

**IMPLEMENTATION OF LOAD BALANCING WITH PER CONNECTION CLASSIFIER AND FAILOVER AND UTILIZATION OF TELEGRAM BOT (CASE STUDY : PT TUJUH MEDIA ANGKASA)**

Ariya Pramudita<sup>\*1</sup>, Rushendra<sup>2</sup>

<sup>1,2</sup>Informatics Study Program, Faculty of Computer Science, Universitas Mercu Buana, Indonesia  
Email: <sup>1</sup>[ariyapram4625@gmail.com](mailto:ariyapram4625@gmail.com), <sup>2</sup>[rushendra@mercubuana.ac.id](mailto:rushendra@mercubuana.ac.id)

(Article received: June 28, 2023; Revision: August 19, 2023; published: February 16, 2024)

**Abstract**

For customers of PT. Seven Media Angkasa, which is engaged in providing fast online shopping services in Indonesia, definitely needs stable internet to process requests from customers. Even though it already has 2 ISPs, sometimes there are frequent downtimes which will disrupt the service process for customers who want to shop. In this case, one of the Load Balancing methods is the Per Connection Classifier (PCC) which is able to specify a packet to the gateway of a particular connection. Failover for backing up The weakness of the PCC method is Failover which can switch automatically if one of the systems fails so that it becomes a backup for the system that has failed. Added Telegram Bot as a DHCP Alert which can detect if there is a DHCP Rouge. By using the PCC method, it is able to maximize bandwidth usage and minimize the occurrence of downtime in sending or receiving data. So with the addition of the Failover method, if there is a temporary delay when many incoming requests can interfere with performance, Failover can move manually or automatically if one of the systems fails so that it becomes a backup for a failed system. If gateway 1 is disconnected, the backup gateway will replace gateway 1. If gateway 1 returns to normal, the connection path is used again to become gateway 1. Likewise with gateway 2 when it is disconnected. From testing on the speedtest.net tools, it was found that the Load Balancing applied was able to combine 2 ISPs into one, namely Download to 19.18 Mbps and Upload to 18.47 Mbps. The Telegram Bot is able to send notifications when there is a counter DHCP Server with the contents of the message successfully getting a Mac Address or unknown server from the counter DHCP server, namely DC: 2C: 6E: 81: CF: 34.

**Keywords:** bot telegram, DHCP rouge, failover, load balancing, PCC.

**IMPLEMENTASI LOAD BALANCING DENGAN PER CONNECTION CLASSIFIER DAN FAILOVER SERTA PEMANFAATAN BOT TELEGRAM (STUDI KASUS : PT TUJUH MEDIA ANGKASA )**

**Abstrak**

Pada pelanggan PT. Tujuh Media Angkasa yang bergerak di bidang penyedia layanan belanja daring cepat di Indonesia, pasti butuh internet yang stabil untuk memproses permintaan dari pelanggan. Walaupun sudah memiliki 2 ISP, terkadang sering sekali terjadinya down yang akan mengganggu proses pelayanan terhadap pelanggan yang ingin berbelanja. Dalam hal ini Salah satu metode Load Balancing yaitu Per Connection Classifier (PCC) yang mampu menspesifikasikan suatu paket menuju gateway suatu koneksi tertentu. Failover untuk membackup kelemahan dari metode PCC yaitu Failover yang dapat berpindah secara otomatis, jika salah satu sistem mengalami kegagalan sehingga menjadi backup untuk sistem yang mengalami kegagalan. Penambahan Bot Telegram sebagai DHCP Alert yang dapat mendeteksi apabila ada DHCP Rouge. Dengan menggunakan metode PCC, mampu memaksimalkan penggunaan bandwidth serta meminimalisir terjadinya downtime dalam mengirim atau menerima data. Jadi dengan ditambah metode Failover apabila ada penundaan sementara ketika banyak permintaan masuk yang dapat mengganggu kinerja, Failover dapat bergerak secara manual atau otomatis jika salah satu sistem gagal sehingga itu menjadi cadangan untuk sistem yang gagal. Jika gerbang 1 adalah terputus, gateway cadangan akan menggantikan gateway 1. Jika gateway 1 kembali normal, jalur koneksi digunakan lagi menjadi gateway 1. Begitu juga dengan gateway 2 saat itu terputus. Dari pengujian pada tools speedtest.net didapatkan bahwa Load Balancing yang diterapkan mampu menggabungkan 2 ISP menjadi satu yaitu Download menjadi 19.18 Mbps dan Upload 18.47 Mbps. Dengan Bot Telegram mampu mengirimkan notifikasi apabila ada DHCP Server tandingan dengan isi pesan berhasil mendapatkan Mac Address atau unknown server dari DHCP server tandingan yaitu DC:2C:6E:81:CF:34.

**Kata kunci:** *bot telegram, DHCP rouge, failover, load balancing, PCC.*

## 1. PENDAHULUAN

Pada era perkembangan jaringan komputer dan internet saat ini mengalami perkembangan yang sangat pesat [1]. Hal ini membawa dampak bagi aspek kehidupan serta manfaat bagi pengguna, baik dari instansi pemerintahan, perusahaan maupun perorangan akan kebutuhan teknologi informasi [2].

Pada pelanggan PT.Tujuh Media Angkasa yang bergerak di bidang penyedia layanan belanja daring cepat di Indonesia, pasti memerlukan koneksi internet yang stabil. Saat banyak permintaan dari pelanggan, maka koneksi internet akan terbebani karena harus melakukan banyak proses pelayanan yang diminta oleh para pelanggan [3]. Disini ISP yang digunakan pada pelanggan tersebut adalah vendor dari Telkom dan Biznet. Walaupun sudah memiliki 2 ISP sebagai koneksi internet, terkadang sering sekali terjadinya down yang akan mengganggu proses pelayanan terhadap pelanggan yang ingin berbelanja. Oleh karena itu perlunya suatu cara agar koneksi internet tersebut dapat stabil.

Dalam hal ini teknik *Load Balancing* bisa menjadi pilihan untuk meminimalisir gangguan koneksi tersebut [4]. *Load balancing* adalah suatu jaringan komputer yang menggunakan metode untuk mendistribusikan beban kerja pada dua atau bahkan lebih suatu koneksi jaringan secara seimbang agar pekerjaan dapat berjalan optimal dan tidak *overload* (kelebihan) beban pada salah satu jalur koneksi [5].

Salah satu metode dalam Teknik *Load Balancing* yaitu *Per Connection Classifier (PCC)* yang dapat digunakan untuk mengelompokkan trafik koneksi yang melalui router menjadi beberapa kelompok [6]. Untuk memaksimalkan metode load balancing, maka diterapkan juga teknik failover yang berguna untuk menjaga apabila link utama terjadi gangguan, maka sistem akan secara otomatis akan memfungsikan *link backup* sebagai jalur koneksi cadangan [7]. Beberapa literatur memperjelas bahwa load balance tidak akan menambah besar bandwidth yang diperoleh, tetapi hanya bertugas untuk membagi trafik dari kedua bandwidth tersebut agar dapat terpakai secara seimbang[8]. Akan tetapi disini load balance juga dapat menggabungkan dua jalur koneksi menjadi dua kali lipat dari bandwidth sebelumnya. Bahwa akan menjadi rumus matematika  $1+1=2$  yang biasanya dalam load balance  $1+1=1+1$ .

Namun, beberapa kekurangan pada jaringan adalah terputusnya *server* secara tiba-tiba atau tidak tahu kapan secara pasti akan terputus. Dari hal tersebut membutuhkan monitoring secara real-time untuk mengetahui ketika server terputus [9]. Hal tersebut bisa diatasi dengan *Telegram*, *Telegram* merupakan aplikasi chat yang bersifat open source dan memanfaatkan *API* dari *bot* tersebut maka

akan lebih mudah dalam mengirim informasi jaringan secara *real-time* [10].

Berdasarkan penelitian ini Sistem *Load Balancing* dapat membagi jalur koneksi secara merata pada ISP-1 dan ISP-2 berdasarkan paket permintaan sehingga tidak hanya membebani satu ISP saja. Penerapan sistem *Load Balancing* dapat mengatasi masalah ketika salah satu ISP mengalami pemutusan hubungan [11].

Berdasarkan penelitian ini diperoleh bahwa *Failover* dapat waktu *response time* 0 detik dan bekerja sesuai dengan fungsi *Failover* itu sendiri. Metode *PCC* mampu membuat kinerja jaringan semakin stabil dan membagi trafik secara merata. Dengan hasil metode *ECMP* pada jaringan ISP1 sebesar 23,3 Mb/s dan pada jaringan telekomunikasi 47,5 Mb/s, sedangkan metode *PCC* pada jaringan ISP1 sebesar 60,4 Mb/s dan pada ISP2 21.0 Mb/s. Pengujian *QOS* yang dilakukan diperoleh hasil: Pengujian *Bandwidth* untuk metode *PCC* 1538.3 kbps dan *ECMP* 1434.4 kbps. Pengujian *delay* pada metode *PCC* 0.00609302s dan *ECMP* 0.00671507s. Pengujian kehilangan paket terbaik pada *PCC* 0.64 dan *ECMP* 0.9 [12].

Berdasarkan pengujian yang dilakukan pada penelitian ini, dapat disimpulkan bahwa *Load Balancing PCC*, menggunakan *priority* koneksi sesuai dengan konfigurasi yang telah dibuat dan dapat membagi beban *bandwidth* ketika salah satu ISP mengalami *overload*. Ketika terjadi *overload*, proses pembagian *bandwidth* dapat dibagi secara merata, Proses *Failover* dapat memindahkan link secara otomatis ketika salah satu ISP down, dan Notifikasi jaringan via *Telegram* bekerja sesuai dengan kondisi link jaringan, saat *Traffic up* dan *down* [13].

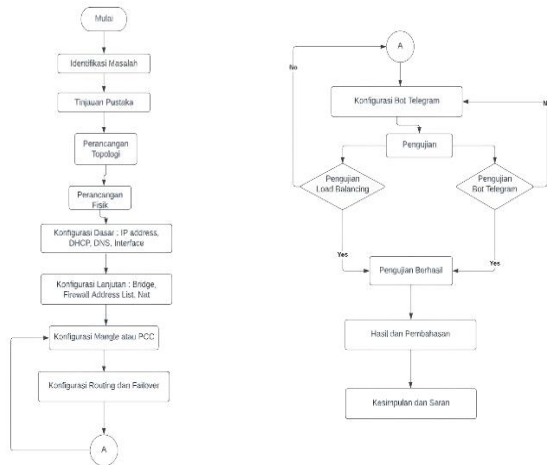
Berdasarkan penelitian menghasilkan simpulan dengan menggunakan *Bot Telegram* dapat mempermudah admin atau teknisi dalam mendapatkan informasi konektivitas jaringan secara *real-time*, dalam mendapatkan informasi user yang login dan logout kedalam jaringan dan dalam mendapatkan informasi kondisi fisik mikrotik router [14].

Berdasarkan kondisi di atas yang saya dapati saat melakukan magang dan dari berbagai penelitian yang telah saya jadikan referensi, maka dalam penelitian ini saya mengusulkan untuk mengimplementasikan *Load Balancing* sebagai pemecahan masalah yang ada diatas, dengan menerapkan metode *Per Connection Classifier (PCC)* dan *Failover* untuk membuat kestabilan koneksi Internet menggunakan *RB941-2nd HAP LITE*. Serta ditambahkan dengan *Bot Telegram* digunakan sebagai *DHCP Alert* yang dapat mendeteksi apabila ada *DHCP Rouge* yang dapat

mengganggu jaringan karena memberikan informasi yang salah terhadap *client*.

## 2. METODE PENELITIAN

Tahap penelitian pada penelitian ini terdapat beberapa tahap, yaitu:



Gambar 1. Tahapan Penelitian

Pada Gambar 1 tahapan penelitian diatas menggambarkan proses penelitian yang akan dilakukan sekaligus menggambarkan penelitian secara keseluruhan. Tahapan yang akan dilakukan yaitu:

### 2.1. Identifikasi Masalah

Pada tahapan pertama dimulai dengan melakukan identifikasi masalah menggunakan metode observasi lapangan untuk langsung bisa mengetahui secara detail masalah apa yang menjadi sumber utama[2]. Maka di dapatkan bahwa sumber masalahnya yaitu apabila koneksi utama sedang full traffic karena keterbatasan *bandwith* yang tidak mampu menampung saat banyaknya permintaan dari para pelanggan diwaktu bersamaan dan juga koneksi yang digunakan sebagai backup pun sering terjadi koneksi terputus karena *backup link* yang tidak aktif secara otomatis [22]. Dari data yang diberikan oleh pihak perusahaan dan dilakukan pengecekan, gangguan jaringan terjadi akibat dari adanya over bandwidth dimana pemakaian *bandwidth user* melebihi *bandwidth* atau kapasitas yang tersedia.

Tabel 1. Tabel Dataset

Date Time	Sensor	Status	Message
04/07/2022 12:42:37	Ether2-ISP 2 Traffic	Down	No response
24/07/2022 08:15:41	Ether2-ISP 2 Traffic	Down	No response
15/08/2022 11:57:21	Ether1-ISP 1 Traffic	Down	No response
20/10/2022 13:55:33	Ether1-ISP 1 Traffic	Down	No response
23/11/2022 00:28:12	Ether2-ISP 2 Traffic	Down	No response
25/1/2023 19:05:49	Ether2-ISP 2 Traffic	Down	No response
01/02/2023 05:28:20	Ether1-ISP 1 Traffic	Down	No response
04/03/2023 12:30:15	Ether1-ISP 2 Traffic	Down	No response
27/03/2023 15:32:03	Ether1-ISP 2 Traffic	Down	No response
12/04/2023 07:40:00	Ether1-ISP 1 Traffic	Down	No response

### 2.2. Tinjauan Pustaka

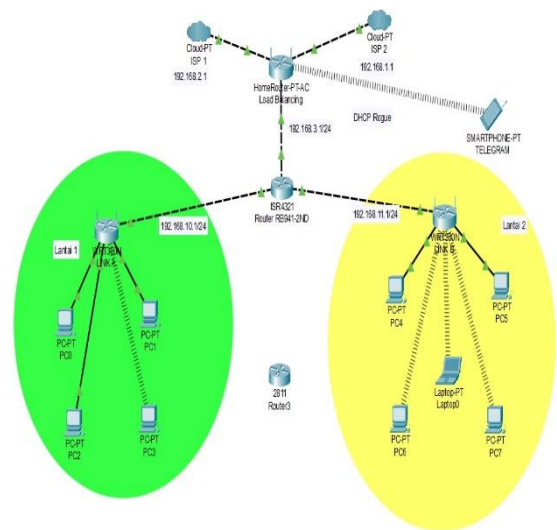
Tinjauan Pustaka dilakukan untuk mengkaji dari penelitian terdahulu mengenai *Load Balancing*, *PCC*, *Failover*, *Bot Telegram*, dan berbagai teori yang berhubungan mengenai permasalahan yang ingin dipecahkan. Tinjauan Pustaka ini juga dapat membantu peneliti dalam mengetahui kelebihan dan kekurangan dari metode yang dipakai dari hasil penelitian terdahulu.

### 2.3. Perancangan

#### a. Perancangan Design Topologi Jaringan

Bentuk desain dalam melakukan perancangan topologi jaringan yang akan diimplementasikan dalam penelitian serta proses penentuan jalur traffic antara dua buah *ISP* yang tersedia.

Pada Gambar 2 dari topologi jaringan dibawah ini dapat dilihat bahwa 2 link *ISP* yang ada dihubungkan menggunakan 1 router yang sudah dikonfigurasi *Load Balancing*, *PCC*, *Failover*, dan *Bot Telegram* kemudian jalur-jalur internet dibagikan lagi melalui *switch* yang ada ke *access point* pada Lantai 1 dan Lantai 2.



Gambar 2. Topologi Jaringan

#### b. Perancangan Fisik

Tabel 2. Table Perangkat Keras (Hardware)

No	Perangkat	Deskripsi	Interface	Address	Network	
1.	Router Mikrotik	RB941-2nD hAP - Lite	ISP	192.168.1.2	192.168.1.1	
			1(ethernet1)	ISP	192.168.2.2	192.168.2.1
			2(ethernet2)	Local ( test admin)	192.168.3.1/24	192.168.3.0
2.	Laptop	Asus, Intel Core i3, RAM 8GB	Ethernet 3	192.168.3.2	192.168.3.1	
3.	ISP	Telkom dan Biznet				
4.	Smartphone	Telegram Bot				

Pada tahap ini dilakukan perancangan yang berhubungan dengan peralatan yang akan digunakan dan penerapan topologi jaringan yang sudah di design sebelumnya. Hal ini dilakukan agar dalam tahapan implementasian nanti akan memudahkan dalam memahami dan apabila ada troubleshooting kedepannya mudah dipahami. Adapun perangkat keras dan perangkat lunak yang digunakan dapat dilihat pada tabel 2.

### 2.4. Implementasi

Pada tahap ini, instalasi dan konfigurasi sesuai parameter desain yang sudah dibuat. implementasi mengacu pada bagaimana desain jaringan telah diterapkan dilapangan, termasuk penyiapan dan konfigurasi. Peneliti mengagregasi bandwidth yang tersedia dari dua atau lebih ISP menggunakan pendekatan load balancing untuk meningkatkan bandwidth yang tersedia dan memastikan keandalan koneksi internet.

Prosedur pemulihan jaringan juga diperlukan untuk menjamin tingkat keandalan *Load Balancing* yang terintegrasi ke dalam jaringan. Oleh karena itu, diperlukan mekanisme *Failover* yang berfungsi sebagai koneksi cadangan jika salah satu saluran ISP mengalami masalah atau tidak tersedia. Jika terjadi pemadaman atau gangguan jaringan, tautan cadangan akan segera menggantikan tautan utama yang tidak berfungsi untuk menjaga konektivitas jaringan. Serta ditambahkan dengan *Bot Telegram* sebagai *DHCP Alert* yang dapat mendeteksi apabila ada *DHCP Rouge* yang dapat mengganggu jaringan karena memberikan informasi yang salah terhadap client.

### 2.5. Pengujian

Pada tahap ini dilakukan pengujian dengan menggunakan bantuan beberapa perangkat tertentu untuk menguji keberhasilan dari metode yang telah diterapkan, seperti pengetesan di *tools speedtest.net* untuk menguji keberhasilan penggabungan dua Koneksi internet menjadi satu dan pengujian semua metode yang diterapkan.

### 2.6. Hasil dan Pembahasan

Pembahasan pada tahapan ini menjelaskan dari proses penerapan *Load Balancing* yang telah dilakukan dan *Bot Telegram*.

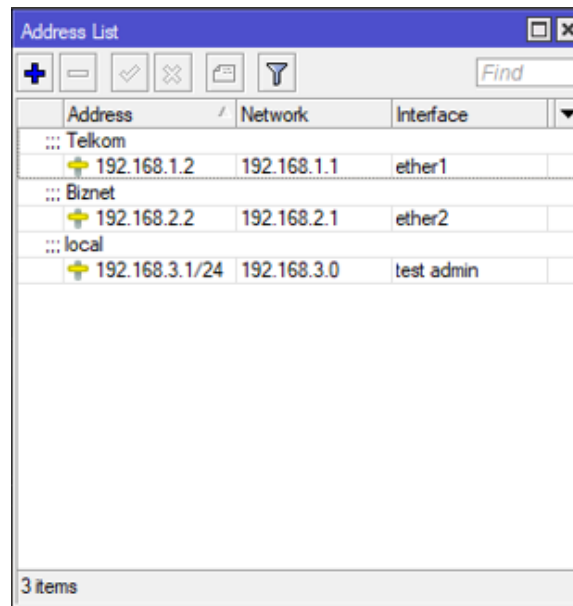
### 2.7. Kesimpulan dan Saran

Membuat kesimpulan dari hasil penelitian dan memberikan saran untuk penelitian selanjutnya.

## 3. HASIL DAN PEMBAHASAN

### 3.1. IP Address

Address List ini berguna untuk membuat daftar alamat IP Address yang dapat digunakan dalam berbagai konfigurasi nanti. Seperti gambar berikut:

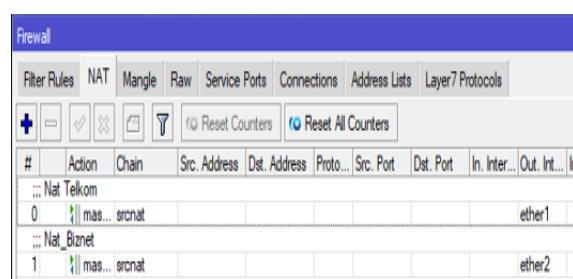


Gambar 3. Setting IP Address

Pada gambar 3 diatas menjelaskan pengalaman ip yang akan digunakan untuk ISP 1 dengan alamat IP 192.168.1.2 , ISP 2 dengan alamat IP 192.168.2.2, dan local sebagai ip admin dan ip untuk jalur output yang didapatkan oleh client dengan interface test admin ( bridge ) yang akan mendapatkan alamat IP 192.168.3.1/24.

### 3.2. Setting rule NAT-Masquerade

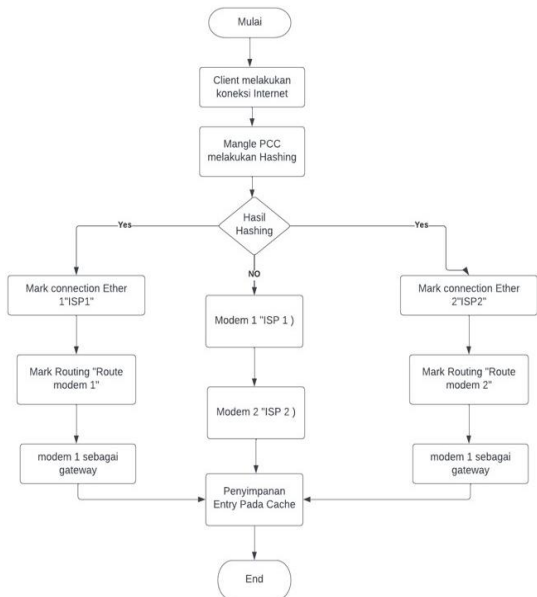
*NAT-Masquerade* Berfungsi sebagai jembatan mengubah alamat sumber paket data yang berasal dari "srcnat" atau dari alamat client agar komputer client bisa berkomunikasi dengan jaringan public, dan untuk Out. Interfacenya pilih ether1 sebagai ISP 1 dan ether2 sebagai ISP 2.



Gambar 4. Setting rule NAT-Masquarade

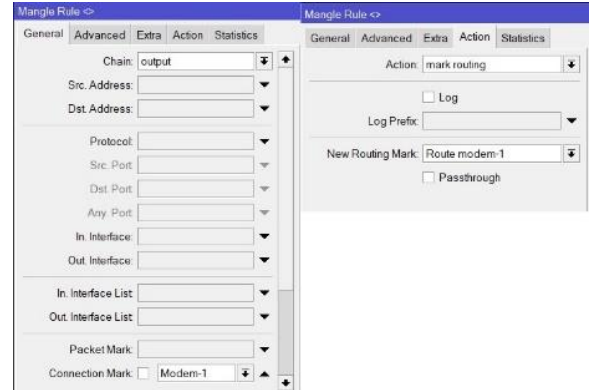
### 3.3. Setting Loadbalancing PCC

Pada perancangan sistem sebelumnya, dilakukan konfigurasi dasar telah berhasil. Konfigurasi *Load Balancing* terdiri dari beberapa tahapan dan tahap tersebut harus berhasil agar teknik *Load Balancing* dapat berjalan dengan baik, dengan bagan alur seperti gamar 5 diatas . Berikut merupakan hasil dari tahapan konfigurasi *Load Balancing PCC*:



Gambar 5. Flowchart PCC

- c. Flag 3 dan 4 : *Action – mark routing* berfungsi untuk menandai paket data dan akan digunakan untuk menentukan *routing* dari paket tersebut. *Chain output* digunakan untuk menandai trafik yang keluar melalui *router mikrotik*.



Gambar 9. Setting PCC Flag 3&4

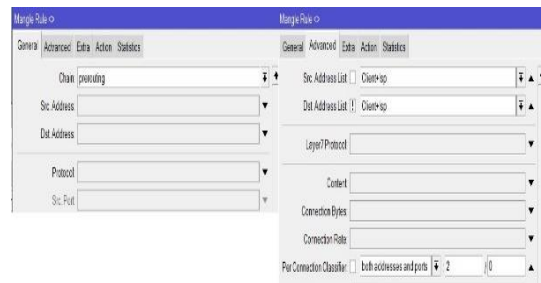
#	Action	Chain	Src. Address	Proto.	Src. Port	Dst. Port	In. Inter.	Out. Int.	In. Inter.	Out. Int.	Src. Ad.	Dst. Ad.
0	accept	prerouting									Client...	Client...
1	mark connection	input					ether1				Client...	Client...
2	mark connection	input					ether2				Client...	Client...
3	mark routing	output									Client...	Client...
4	mark routing	output									Client...	Client...
5	mark connection	prerouting									Client...	Client...
6	mark connection	prerouting									Client...	Client...
7	mark routing	prerouting									Client...	Client...
8	mark routing	prerouting									Client...	Client...

Gambar 6. Setting PCC

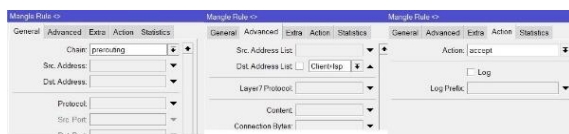
- d. Flag 5 dan 6: *Action – mark connection* berfungsi untuk membuat mark baru pada sebuah *connection traffic*. *Chain prerouting* digunakan untuk menandai trafik yang masuk menuju dan melalui router (trafik *download*). Pada tahapan dibuat sebuah rule untuk memecah traffic data yang melalui *router* kedalam 2 stream yang berbeda dan akan dikirimkan lewat *ether1* dan *ether2*.

Pada Gambar 6 menjelaskan yaitu:

- a. Flag 0 : *Action – accept* berfungsi paket data yang datang ke chain diterima dan tidak dicek lagi di rule bawahnya serta langsung keluar dari *chain*. *Chain – prerouting* berfungsi Untuk trafik yang menuju router (local proces) dan melalui router. > Tidak bisa memilih *out-interface*.

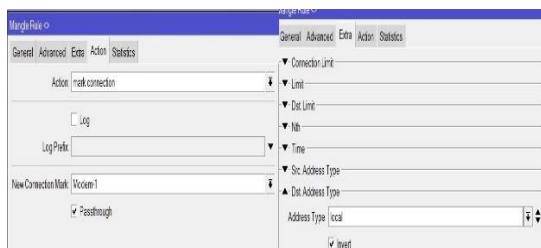


Gambar 10. Setting PCC Flag 5&6

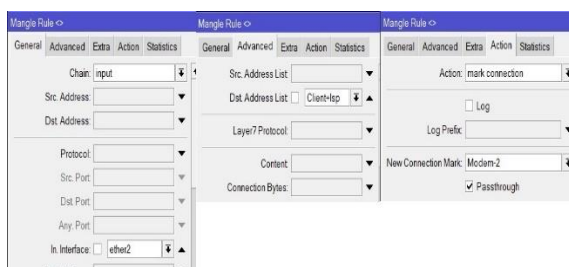


Gambar 7. Setting PCC Flag 0

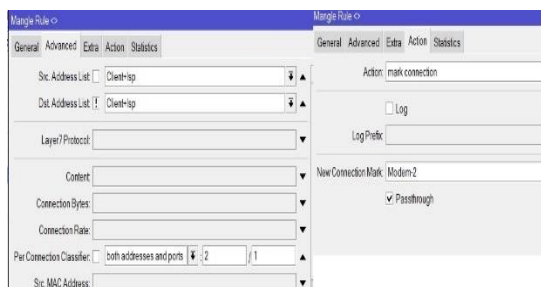
- b. Flag 1 dan 2 : *Action – mark connection* berfungsi untuk membuat *mark* baru pada sebuah *connection traffic* dan *Chain input* Digunakan untuk menandai trafik yang masuk menuju ke *router mikrotik* yaitu *ether 1* dan *ether 2* yang memuat *isp 1* dan *isp 2*.



Gambar 11. Setting PCC Flag 5&6



Gambar 8. Setting PCC Flag 1&2

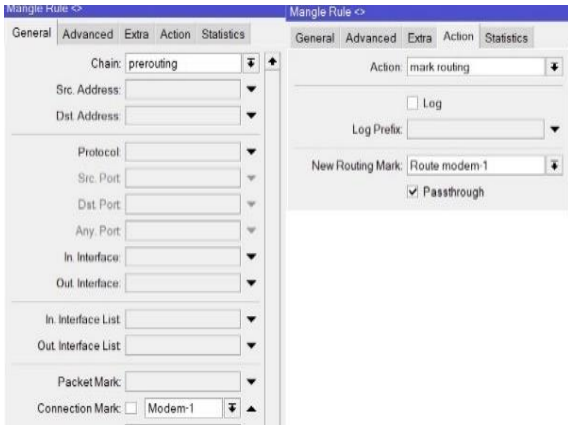


Gambar 12. Setting PCC Flag 5&6



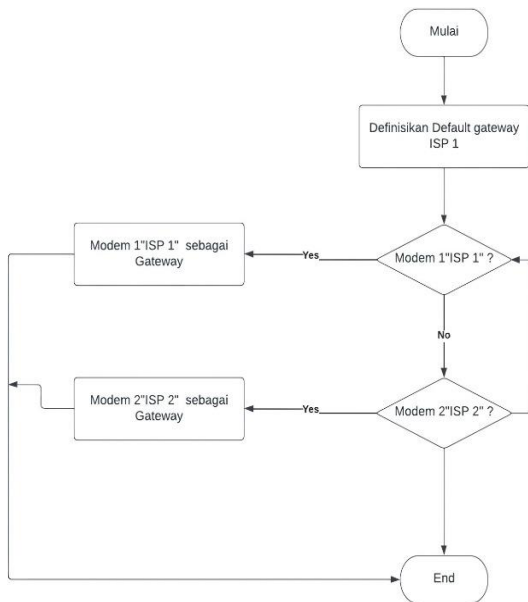
Pada *ISP2* lakukan konfigurasi serupa, perbedaannya hanya pada bagian parameter reminder di *Advanced* isikan 1 dan *New Connection Mark* isikan Modem-2.

- e. Flag 7 dan 8 : *Action – mark routing* berfungsi untuk menandai paket data dan akan digunakan untuk menentukan *routing* dari paket tersebut. *Chain prerouting* digunakan untuk menandai trafik yang masuk menuju dan melalui router (trafik *download*).



Gambar 13. Setting PCC Flag 7&8

### 3.4. Setting Failover



Gambar 14. Flowchart Failover

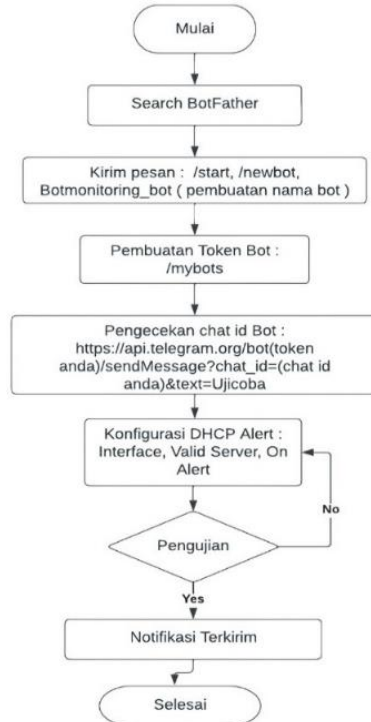
Pada gambar 14 diatas menjelaskan diagram alur yang berguna untuk mengetahui proses alur tahapan ketika penerapan *Failover*. Dengan *rule* seperti tabel dibawah:

Table 6. Rule Failover

Modem 1 "ISP 1"	Modem 2 "ISP 2"	Rule
Hidup	Hidup	Menggunakan Gateway semua ISP
Mati	Hidup	Menggunakan Gateway ISP 2
Hidup	Mati	Menggunakan Gateway ISP 1
Mati	Mati	-

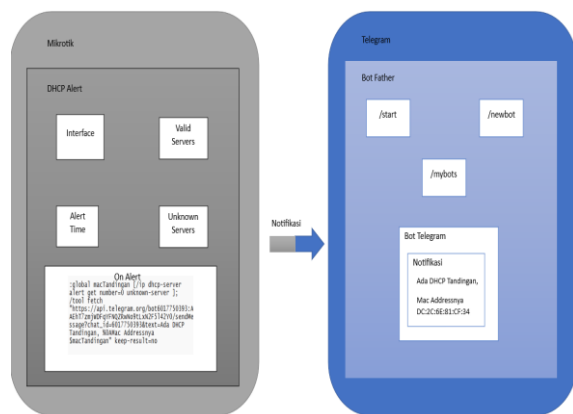
Rule tabel 6 diatas dapat menunjukkan konfigurasi failover yang berfungsi untuk melakukan backup(pencadangan data) link ketika terjadi salah satu pemutusan *gateway* maka dengan teknik tersebut akan mencari jalur *gateway* secara otomatis.

### 3.5. Konfigurasi Bot Telegram



Gambar 15. Flowchart Bot Telegram

Pada Gambar diatas menjelaskan diagram alur yang berguna untuk mengetahui proses dari tahapan *Bot Telegram* yang akan diterapkan.

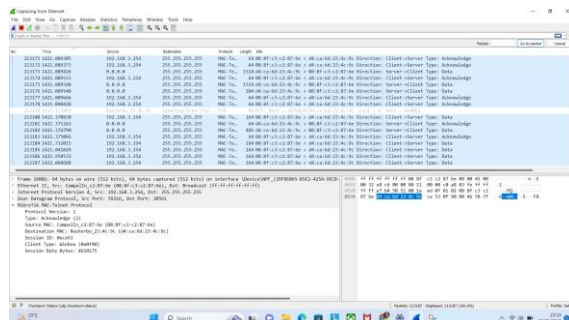


Gambar 17. Mock-up

Pada Gambar 18 diatas adalah *Mock-up* dari bagaimana *Bot Telegram* nantinya akan bekerja, dengan penjelasan seperti dibawah:

- Dengan interface diisi *test admin* karena *test admin* berada dalam 1 *segmen network* yang sama dan sebagai akses *admin*.

- *Valid Servers* diisi dengan *Mac Address* dari *test admin*.
- *Alert Timeout* atau waktu peringatan akan nada setiap satu jam bila ada *DHCP* tandingan yang terdeteksi.
- *Unknown servers* akan terisi otomatis oleh *Mac Address* dari *DHCP* tandingan bila terdeteksi adanya *DHCP Server* lain.
- *On Alert* diisi dengan script. Baris pertama berguna untuk mendapatkan *Mac Address* dari *DHCP server* tandingan, baris kedua berguna untuk mengirimkan notifikasi apabila ada *DHCP server* tandingan, dan baris ketiga untuk menampilkan *Mac Address* dari *DHCP server* tandingan ke telegram.



Gambar 20. Hasil PCC

Measurement	Captured	Displayed
Packets	213187	213187 (100.0%)
Time span, s	1422.061	1422.061
Average pps	149.9	149.9
Average packet size, B	828	828
Bytes	176479178	176479178 (100.0%)
Average bytes/s	124 k	124 k
Average bits/s	992 k	992 k

Gambar 21. Hasil PCC

### 3.6. Hasil Load Balancing

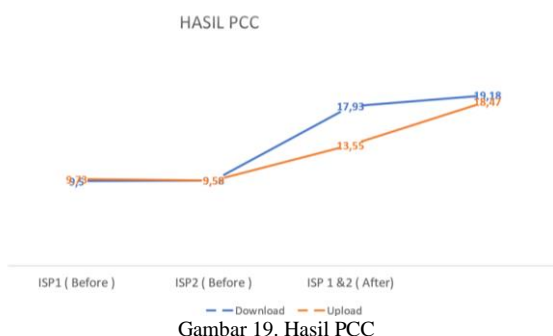
Pada tahap pengujian *Load Balancing* dengan metode *PCC* ini peneliti mendapatkan hasil dari pembagian beban trafik pada kedua jalur koneksi yang dapat dilihat bagian *Interface List* pada gambar 18 dibawah.

I	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	FP Tx	FP Rx	FP Tx Packet (p/s)	FP Rx
30	97.1 kbps	109.5 kbps	38	39	97.1 kbps	109.5 kbps	38	
38	208.3 kbps	22.3 kbps	22	28	126.4 kbps	91.6 kbps	39	
38	179.3 kbps	231.0 kbps	58	42	263.3 kbps	228.5 kbps	52	
38	0 bps	0 bps	0	0	0 bps	0 bps	0	
38	0 bps	0 bps	0	0	0 bps	0 bps	0	
38	200.7 kbps	227.6 kbps	83	85	223.6 kbps	0		

Gambar 18. Hasil PCC

Dari pengujian pada *tools speedtest.net* bahwa speed dari kedua *ISP* yang berhasil digabungkan menjadi satu, berhasil mendapatkan speed 20 *Mbps* yang diperoleh dari masing – masing speed yang sama yaitu 10 *Mbps* dari *ISP 1* dan *ISP 2*. Dengan digambarkan seperti Gambar yaitu :

- Sebelumnya *ISP 1* download 9.50 *Mbps* dan upload 9.73 *Mbps*.
- Sebelumnya *ISP 2* download 9.58 *Mbps* dan upload 9.59 *Mbps*.
- Sesudahnya pengujian pertama didapatkan download 17.93 *Mbps* dan upload 13.55 *Mbps*.
- Sesudahnya pengujian kedua didapatkan download 19.18 *Mbps* dan upload 18.47 *Mbps*.



Gambar 19. Hasil PCC

Pada Gambar 4.3 4 dan 4.3 5 merupakan hasil pengujian dengan aplikasi Wireshark dan diperoleh hasil QOS yaitu:

- Delay = 6ms
- Jitter = 0,5165 ms
- Throughput = 0,1241 mb
- Packet loss = 0 %.

### 3.7. Hasil Failover

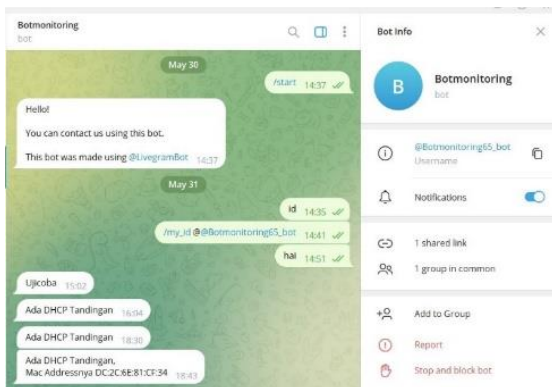
Untuk tahap pengujian failover, peneliti membuat simulasi seolah-olah internet di sisi provider pada *ISP 2* ( Modem 2 ) bermasalah, maka nanti rule routing akan secara otomatis dialihkan ke koneksi yang masih aktif adalah pada *ISP 1*( Modem 1 ) dan “Route to modem 1”.

Route	Next Hop	Rules	VRF
AS	0.0.0.0/0	192.168.1.1 reachable ether1	1
AS	0.0.0.0/0	192.168.2.1 unreachable	2
AS	0.0.0.0/0	192.168.1.1 reachable ether1	1 Route modem...
AS	0.0.0.0/0	192.168.2.1 reachable ether2	1 Route modem...
DAC	192.168.1.1	ether1 reachable	0 192.168.1.2
DAC	192.168.2.1	ether2 reachable	0 192.168.2.2
DAC	192.168.3.0/24	test admin reachable	0 192.168.3.1

Gambar 22. Hasil Failover

Pada Gambar 16 menjelaskan bahwa route yang aktif ada pada flag 4, 5, dan 6 dimana ada keterangan *DAC: Dynamic Active Connect* yang berarti *IP* yang aktif dan connect adalah *IP dynamic* yang sudah diset untuk *ISP-1*, *ISP-2*, dan *LAN*. Arti *distance* merupakan pembandingan atau perbedaan pada route tersebut, jika dilihat untuk *ISP-1*, *ISP-2* dan *test admin* tidak ada perbedaan untuk *distance*-nya.

### 3.8. Hasil Bot Telegram



Gambar 23. Hasil Bot Telegram

Gambar diatas menunjukkan bahwa *Bot Telegram* yang telah diterapkan berhasil mendeteksi adanya *DHCP* tandingan dengan *Mac Address* yang didapat DC:2C:6E:81:CF:34 dan berhasil dikirim melalui notifikasi *Telegram* diatas.

### 4. DISKUSI

Pada penelitian ini menggunakan dua link ISP dengan perbandingan bandwidth 1:1 yaitu 10 Mbps dan 10 Mbps. Dengan menggunakan metode PCC yang mampu menentukan gateway untuk setiap data paket yang masih berhubungan dengan data yang telah ada sebelumnya melewati salah satu gateway, hubungan antara client dan server lebih aman karena menggunakan jalur yang sama, tidak mengganggu pengguna akhir atau end-user, lebih stabil. Beberapa literatur memperjelas bahwa load balance tidak akan menambah besar bandwidth yang diperoleh, tetapi hanya bertugas untuk membagi trafik dari kedua bandwidth tersebut agar dapat terpakai secara seimbang[8]-[11]-[16]. Namun pada penelitian ini akan menjadi rumus matematika  $1+1=2$  yang biasanya dalam load balance  $1+1=1+1$ . Bisa dilihat dalam hasil dengan pengujian pada tools speedtest.net didapatkan bahwa Load Balancing yang diterapkan mampu menggabungkan 2 ISP menjadi satu yaitu Download menjadi 19.18 Mbps dan Upload 18.47 Mbps.

Serta penambahan Failover untuk membackup kelemahan dari metode PCC yaitu Failover mampu untuk dapat bergerak secara manual atau otomatis jika salah satu sistem gagal sehingga itu menjadi cadangan untuk sistem yang gagal [11]. Jika gerbang 1 adalah terputus, gateway cadangan akan menggantikan gateway 1. Jika gateway 1 kembali normal, jalur koneksi digunakan lagi menjadi gateway 1. Begitu juga dengan gateway 2 saat itu terputus. Jadi dengan ditambah metode failover dapat membantu menutupi kekurangan dari PCC.

Bot Telegram digunakan sebagai DHCP Alert yang dapat mendeteksi apabila ada DHCP Rouge yang dapat mengganggu jaringan karena memberikan informasi yang salah terhadap client[13].

### 5. KESIMPULAN

Kesimpulan yang diperoleh setelah melakukan tahapan-tahapan pada penelitian adalah sebagai berikut:

1. Dengan menggunakan metode PCC, mampu memaksimalkan penggunaan *bandwidth* serta meminimalisir terjadinya *downtime* dalam mengirim atau menerima data.
2. *Failover* mampu untuk dapat bergerak secara manual atau otomatis jika salah satu sistem gagal sehingga itu menjadi cadangan untuk sistem yang gagal. Jika gerbang 1 adalah terputus, *gateway* cadangan akan menggantikan *gateway* 1. Jika *gateway* 1 kembali normal, jalur koneksi digunakan lagi menjadi *gateway* 1. Begitu juga dengan *gateway* 2 saat itu terputus. Jadi dengan ditambah metode *Failover* dapat membantu menutupi kekurangan dari PCC.
3. Dari pengujian pada tools speedtest.net yaitu :
  - Sebelumnya ISP 1 *download* 9.50 Mbps dan *upload* 9.73 Mbps.
  - Sebelumnya ISP 2 *download* 9.58 Mbps dan *upload* 9.59 Mbps.
  - Sesudahnya pengujian pertama didapatkan *download* 17.93 Mbps dan *upload* 13.55 Mbps.
  - Sesudahnya *download* 19.18 Mbps dan *upload* 18.47 Mbps.
4. Dari Pengujian pada tools Wireshark yaitu :
  - *Delay* = 6ms
  - *Jitter* = 0,5165 ms
  - *Throughput* = 0,1241 mb
  - *Packet loss* = 0 %
5. Dengan *Bot Telegram* mampu mengirimkan notifikasi apabila ada *DHCP Server* tandingan dengan isi pesan berhasil mendapatkan *Mac Address* atau *unknown server* dari *DHCP server* tandingan yaitu DC:2C:6E:81:CF:34.

### UCAPAN TERIMA KASIH

Penulis memanjatkan doa atas ke hadirat Tuhan Yang Maha Esa dengan segala rahmat dan karunianya serta, sehingga penulis dapat menyelesaikan skripsi dengan judul "Implementasi Load Balancing Dengan Per Connection Classifier Dan Failover Serta Pemanfaatan Bot Telegram (Studi Kasus : PT TUJUH MEDIA ANGKASA)". Penulis menyadari bahwa beberapa pihak telah membantu dalam pembuatan skripsi ini. Hadiah terbesar penulis kepada orang tua dan teman-teman yang telah banyak membantu. Secara khusus penulis ingin mengucapkan terima kasih kepada dosen pembimbing yang telah memahami, meluangkan waktu, dan tenaganya, serta memberikan perhatian untuk membantu selama penyusunan skripsi ini. Terimakasih juga kepada seluruh staff Tujuh Media Angkasa yang telah memberikan data dan pembelajaran untuk digunakan dalam penelitian ini, dan civitas akademika



FASILKOM Program Studi Teknik Informatika Program Studi Teknik Informatika Universitas Mercu Buana yang telah mendukung penelitian ini.

#### DAFTAR PUSTAKA

- [1] A. Mustofa and D. Ramayanti, "Implementasi Load Balancing Dan Failover To Device Mikrotik Router Menggunakan Metode Nth (Studi Kasus: Pt. Go-Jek Indonesia)", Vol. 7, No. 1, pp. 139–144, 2020. <https://doi.org/10.25126/jtiik.202071638>
- [2] L. H. Efendi, D. W. Chandra, (n.d.). "Implementasi Weighted Load Balancing Per Connection Classifier Dengan Teknik Failover Menggunakan Mikrotik RB941-2ND (Studi Kasus : Dinas Pemberdayaan Perempuan, Perlindungan Anak dan Keluarga Berencana Kabupaten Grobogan)".
- [3] E. Permatasari and R. Rushendra, "Analisa Pembagian Penggunaan Bandwidht Untuk Jaringan Internet Pelanggan Mnc Play Area Jakarta Dengan Menggunakan Teknologi Link Aggregation Dan Load Balance," *Al Qalam: Jurnal Ilmiah Keagamaan Dan Kemasyarakatan*, Vol. 15, No. 2, 2021. <https://doi.org/10.35931/aq.v15i2.769>
- [4] M. Dartono, and D. Irawan, (n.d.). "Penerapan Metode Per Connection Classifier (PCC) Pada Perancangan Load Balancing Dengan Router".
- [5] F. Fauzi and R. Darmawan, "Analisis dan Perancangan Load Balancing dan Failover menggunakan link kartu GSM," *Jurnal Ilmiah Informatika Dan ....*, 2021 <https://journal.teknikunkris.ac.id/index.php/jiifor/article/view/247>
- [6] A. M. Elhanafi, I. Lubis, D. Irwan, and A. Muhazir, "Simulasi Implementasi Load Balancing PCC Menggunakan Simulator Gns3. Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP)", Vol. 1, No. 2, 2018. <https://doi.org/10.34012/jutikomp.v1i2.236>
- [7] T. Rahman, E. Sulistianto, A. Sudibyo, S. Sumarna, and B. Wijonarko, "Per Connection Classifier Load Balancing dan Failover MikroTik pada Dua Line Internet", *JIKA (Jurnal Informatika)*, Vol. 5, No. 2, 2021. <https://doi.org/10.31000/jika.v5i2.4517>
- [8] F. Ardianto, B. Alfaresi, and A. Darmadi, "Rancang Bangun Load Balancing Dua Internet Service Provider (ISP) Berbasis Mikrotik. Jurnal Surya Energy", Vol. 3, No. 1, 2018. <https://doi.org/10.32502/jse.v3i1.1232>
- [9] R. R. Abdullah, and A. Nurhayati, "Monitoring Sistem Keamanan Jaringan Berbasis Telegram Bot Pada Local Area Network", *Journal of Informatics and Communications Technology*, Vol. 1, No. 2, pp. 45–053, 2019. [www.sourcefire.com](http://www.sourcefire.com)
- [10] M. I. Oktavianto and Y. R. Prayogi, "Sistem Monitoring Jaringan Load balancing Dengan Metode Equal Cost Multipath (ECMP) Menggunakan Media Telegram", *Journal of Computer Science and ....*, 2019 <https://journal.unusida.ac.id/index.php/jik/article/view/60>
- [11] E. Safrianti, "Peer Connection Classifier Method for Load Balancing Technique", *International Journal of Electrical, Energy and Power System Engineering*, Vol. 4, No. 1, 2021. <https://doi.org/10.31258/jjeepse.4.1.127-133>
- [12] R. Fahrizal, M. I. Santoso and M. Z. Arifin, "Implementation of Multipath Routing With Equal Cost Multipath (ECMP) and Per Connection Classifier (PCC)," *2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE)*, pp. 169–173, 2020. <https://doi.org/10.1109/ICIEE49813.2020.9277496>
- [13] R. Kurniawan and A. D. Putri, "Perancangan Jaringan menggunakan Metode Load Balancing PCC, Failover, dan Notifikasi", *SENTINEL*, Vol. 2, No. 2, 2019. <https://doi.org/10.56622/sentineljournal.v2i2.15>
- [14] D. K. Hakim, S. A. Nugroho, and J. R. D. Waluh, "Implementasi Telegram Bot untuk Monitoring Mikrotik Router The Implementation of Bot Telegram for Monitoring Microtic Router," *SAINTEKS*, Vol. 16, 2019.
- [15] Y. Pangestu, D. Setiyadi, and F. N. Khasanah, "Metode Per Connection Classifier Untuk Implementasi Load Balancing Jaringan Internet," *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, Vol. 6, No. 1, 2018. <https://doi.org/10.33558/piksel.v6i1.1389>
- [16] T. Sukendar, "Keseimbangan Bandwidth Dengan Menggunakan Dua ISP Melalui Metode Nth Load Balancing Berbasis Mikrotik," *Jurnal Teknik Komputer Amik Bsi*, Vol. 3, No. 1, 2017.
- [17] F. W. Christanto, S. Susanto, and A. Priyanto, "Load Balancing-Failover Methods using Static Route with Address List, ECMP, PCC, and Nth for Optimizing LAN Network: A Comparison," *International Journal of Communication Networks and Information Security*, Vol. 11, No. 3, 2019. <https://doi.org/10.17762/ijcnis.v11i3.4340>
- [18] S. A. Haris and H. Suhartono, "Menjaga

Kestabilan Jaringan Load Balancing Nth Dengan Teknik Failover Pada PT. Jakarta Samudera Sentosa Jakarta", Vol. 6, No.1, 2018.

- [19] F. Ardianto and T. Akbar, "Perancangan Sistem Monitoring Keamanan Jaringan Jarak Jauh Menggunakan Mikrotik Operational System Melalui Virtual Private Network" *Jurnal Surya Energy*, Vol. 2, No. 1, 2017.
- [20] Y. Pangestu, D. Setiyadi, and F.N. Khasanah, "Metode Per Connection Classifier Untuk Implementasi Load Balancing Jaringan Internet", *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, Vol. 6, No. 1, 2018. <https://doi.org/10.33558/piksel.v6i1.1389>
- [21] F. A. Putra and A. Subardono, "Analisis Kinerja Per Connection Classifier dan Failover pada Multiple Gateway Internet Networks", *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, Vol. 10, No. 4, 2021. <https://doi.org/10.22146/jnteti.v10i4.2065>
- [22] Suryanto, T. Prasetyo, N. Hikmah, "Implementasi Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Dengan Failover Berbasis Mikrotik Router (Studi Kasus PT. Sumber Rejeki Power)", *Seminar Nasional Inovasi Dan Tren (SNIT)*, Vol. 1, No. 1, 2018.
- [23] Suryanto, T. Prasetyo, N. Hikmah, "Implementasi Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Dengan Failover Berbasis Mikrotik Router (Studi Kasus PT. Sumber Rejeki Power)", *Seminar Nasional Inovasi Dan Tren (SNIT)*, Vol. 1, No. 1, 2018.
- [24] Rushendra, K. Ramli, N. Hayati, E.Ihsanto, T. S. Gunawan and A. H. Halbouni, "Development of Intrusion Detection System usisng Residual Feedforward Neural Network Algorithm," *2021 4th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI*, 2021. <https://doi.org/10.1109/ISRITI54043.2021.9702773>.