125	2
Jelek	125 s/d 225	1

III. METODE PENELITIAN

Metode penelitian yang dilakukan secara bertahap dan saling berkaitan satu sama lain pada penelitian ini, yaitu:

1. Pengumpulan Data

Metode pengumpulan data pada penelitian ini dilakukan dengan metode literatur dan eksperimen. Metode literatur dilakukan dengan cara membaca dan membandingkan referensi tentang QoS, *load balancing*, *failover*, zeroshell, dan referensi lain yang terkait. Metode eksperimen dilakukan dengan cara melakukan penelitian secara langsung terhadap objek yang diteliti dengan melakukan pengujian *load balancing* dan *failover*.

2. Perancangan

Perancangan dilakukan untuk mendapatkan rancangan sistem yang akan diimplementasikan pada tahap implementasi. Rancangan sistem berupa diagram blok untuk sistem *load balancing* dan *failover* pada zeroshell.

3. Implementasi

Implementasi dilakukan sesuai dengan rancangan sistem yang telah ditentukan pada tahap perancangan.

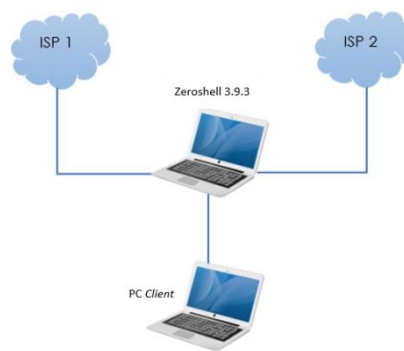
4. Pengujian

Pengujian dilakukan setelah implementasi sistem berhasil dilakukan. Pengujian pada sistem dilakukan dengan 3 cara, yaitu pengujian dengan

internet speed tester, pengujian dengan *download video*, serta pengujian dengan melakukan *ping* kemudian mematikan salah satu jalur koneksi.

IV. PERANCANGAN

Dua ISP yang berasal dari dua modem yang berbeda menjadi penyedia internet yang akan diakses oleh PC *client* melalui laptop. PC zeroshell mengatur koneksi *client* dengan dua ISP menggunakan mekanisme *load balancing* dan *failover*.



Gambar 1. Diagram Blok Load Balancing dan Failover

Gambar 1 merupakan diagram blok dari *load balancing* dan *failover* pada zeroshell di mana zeroshell 3.9.3 diterapkan sebagai *load balancer* dan *failover*.

V. IMPLEMENTASI

A. Hardware untuk Load Balancing dan failover



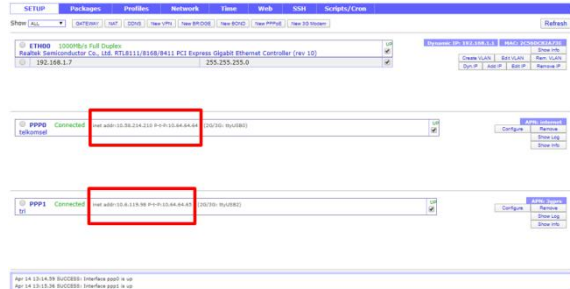
Gambar 2. Hardware untuk Load Balancing dan Failover

Gambar 2 merupakan rangkaian *hardware* yang digunakan untuk *load balancing* dan *failover* pada zeroshell. Berikut penjelasan *hardware* yang digunakan:

1. PC zeroshell, laptop yang dipasang zeroshell 3.9.3. Laptop yang digunakan adalah Asus dengan RAM 10 GB.

2. Dua modem GSM Huawei E220 menggunakan kartu Telkomsel sebagai ISP 1 dan kartu Tri sebagai ISP 2.
3. PC *client*, laptop yang terhubung dengan PC zeroshell menggunakan kabel LAN.

B. Pengaturan IP Address



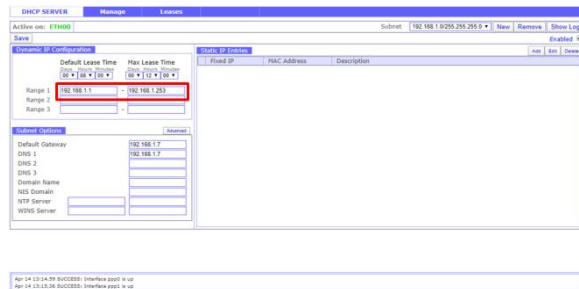
Gambar 3. Antarmuka Web Zeroshell

Gambar 3 merupakan antarmuka *web* zeroshell. Terdapat 3 *interface* yang digunakan, yaitu:

1. ETH00, PC *client* yang terhubung dengan PC zeroshell.
2. PPP0, GSM Huawei E220 menggunakan kartu Terkomsel dan berada di USB 2 sebagai ISP 1 dengan *IP address dynamic*,
3. PPP1, GSM Huawei E220 menggunakan kartu Tri dan berada di USB 1 sebagai ISP 2 dengan *IP address dynamic*.

C. Pengaktifan DHCP Server

Fungsi DHCP *server* adalah pada saat penambahan PC *client*, tidak perlu melakukan konfigurasi *IP address* secara manual karena *IP address* akan didapat secara otomatis sesuai *range* yang sudah diatur.

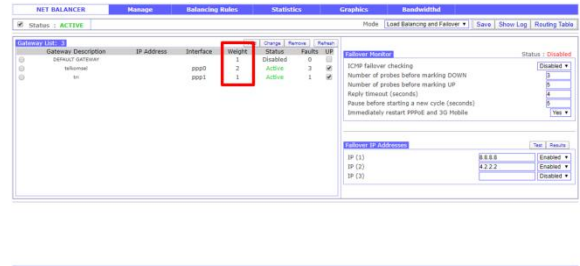


Gambar 4. DHCP Server pada Zeroshell

Gambar 4 merupakan range untuk Dynamic IP, yaitu 192.168.1.1 – 192.168.1.253.

D. Pengaktifan Net Balancer

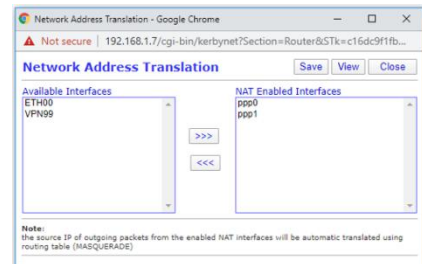
NetBalancer menggunakan algoritma *weight round-robin*, menimbang dari beban pada *gateway*. *Failover* memindahkan *gateway* ketika terjadi masalah pada salah satu *gateway*.



Gambar 5. Net Balancer pada Zeroshell

E. Pengaktifan NAT

NAT diatur untuk menghubungkan PC *client* dengan internet.



Gambar 6. NAT pada Zeroshell

VI. PENGUJIAN

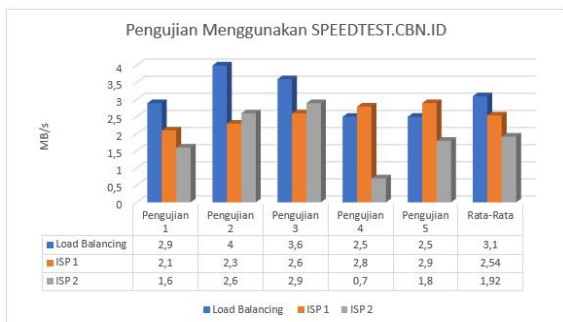
Pengujian *load balancing* pada zeroshell dilakukan dengan menguji kecepatan internet melalui situs *internet speed tester*, yaitu situs speedtest.cbn.id dan speedtest.net. Selain melakukan pengujian kecepatan internet, pengujian *load balancing* juga dilakukan dengan cara melakukan *download* video yang berukuran 61,1 MB dari YouTube. Selama proses *download* berlangsung, dilakukan *capture packet* menggunakan aplikasi *wireshark* untuk menghitung nilai *throughput*, *packet loss*, *delay*, dan *jitter*.

Pengujian *failover* dilakukan dengan melakukan *ping* kemudian mematikan salah satu jalur koneksi atau ISP untuk mendapatkan fungsi dari *failover*. Selama proses *ping* berlangsung, dilakukan *capture packet* menggunakan aplikasi *wireshark* untuk menghitung nilai *delay*. Masing-masing pengujian dilakukan sebanyak lima kali untuk mendapatkan hasil pengujian yang optimal.

A. Hasil Pengujian Load Balancing Menggunakan Internet Speed Test

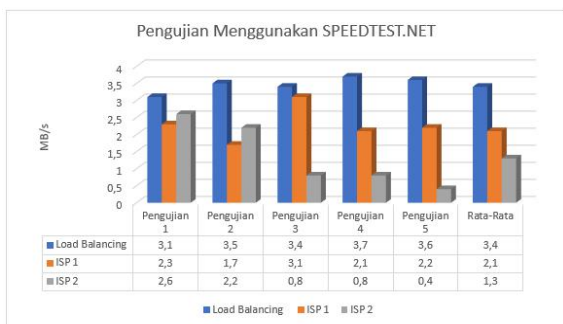
Pengujian pertama dilakukan dengan menguji kecepatan internet melalui situs speedtest.cbn.id sebanyak lima kali. Selanjutnya dilakukan pengujian kecepatan internet melalui situs speedtest.net sebanyak lima kali. Data yang didapat berupa kecepatan internet di setiap pengujian (*throughput*). Data hasil pengujian pada situs *internet speed tester*

dijumlahkan, lalu dibagi dengan banyak pengujian yang dilakukan untuk mendapat rata-rata *throughput*.



Gambar 7. Hasil Pengujian dengan Speedtest.cbn.id

Dari data hasil pengujian di atas, pengujian 1, pengujian 2, dan pengujian 3, sistem *load balancing* memiliki nilai *throughput* lebih besar dibandingkan dengan ISP 1 dan ISP 2. Sedangkan pengujian 4 dan pengujian 5, *load balancing* memiliki nilai *throughput* lebih kecil dibandingkan dengan ISP 1 dan ISP 2. Setelah dihitung nilai rata-rata *throughput*, didapatkan *load balancing* memiliki nilai *throughput* lebih besar dibandingkan dengan ISP 1 dan ISP 2.



Gambar 8. Hasil Pengujian dengan Speedtest.net

Dari data hasil pengujian di atas, *load balancing* memiliki nilai *throughput* lebih besar dibandingkan dengan ISP 1 dan ISP 2 pada setiap pengujian. Sehingga, pada rata-rata, *load balancing* memiliki nilai *throughput* yang paling besar.

Berdasarkan pengujian yang dilakukan dengan kedua *internet speed tester*, sistem *load balancing* dapat berfungsi pada zeroshell yang terbukti dari nilai *throughput* sistem *load balancing* yang lebih besar dibandingkan dengan nilai *throughput* kedua ISP yang tidak menggunakan *load balancing*.

B. Hasil Pengujian Load Balancing dengan Download Video

Pengujian *load balancing* dengan *download video* dilakukan untuk mendapatkan nilai QoS, yaitu *throughput*, *packet loss*, *delay*, dan *jitter*.

TABEL V. HASIL PENGUJIAN DENGAN *DOWNLOAD VIDEO*

QoS	Pengujian ke-					Rata-Rata
	1	2	3	4	5	
<i>Throughput</i> (Mbps)	0,25	0,29	0,25	0,28	0,26	0,27
<i>Packet Loss</i> (%)	0	0	0	0	0	0
<i>Delay</i> (ms)	4,63	4,05	5,23	4,24	4,44	4,52
<i>Jitter</i> (ms)	5,47	4,83	5,91	5	5,41	5,32

Data di atas merupakan hasil pengujian yang dilakukan sebanyak lima kali. Nilai rata-rata setiap parameter QoS disesuaikan dengan standar TIPHON untuk menentukan kategori dan indeks dari hasil pengujian (TABEL VI).

TABEL VI. HASIL PENGUJIAN SESUAI STANDAR TIPHON

QoS	Nilai	Standar TIPHON	
		Kategori	Indeks
<i>Throughput</i> (Mbps)	0,27	Sangat Bagus	4
<i>Packet Loss</i> (%)	0	Sangat Bagus	4
<i>Delay</i> (ms)	4,52	Sangat Bagus	4
<i>Jitter</i> (ms)	5,32	Bagus	3

Dari data di atas, berdasarkan standar TIPHON, nilai *throughput* memiliki kategori “sangat bagus” dengan indeks 4. Nilai *packet loss* memiliki kategori “sangat bagus” dengan indeks 4. Nilai *delay* memiliki kategori “sangat bagus” dengan indeks 4. Nilai *jitter* memiliki kategori “bagus” dengan indeks 3.

C. Hasil Pengujian Failover

```

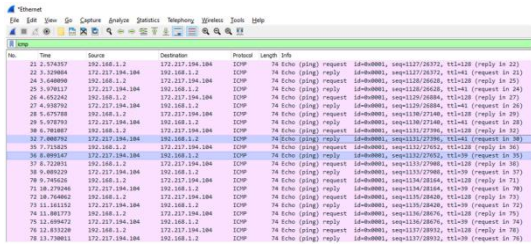
Select Command Prompt
c:\Users\TMJ\PJ>ping www.google.com -t

Pinging www.google.com [172.217.194.104] with 32 bytes of data:
Reply from 172.217.194.104: bytes=32 time=754ms TTL=41
Reply from 172.217.194.104: bytes=32 time=330ms TTL=41
Reply from 172.217.194.104: bytes=32 time=286ms TTL=41
Reply from 172.217.194.104: bytes=32 time=303ms TTL=41
Reply from 172.217.194.104: bytes=32 time=307ms TTL=41
Reply from 172.217.194.104: bytes=32 time=383ms TTL=39
Reply from 172.217.194.104: bytes=32 time=367ms TTL=39
Reply from 172.217.194.104: bytes=32 time=533ms TTL=39
Reply from 172.217.194.104: bytes=32 time=397ms TTL=39
Reply from 172.217.194.104: bytes=32 time=898ms TTL=39

Ping statistics for 172.217.194.104:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 286ms, Maximum = 898ms, Average = 455ms
Control-C
^C
    
```

Gambar 9. Pengujian *Failover* pada Zeroshell

Gambar 9 merupakan pengujian *failover* yang dilakukan dengan melakukan *ping* ke *web google.com* dengan pengiriman data sebesar 32 *byte*. Saat proses *ping* berlangsung, satu jalur koneksi dimatikan untuk mendapatkan fungsi dari *failover*. TTL (*Time To Live*) terlihat berubah dari 41 menjadi 39 menandakan pengiriman paket yang terjadi sudah pada jaringan yang berbeda.



Gambar 10. Hasil Capture Packet dengan Wireshark

Nilai delay didapatkan dari pengukuran dua time since reference, yaitu 8,099147 second – 7,008792 second = 1,090355 second.

TABEL VII. HASIL PENGUJIAN DELAY

QoS	Pengujian ke-					Rata-Rata
	1	2	3	4	5	
Delay (ms)	1090	2150	2270	1070	3050	1926

Data di atas merupakan hasil pengujian yang dilakukan sebanyak lima kali. Nilai rata-rata setiap parameter QoS disesuaikan dengan standar TIPHON untuk menentukan kategori dan indeks dari hasil pengujian (TABEL VIII).

TABEL VIII. NILAI DELAY SESUAI STANDAR TIPHON

QoS	Nilai	Standar TIPHON	
		Kategori	Indeks
Delay (ms)	1926	Jelek	1

Dari data di atas, berdasarkan standar TIPHON, nilai delay memiliki kategori “jelek” dengan indeks 1.

VII. KESIMPULAN DAN SARAN

Berdasarkan pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem load balancing dan failover dapat berfungsi pada zeroshell 3.9.3.
2. Sistem yang menerapkan load balancing memiliki throughput lebih besar dibandingkan dengan sistem yang tidak menerapkan load balancing.
3. Nilai throughput sistem load balancing pada zeroshell 3.9.3 memiliki kategori TIPHON “sangat bagus” dengan indeks 4.
4. Nilai packet loss sistem load balancing pada zeroshell 3.9.3 memiliki kategori TIPHON “sangat bagus” dengan indeks 4.
5. Nilai delay sistem load balancing pada zeroshell 3.9.3 memiliki kategori TIPHON “sangat bagus” dengan indeks 4.
6. Nilai jitter sistem load balancing pada zeroshell 3.9.3 memiliki kategori TIPHON “bagus” dengan indeks 3.
7. Nilai delay sistem failover pada zeroshell 3.9.3 memiliki kategori TIPHON “jelek” dengan indeks 1.

Berdasarkan hasil penelitian yang telah dilakukan, perlu dilakukan peningkatan analisis untuk mendapatkan hasil evaluasi yang lebih optimal. Berikut adalah saran yang dapat dilakukan:

1. Menambahkan skenario pengujian sistem load balancing dengan 2 ISP untuk menentukan kapasitas beban yang digunakan pada masing-masing jalur ISP saat load balancing digunakan.
2. Menggunakan standar QoS yang berbeda sebagai pembandingan standar QoS satu dengan yang lain untuk mendapatkan kategori dan indeks yang variatif.

REFERENSI

- [1] N. Fauzi, W. Yahya, and A. Bhawiyuga, “Implementasi Load Balancing Pada Server Dengan Menggunakan Algoritme Least Traffic Pada Software-Defined Network,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 3134–3141, 2018.
- [2] H. Nasser and T. Witono, “Analysis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler,” *J. Inform.*, vol. 12, no. 1, pp. 25–32, 2016.
- [3] H. Stipp, “Number of Internet Users Worldwide from 2005 to 2019,” 2019. <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide> (accessed Jan. 21, 2020).
- [4] Miniwatts Marketing Group, “World Internet Usage and population statistics,” 2019. <https://www.internetworldstats.com/stats.htm> (accessed Jan. 21, 2020).
- [5] D. Darmawan and T. Imanto, “Analisa Link Balancing dan Failover 2 Provider Menggunakan Border Gateway Protocol (BGP) Pada Router Cisco 7606s,” *J. Nas. Teknol. dan Sist. Inf.*, vol. 3, no. 3, pp. 326–333, 2017.
- [6] Suryanto, T. Prasetyo, and N. Hikmah, “Implementasi Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Dengan Failover Berbasis Mikrotik Router,” *Semin. Nas. Inov. dan Tren*, vol. 1, no. 1, pp. 230–238, 2018.
- [7] R. Rasna and A. Ashari, “Application of Load Balancing with the Nth Method on Multiple Gateway Internet Networks,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 2, pp. 159–168, 2019.
- [8] A. Frayogi, W. Yahya, and R. A. Setiawan, “Perbandingan Kinerja RouterOS Mikrotik dan Zeroshell pada Mekanisme Load Balancing Serta Failover,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 7, pp. 2689–2697, 2018.
- [9] P. Oktivasari and R. Sanjaya, “Implementasi Sistem Load Balancing Dua ISP Menggunakan Mikrotik dengan Metode Per Connection Classifier,” *Multinetics*, vol. 1, no. 2, pp. 33–37, 2015.
- [10] M. Rosalia, R. Munadi, and R. Mayasari, “Implementasi HigRosalia, M., Munadi, R., & Mayasari, R. (2016). Implementasi High Availability Server Menggunakan Metode Load Balancing dan Failover pada Virtual Web Server Cluster. E-Proceeding of Engineering, 3(3), 4496–4503.h Availability Server Menggu,” *e-Proceeding Eng.*, vol. 3, no. 3, pp. 4496–4503, 2016.
- [11] S. R.S and B. R. Yashwantrao, “Load Balancing and its Algorithms in Cloud Computing : A Survey,” *Int. J. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 95–100, 2017.
- [12] A. P. Rofiatul Lailiy Siregar, “Implementasi Jaringan Hotspot dengan Captive Portal Zeroshell dan User Management LDAP,” *J. Manaj. Inform.*, vol. 9, no. 2, pp. 87–96, 2019.
- [13] T. Pratama, M. A. Irwansyah, and Yulianti, “Perbandingan Metode PCQ, SQ, RED Dan FIFO Pada Mikrotik Sebagai Upaya Optimalisasi Layanan Jaringan Pada Fakultas

Teknik Universitas Tanjungpura,” *J. Tek. Inform. Univ. Tanjungpura*, vol. 3, no. 3, pp. 298–303, 2015.