

# Simulasi Implementasi *Load Balancing* pada Jaringan Internet dengan Metode *Per Connection Classifier* (PCC) dan Server dengan Metode IP Hash Menggunakan GNS3

Matthew Tanujaya<sup>1</sup>, Christine Dewi<sup>2\*</sup>

<sup>1</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

<sup>2\*</sup> Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

## article info

### Article history:

Received 6 September 2023

Received in revised form

24 November 2023

Accepted 10 December 2023

Available online January 2024

### DOI:

<https://doi.org/10.35870/jtik.v8i1.1550>

### Keywords:

Load Balancing; PCC; IP

Hash; GNS3.

### Kata Kunci:

Load Balancing; PCC; IP

Hash; GNS3.

## abstract

The internet connection should typically function without any issues at all times. However, there are instances where the link does not function properly because to impediments or other issues.. The GNS3 Network Simulator tool is used in this study to carry out the simulation. The goal of this strategy is to determine whether it can sustain the internet network's quality and traffic levels while distributing the load throughout the network channel. Additionally, correct server path management will be accomplished using the Ip Hash technique. By using the PCC load balancing method, the traffic load is obtained more smoothly without data accumulation. Because the other network functions as an internet provider, internet access will still be accessible even if one of these networks dies. While using Ip hash, the results of the test obtained an average throughput with the small data request category of 16.69 KB/s. with the medium data request category of 54.31KB/s, and 1200 KB/s for the large data request category.

## abstract

The internet connection should typically function without any issues at all times. However, there are instances where the link does not function properly because to impediments or other issues.. The GNS3 Network Simulator tool is used in this study to carry out the simulation. The goal of this strategy is to determine whether it can sustain the internet network's quality and traffic levels while distributing the load throughout the network channel. Correct server path management will also be accomplished using the IP Hash technique. Using the PCC load balancing method, the traffic load is obtained more smoothly without data accumulation. Because the other network functions as an internet provider, internet access will still be accessible even if one of these networks dies. While using IP hash, the test results obtained an average throughput with the small data request category of 16.69 KB/s. with the medium data request category of 54.31KB/s and 1200 KB/s for the large data request category.

\*Corresponding Author. Email: [christine.dewi@uksw.edu](mailto:christine.dewi@uksw.edu) <sup>2\*</sup>.

© E-ISSN: 2580-1643.

Copyright © 2024 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISE'I). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.



Association for Computing Machinery

ACM Computing Classification System (CCS)

EBSCOhost

Communication and Mass Media Complete (CMMC)

## 1. Latar Belakang

Perkembangan pemakaian internet yang meningkat pesat saat ini menyebabkan permintaan akan layanan internet yang harus ditingkatkan. Sehingga diperlukan pengaturan beban pada trafik jaringan agar beban tersebut menjadi rata dan seimbang. Agar kualitas koneksi internet tetap terjaga dapat dilakukan dengan membagi beban ke dalam suatu jalur yang lain atau *link* dengan menggunakan teknik *load balancing*. Teknik *load balancing* ialah sebuah teknik jaringan komputer yang menyalurkan tiap beban pekerjaan secara seimbang pada beberapa jalur jaringan sehingga pekerjaan dapat berjalan dengan baik dan tidak terjadi *overload* (kelebihan beban) pada salah satu jalur jaringan. Ada tiga algoritma yang dapat digunakan pada Teknik *load balancing* yaitu Round Robin, Least Connection, Least Response Time, IP Hash dan lain-lain [1]. Teknik ini menggunakan konsep dengan menyeimbangkan beban dari Server atau Router yang digunakan untuk mendistribusikan jaringan internet pada pengguna dengan menggabungkan beberapa Line Internet Service Provider (ISP). Namun dengan adanya beberapa ISP, masing-masing ISP dapat memberikan *bandwidth* yang berbeda-beda. Maka dari itu untuk memberikan layanan yang sama kepada semua pengguna, semua *bandwidth* harus digabungkan menjadi satu. Maka dari itu teknik *load balancing* sangat berguna untuk menggabungkan beberapa *bandwidth* dari ISP yang berbeda-beda [2].

Metode *Per Connection Classifier* (PCC) ialah salah satu algoritma atau metode yang terdapat pada *load balancing*. PCC juga dapat digunakan untuk mengelompokkan trafik koneksi keluar masuk dan melalui router menjadi beberapa kelompok. *Src-address*, *ds-address*, *src-port*, dan *dst-port* menentukan jenis pengelompokan ini. Kelebihan metode PCC yaitu memiliki kemampuan untuk menentukan *gateway* untuk setiap paket data yang masih berhubungan dengan data yang telah dilewatkan pada salah satu *gateway* sebelumnya [2]. Metode *Ip Hash* merupakan salah satu metode dalam teknik *Load balancing*. Metode ini menyalurkan permintaan lalu lintas yang masuk sesuai dengan data yang terkait dengan IP pengguna. Data seperti tujuan IP, URL, domain, dan nomor port akan menentukan server mana yang akan dituju [1]. Agar dapat melakukan perutean lalu lintas ke *server backend* yang sama, kebijakan dari *Ip Hash* menggunakan alamat IP sumber *request* yang masuk

sebagai kunci *hashing* [6]. Sumber algoritma *Ip hash* memiliki sifat deterministik, yang berarti setiap langkah hanya bisa diproses sebanyak satu kali. Apabila jalur tidak ditemukan, maka algoritma dianggap selesai. Algoritma optimasi ini akan menghasilkan solusi yang sama bagi setiap input. *Hash* digunakan pada alamat IP yang melakukan *request*, dan *hash* harus memperhitungkan berapa banyak *server* dan bobot yang dituju pada Alamat IP tersebut [1]. Hasil daripada analisa dari penelitian sebelumnya menjadi hal yang sangat dibutuhkan pada penelitian ini. Penelitian sebelumnya dapat dijadikan sebagai pertimbangan serta panduan untuk melakukan implementasi *Load balancing* pada Jaringan Internet dengan metode *Per Connection Classifier* (PCC) dan *Server* dengan metode *Ip Hash* menggunakan GNS3.

Sumiati, Trisnawan, & Fauzi (2020) mengatakan dalam penelitiannya pengujian *throughput* dan *packet loss* menunjukkan bahwa algoritma *source IP Hash* dapat menangani permintaan klien lebih baik daripada *least traffic*. Algoritma ini juga dapat menyalurkan alamat IP dari sumber yang sama ke *server* yang sama [1]. Nuraini (2022) mengatakan dalam penelitiannya bahwa Metode *Load balancing* membagi beban yang masuk ke berbagai server yang menyediakan layanan. Hasil pengujian dengan 10.000 koneksi menunjukkan *throughput* rata-rata 11473,72 bps, sedangkan *throughput* dengan rata-rata 7236,6 bps didapatkan untuk sistem tanpa *load balancing*. Dari hasil pengujian ini, dapat disimpulkan bahwa *throughput* rata-rata sistem *load balancing* lebih besar daripada sistem tanpa *load balancing*. Sehingga penerapan *Load balancing* memungkinkan peningkatan kinerja layanan sistem yang berkelanjutan [3].

Laksamana (2021) mengatakan dalam penelitiannya, *Load balancing* pada jaringan memungkinkan pembagian beban trafik ke dua atau lebih jaringan secara seimbang agar pekerjaan dapat berjalan dengan lebih baik dan tidak membebani salah satu jalur koneksi internet. Pada jaringan ini, teknik *failover* digunakan agar apabila salah satu *gateway* terputus, maka secara otomatis *gateway* yang lain akan mengambil alih semua trafik jaringan [4]. Dalam penelitian ini, penulis melakukan sebuah simulasi terhadap implementasi Teknik *Load balancing* dengan menggunakan metode *Per Connection Classifier* (PCC) dan *Ip Hash*. Penelitian ini akan dikerjakan dengan menggunakan *Software Network Simulator* GNS3. Pada

GNS3 tersebut akan dirancang topologi yang akan menyerupai topologi yang ada pada suatu instansi atau perusahaan dan akan dilakukan konfigurasi dari tiap rancangan tersebut seperti konfigurasi pada *Router*, *Server*, komputer dsb. Setelah itu pada topologi tersebut akan diimplementasikan teknik *Load balancing* dengan menggunakan metode *Per Connection Classifier* (PCC) dan *Ip Hash* untuk melihat apakah teknik ini dapat menjaga kualitas dan trafik dari jaringan internet serta dapat membagi beban yang ada pada jalur jaringan dan kegunaan metode *Ip Hash* untuk mengelola jalur server dengan baik.

*Load balancing* merupakan teknik penting dalam mengoptimalkan kinerja dan keandalan server web. Berbagai metode telah dieksplorasi dan diimplementasikan untuk mencapai *load balancing* yang efisien. Misalnya, penelitian oleh Aldori berfokus pada implementasi *load balancing* menggunakan metode *Per Connection Classifier* (PCC) berbasis Mikrotik di sebuah lembaga pendidikan tertentu. Demikian pula, Amalia menganalisis kinerja server web menggunakan *Load balancing* as a Service dalam lingkungan virtual Openstack. Studi-studi ini menyoroti aplikasi praktis dan manfaat dari teknik *load balancing* dalam pengaturan dunia nyata [5][6]. Selain itu, Apriliansyah *dkk* mengimplementasikan *load balancing* pada server web menggunakan Nginx, memperlihatkan fleksibilitas dari berbagai metode *load balancing*. Bhayangkara dan Riadi juga berkontribusi dalam bidang ini dengan menerapkan Proxy Server dan *Load balancing* dengan metode PCC berbasis Mikrotik, menunjukkan relevansi *load balancing* dalam infrastruktur jaringan [7][8]. Lebih lanjut, penelitian oleh Budiyo *dkk* menekankan penggunaan *load balancing* pada server web lokal menggunakan Policy Based Routing, yang menambahkan beragam strategi *load balancing* yang tersedia [9]. Selain itu, penelitian oleh Nugroho *dkk* melakukan analisis kualitas layanan *load balancing* menggunakan metode PCC, ECMP, dan NTH, yang memperkaya pemahaman tentang berbagai teknik *load balancing* dan dampaknya terhadap kualitas layanan [10][11][12][13][14].

Penelitian dengan pengenalan objek menggunakan algoritma berbasis CNN, termasuk berbagai variasi seperti Densenet Yolo V2, Resnet 50 Yolo V2, dan Yolo V4 SPP, telah memunculkan temuan menarik. Dewi dan Christanto (2022) melakukan evaluasi

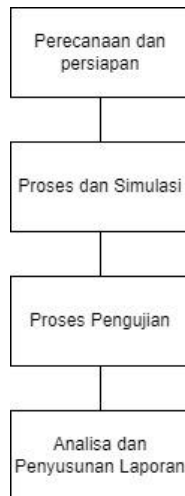
menyeluruh terhadap beberapa algoritma ini, menyoroti keunggulan dari model-model seperti Densenet Yolo V2 CSP SPP dan Yolo V4 CSP SPP, yang menunjukkan peningkatan signifikan dalam tingkat presisi hingga 8,88% [15]. Selanjutnya, peneliti lain, seperti Apriliansyah, Fitri, dan Iskandar (2022), menggarisbawahi pentingnya sistem penyeimbangan beban untuk menjaga stabilitas server web dalam menghadapi lonjakan trafik. Dalam penelitian mereka, metode algoritma Least Connection terbukti memberikan hasil yang optimal dalam hal stabilitas dan responsivitas server [16]. Selanjutnya, Bhayangkara dan Riadi (2014) membahas implementasi proxy server dan *load balancing* menggunakan Mikrotik dalam studi kasus Shmily.net, yang menghasilkan peningkatan keandalan koneksi internet secara *real-time* [17]. Analisis terhadap metode *load balancing* juga dilakukan oleh Amalia (2020) dalam lingkungan virtual OpenStack, dapat meningkatkan kinerja web server melalui teknik ini [18].

Sementara itu, penelitian Ohhyver dan Chandra (2023) mengenai simulasi keamanan jaringan dengan menggunakan OpenvSwitch dan NIDS menunjukkan tingkat keberhasilan yang signifikan dalam mendeteksi serangan DOS [19]. Implementasi konkret dari konsep penyeimbangan beban juga terjadi di lapangan, seperti yang dijelaskan oleh Saputra, Aspriyono, dan Al Akbar (2022), yang berhasil menerapkan metode PCC untuk mengelola beban trafik di Perumahan Bumi Mas [20]. Secara keseluruhan, penelitian-penelitian tersebut, mulai dari evaluasi algoritma hingga implementasi praktis, memberikan gambaran yang komprehensif tentang konsep dan aplikasi penyeimbangan beban dalam konteks yang beragam. Dengan demikian, literatur ini memberikan landasan yang kuat untuk memahami dan mengimplementasikan teknologi ini dalam berbagai skenario [21][22][23][24].

Dari berbagai penelitian dan implementasi yang telah disajikan, jelas bahwa teknik *load balancing* memiliki peran yang krusial dalam memastikan kinerja, stabilitas, dan keandalan jaringan serta server web. Berbagai algoritma seperti *Per Connection Classifier* (PCC), IP Hash, dan metode lainnya telah terbukti efektif dalam mendistribusikan beban trafik secara merata, meningkatkan *throughput*, dan mengelola koneksi internet dengan lebih baik. Dengan memahami konsep dan penerapan teknik *load balancing*

ini, diharapkan dapat terus meningkatkan efisiensi dan kualitas layanan dalam berbagai lingkungan jaringan.

## 2. Metode Penelitian

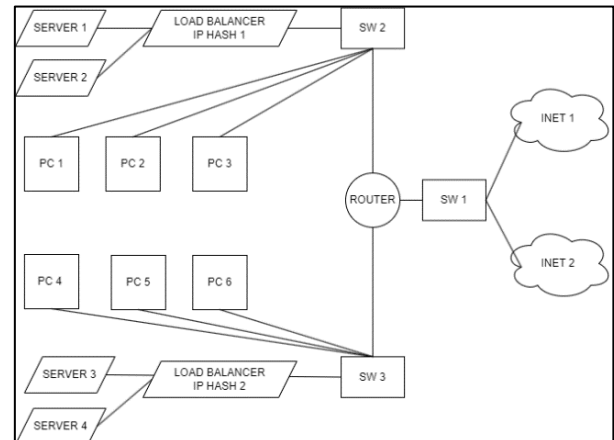


Gambar 1. Tahapan Penelitian

Berikut ialah penjelasan tahapan penelitian pada Gambar 1:

- 1) Persiapan dan perencanaan  
Pada tahapan ini, dilakukan persiapan *Software* untuk melakukan penelitian yaitu GNS3 dan membaca literatur-literatur yang ada. Setelah persiapan selesai, dilakukan perencanaan seperti pembuatan topologi dan sistematika dari rancangan pada jalur jaringan pada topologi.
- 2) Proses dan Simulasi  
Pada tahapan ini, dilakukan konfigurasi pada topologi yang ada di GNS3 dan melakukan percobaan trafik jaringan pada topologi tersebut. Setelah konfigurasi dan percobaan telah selesai, selanjutnya yaitu mengimplementasikan teknik *Load balancing* dengan menggunakan metode *Per Connection Classifier* (PCC) dan *Ip Hash*.
- 3) Testing  
Pada tahapan ini dilakukan proses uji coba pada teknik *Load balancing* yang menggunakan metode *Per Connection Classifier* (PCC) dan *Ip Hash* untuk melihat apakah teknik ini dapat menjaga kelancaran pada jaringan internet yang ada pada topologi yang telah dibuat.
- 4) Analisa dan penyimpulan hasil  
Pada tahap ini dilakukan analisa terhadap percobaan implementasi teknik *Load balancing*

dengan menggunakan metode *Per Connection Classifier* (PCC) dan *Ip Hash*, apakah berhasil atau tidak dalam menjaga kelancaran jaringan internet dan kelancaran lalu lintas *request* pada *server* lalu membuat kesimpulan atas penelitian yang telah dilakukan.



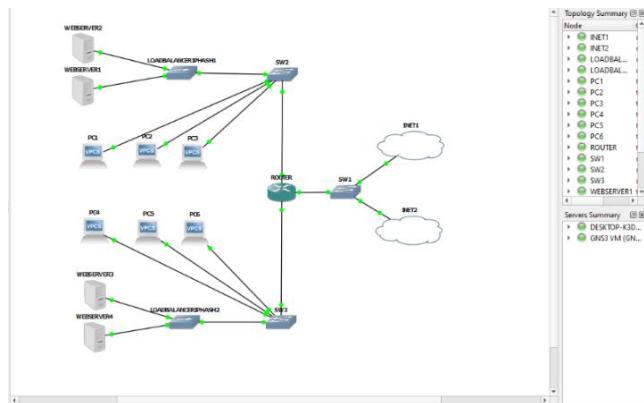
Gambar 2. Topologi *Load balancing* pada Penelitian

Topologi yang akan digunakan pada penelitian ini dapat dilihat pada Gambar 2. INET 1 dan INET 2 merupakan jaringan internet yang akan digunakan. SW 1 akan menerima jaringan internet dari kedua ISP tersebut dan pada SW 1 itulah akan diimplementasikan *Load balancing*. Setiap trafik jaringan akan diatur oleh SW 1 yang telah memiliki *Load balancing*. Lalu pada *router* akan menyalurkan jaringan pada SW 2 dan SW 3. SW 2 dan SW 3 akan menyalurkan jaringan internet pada PC, *server*, dan *load balancer* yang terhubung sesuai dengan SW masing-masing. *Load balancer* akan menjadi *server* yang menerapkan algoritma *Ip Hash* dan melakukan perhitungan pada *hash* tiap *client* yang ingin masuk kepada *server*. Pada Setiap PC dan *server* akan dilakukan konfigurasi dengan memberikan nama pada setiap PC dan *server* lalu memberikan alamat IP.

## 3. Hasil dan Pembahasan

Penulisan Pada penelitian ini akan dilakukan simulasi implementasi *Load balancing* pada jaringan internet dengan metode *Per Connection Classifier* (PCC) dan kepada *server* dengan metode *Ip Hash* menggunakan GNS3. Perancangan topologi *Load balancing*, pembagian daripada *bandwidth* pada jaringan LAN yang terhubung pada tiap *Switch*, PC, dan *server* menggunakan *Router* Mikrotik dilakukan dengan

skema dua ISP serta perancangan jalur dari PC client ke web server menggunakan *load balancer ip hash* yang telah dibuat pada GNS3. Rancangan skema tersebut dapat dilihat pada gambar 3.



Gambar 3. Skema Topologi Load balancing

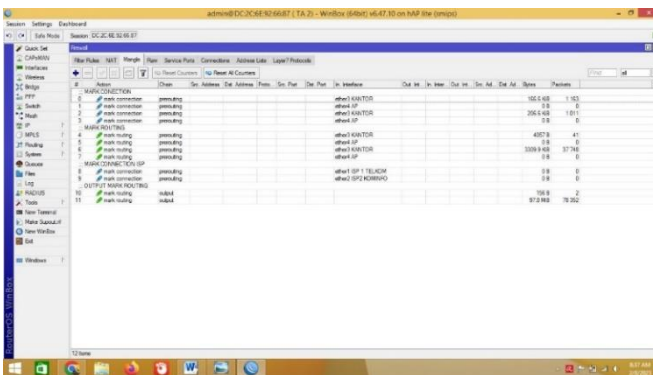
Pada setiap PC dan server juga dilakukan konfigurasi dengan memberikan *Ip Address* pada setiap PC. Adapun IP yang digunakan pada PC *client* dan *server* ialah sebagai berikut.

Tabel 1. Ip Address Perangkat

| Perangkat   | Ip Address  | Subnet Mask    | Default Gateway |
|-------------|-------------|----------------|-----------------|
| PC 1        | 117.168.1.2 | 255.255.25.5.0 | 117.168.1.1     |
|             | 117.168.1.3 | 255.255.25.5.0 | 117.168.1.1     |
| PC 2        | 117.168.1.4 | 255.255.25.5.0 | 117.168.1.1     |
|             | 117.168.5.2 | 255.255.25.5.0 | 117.168.5.1     |
| PC 3        | 117.168.5.3 | 255.255.25.5.0 | 117.168.5.1     |
|             | 117.168.5.4 | 255.255.25.5.0 | 117.168.5.1     |
| WEBSERV ER1 | 117.168.1.6 | 255.255.25.5.0 | 117.168.1.1     |
|             | 117.168.1.7 | 255.255.25.5.0 | 117.168.1.1     |
| WEBSERV ER2 | 117.168.5.6 | 255.255.25.5.0 | 117.168.5.1     |
|             | 117.168.5.7 | 255.255.25.5.0 | 117.168.5.1     |
| LOAD        |             |                |                 |
| BALANCE     | 117.168.1.5 | 255.255.25.5.0 | 117.168.1.1     |
| R IP HASH   |             |                |                 |
| 1           |             |                |                 |

|           |             |                |             |
|-----------|-------------|----------------|-------------|
| LOAD      |             |                |             |
| BALANCE   | 117.168.1.5 | 255.255.25.5.0 | 117.168.1.1 |
| R IP HASH | 5.5         | 5.0            | 5.1         |
| 2         |             |                |             |

Pada *router* juga dilakukan konfigurasi pada *Interface* agar dapat memudahkan dalam membedakan *port ether link* dengan *port ether* pada tiap LAN. Setelah melakukan konfigurasi pada *Router*, agar konfigurasi *Load balancing* dapat berjalan dilakukan penambahan pada beberapa *rules mangle* pada *router* seperti pada Gambar 4.

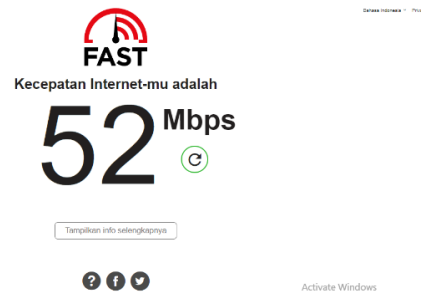


Gambar 4. Rules Mangel

*Rules* yang pertama yaitu dengan melakukan *Mark Conection* untuk menandakan koneksi pada *port ether* yang ditujukan kepada koneksi *port* INET 1 dan INET 2. Lalu yang kedua dilakukan *mark routing* untuk menandakan jalannya koneksi internet pada *port ether* ke masing-masing ISP pada *port ether*. Dan yang ketiga dilakukan lagi *Mark Conection* pada masing-masing INET 1 dan INET 2. Dan yang terakhir dilakukan *Mark Routing* untuk menandai dan mengatur jalur *output* dari koneksi internet INET 1 dan INET 2. Setelah konfigurasi *mangle* dilakukan kemudian dilakukan konfigurasi *routing* pada INET 1 dan INET 2 ke masing-masing *output interface* sehingga terdapat keterangan *reachable*. Pada tahap ini dapat dipastikan bahwa konfigurasi *Load balancing* pada jaringan internet dengan metode *Per Conection Classifier* (PCC) dapat berjalan dengan semestinya.



Gambar 5. Hasil *Speedtest* Sebelum Penerapan Metode PCC



Gambar 6. Hasil *Speedtest* Sesudah Penerapan Metode PCC

Setelah penerapan metode *Per Connection Classifier* dilakukan, selanjutnya dilakukan penerapan metode *Ip Hash* untuk membantu PC Client yang mengakses *web server* agar dapat berjalan dengan lancar. Sebelum client mengakses *web server*, terlebih dahulu akan melewati *load balancer*. *Load balancer* tersebut terhubung dengan *jaringan client* dan *jaringan server web*. Agar *Ip Hash* dapat berjalan pada *load balancer*, ditambahkan *source code algoritma Ip Hash* pada *load balancer*.

```

if tcp_header.dst_port == 80:
    Source_ip = ip_header.src
    arr_aktet = str(source_ip).split('.')

    biner1 = bin(int(arr_aktet[0]))
    biner2 = bin(int(arr_aktet[1]))
    biner3 = bin(int(arr_aktet[2]))
    biner4 = bin(int(arr_aktet[3]))

    biner_source_ip = (biner1 + str('.') + biner2 + str('.') +
                      biner3 + str('.') + biner4)

    # Hashing
    hashing = md5(biner_source_ip)
    mod = hashing % 3
    index = mod

    server_mac_selected =
self.serverlist[index]['mac']
server_ip_selected =
self.serverlist[index]['ip']
server_outport_selected =
int(self.serverlist[index]['outport'])
index = index + 1

print("biner", biner_source_ip)
print("hash", hashing)
print("mac", server_mac_selected)
print("ip", server_ip_selected)
print("outport", server_outport_selected)

print("server", server_ip_selected)

```

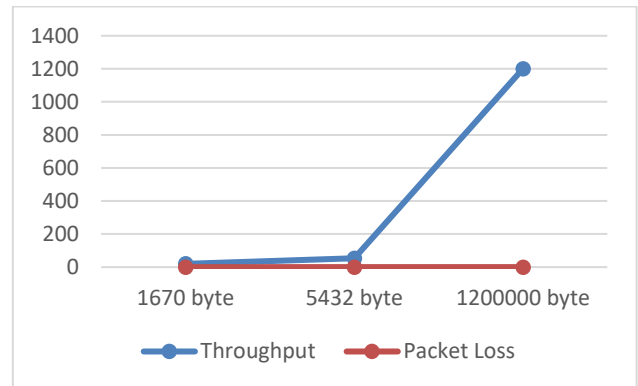
Gambar 7. *Source Code Ip Hash*

Agar klien dapat menggunakan layanan HTTP yang disediakan oleh *server*, klien harus mengetahui alamat IP *Load Balancer*. *Load Balancer* akan mengirimkan permintaan HTTP klien ke *web server*. Setelah konfigurasi dilakukan, selanjutnya dilakukan pengujian untuk melihat hasil dari performa *Ip Hash*.

Tabel 2. Pengujian *Traffic*

| No | Server     | Banyak Request | Source Ip Hash              |
|----|------------|----------------|-----------------------------|
| 1  | WEBSERVER1 | 3              | 117.168.1.2 dan 117.168.1.4 |
| 2  | WEBSERVER2 | 3              | 117.168.1.3                 |
| 3  | WEBSERVER3 | 3              | 117.168.5.3 dan 117.168.5.4 |
| 4  | WEBSERVER4 | 3              | 117.168.5.2                 |

Pada tabel 2 merupakan hasil saat menguji metode *Ip Hash* dengan jumlah client yang digunakan sebanyak 3 client. Berdasarkan hasil saat menjalankan *Ip Hash*, *Web Server 1* menerima request dari 117.168.1.2 dan 117.168.1.4 yang merupakan PC 1 dan 3. *Web Server 2* menerima request dari 117.168.1.3 yang merupakan PC 2. *Web Server 3* menerima request dari 117.168.5.3 dan 117.168.5.4 yang merupakan PC 5 dan PC 6. *Server 4* menerima request dari 117.168.5.2 yang merupakan PC 4.



Gambar 8. Pengujian Koneksi *Web Server*

Pada Gambar 8 merupakan hasil saat menguji metode *Ip Hash* dengan pengiriman request sebanyak 60 request dengan ukuran data yang bermacam-macam. Pengujian *web server* dilakukan dengan menggunakan *httperf* pada *host client*. Paket yang dikirim sebesar 1670 byte dengan kategori rendah, 5432 byte dengan kategori menengah, dan 1,2 MB dengan kategori tinggi. Hasil dari pengujian ditemukan bahwa rata-rata *throughput* dengan kategori request data kecil sebesar 16,69 KB/s,

dengan kategori *request* data menengah sebesar 54,31KB/s, dan 1200 KB/s untuk kategori *request* data besar. Hasil pengujian dari *request* sebanyak 60 *request* dengan *rate* 10 *request/second* dapat menghasilkan rata-rata *packet loss* sebanyak 0% bagi data kecil, 0% bagi data menengah dan 0% bagi data yang berukuran >1MB.

#### 4. Kesimpulan

Berdasarkan hasil penelitian yang didapatkan, dapat disimpulkan bahwa hasil dari penelitian ini ialah: Pertama, Metode *load balancing* PCC menghasilkan beban trafik yang lebih lancar tanpa penumpukan data. Apabila salah satu dari beberapa jaringan tersebut mati, maka akses internet akan tetap tersedia dikarenakan jaringan yang lainnya akan berfungsi sebagai penyedia internet pengganti. Apabila setiap komputer dapat mengakses jaringan maka komputer tersebut akan secara otomatis memilih jaringan yang telah ditentukan. Kedua, Penggunaan metode *Ip Hash* dalam menyeimbangkan jalur request pada tiap *client* terbukti lebih baik tanpa menggunakan metode *Ip Hash*. Hal itu dapat terjadi dikarenakan metode *Ip Hash* menghitung nilai-nilai hash dari tiap *client* dan mendapatkan nilai yang mengarahkan permintaan kepada salah satu *server* yang dituju.

#### 5. Daftar Pustaka

- [1] Sumiati, A., Trisnawan, P.H. and Fauzi, M.A., 2020. Implementasi Load Balancing Web Server dengan Algoritma Source IP Hash pada Software Defined Network (SDN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(3), pp.919-928.
- [2] Elhanafi, A.M., Lubis, I., Irwan, D. and Muhazir, A., 2018. Simulasi Implementasi Load Balancing PCC Menggunakan Simulator Gns3. *JURNAL TEKNOLOGI DAN ILMU KOMPUTER PRIMA (JUTIKOMP)*, 1(2), pp.159-165. DOI: <https://doi.org/10.34012/jutikomp.v1i2.236>.
- [3] Nuraini, R., 2022. Implementasi Metode Load Balancing Untuk Peningkatan Nilai Troughput Pada Server. *KLIK-KUMPULAN JURNAL ILMU KOMPUTER*, 9(3), pp.467-478. DOI: <http://dx.doi.org/10.20527/klik.v9i3.524>.
- [4] Laksamana, A.R., 2021. Implementasi Dan Analisis Load Balancing Pada Router Pfsense. *Universitas Srinwijaya, Palembang*.
- [5] Aldori, R., 2021. Implementasi Load Balancing Menggunakan Metode PCC (Per Connection Classifier) Berbasis Mikrotik pada SMK Tunas Harapan Jakarta. *TECHSI-Jurnal Teknik Informatika*, 13(2), pp.69-82. DOI: <https://doi.org/10.29103/techsi.v13i2.5380>.
- [6] Amalia, R., 2020. Analisis Kinerja Web Server Menggunakan Metode Load Balancing As a Service Pada Lingkungan Virtual Openstack. Skripsi, Politeknik Negeri Jakarta.
- [7] Apriliansyah, F., Fitri, I. and Iskandar, A., 2020. Implementasi Load Balancing Pada Web Server Menggunakan Nginx. *Jurnal Teknologi dan Manajemen Informatika*, 6(1), pp.18-26. DOI: <https://doi.org/10.3997/2214-4609.201801770>.
- [8] Bhayangkara, F.J. and Riadi, I., 2014. Implementasi Proxy Server Dan Load Balancing Menggunakan Metode Per Connection Classifier (Pcc) Berbasis Mikrotik (Studi Kasus : Shmily.Net). *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)*, 2(2), pp.1206–1217. DOI: <http://dx.doi.org/10.12928/jstie.v2i2.2729>.
- [9] Budiyono, S.E., Rohana, T. and Al Mudzakir, T., 2021. Penggunaan Load Balancing Pada Web Server Lokal Dengan Metode Policy Based Routing. *Scientific Student Journal for Information, Technology and Science*, 2(1), pp.108-112. DOI: <https://doi.org/10.53513/jis.v20i2.3742>.

- [10] Cloud, O., 2022. Load Balancer Policies. Available at: <https://docs.oracle.com/en-us/iaas/Content/Balance/Reference/lbpolicies.htm>. Diakses 18 Mei 2023.
- [11] Dewi, C., 2023. Scientific Article Publication Training in Reputable International Journals. *Jurnal Pemberdayaan Masyarakat Madani (JPMM)*, 7(1), pp.114-126. DOI: <https://doi.org/10.21009/JPMM.007.1.10>
- [12] Dewi, C., Chen, A.P.S. and Christanto, H.J., 2023. Recognizing similar musical instruments with YOLO models. *Big Data and Cognitive Computing*, 7(2), p.94. <https://doi.org/10.3390/bdcc7020094>.
- [13] Dewi, C., Chen, R.C., Yu, H. and Jiang, X., 2023. XAI for Image Captioning using SHAP. *Journal of Information Science & Engineering*, 39(4). DOI: <https://doi.org/10.6688/JISE.202307>.
- [14] Dewi, C., Chen, R.C., Liu, Y.T. and Yu, H., 2021. Various generative adversarial networks model for synthetic prohibitory sign image generation. *Applied Sciences*, 11(7), p.2913. DOI: <https://doi.org/10.3390/app11072913>.
- [15] Dewi, C. and Juli Christanto, H., 2022. Combination of deep cross-stage partial network and spatial pyramid pooling for automatic hand detection. *Big Data and Cognitive Computing*, 6(3), p.85. DOI: <https://doi.org/10.3390/bdcc6030085>.
- [16] Hakim, A.R., 2018. Penerapan Load Balancing Pada Router Pfsense Berbasis Free Bsd. *Edik Informatika*, 4(1), pp.23–28. DOI: <https://doi.org/10.22202/ei.2017.v4i1.2534>.
- [17] Tanenbaum, A.S., 2001. Organisasi komputer terstruktur, jilid 1. *Diterjemahkan dari Bahasa Inggris oleh TAH Al-Hamdany*. Jakarta: Salemba Teknika.
- [18] Ibrahim, I.M., Ameen, S.Y., Yasin, H.M., Omar, N., Kak, S.F., Rashid, Z.N., Salih, A.A., Salim, N.O.M. and Ahmed, D.M., 2021. Web Server Performance Improvement Using Dynamic Load Balancing Techniques: A Review. *Asian Journal of Research in Computer Science*, 10(1), pp.47–62. DOI: <https://doi.org/10.9734/ajrcos/2021/v10i130234>.
- [19] Jader, O.H., Zeebaree, S.R.M. and Zebari, R.R., 2019. A state of art survey for web server performance measurement dan load balancing mechanisms. *International Journal of Scientific dan Technology Research*, 8(12), pp.535–543.
- [20] Kurniawan, R. and Putri, A.D., 2019. Perancangan Jaringan menggunakan Metode Load Balancing PCC, Failover, dan Notifikasi. *Sentinel*, 2(2), pp.166–179. DOI: <https://doi.org/10.56622/sentineljournal.v2i2.15>.
- [21] Hadi, S. and Pramono, B.A., 2022. Analisa Komparasi Metode Pembagian Trafik Jaringan (Load Balancing) antara Metode PCC dan Metode ECMP: Studi Kasus pada Jaringan USM. *Indonesian Journal of Computer Science*, 11(3). DOI: <https://doi.org/10.33022/ijcs.v11i3.3095>.
- [22] Nugroho, K.T., Julianto, B., Tisna, D.R. and Nur, D.F., 2023. Quality Analysis of Service Load Balancing Using PCC, ECMP, and NTH Methods. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, 12(1). DOI: <https://doi.org/10.23887/janapati.v12i1.55894>.
- [23] Ohhyver, J.T. and Chandra, D.W., 2023. Simulasi Keamanan Jaringan pada DPDK OpenvSwitch Berbasis Network-Based Intrusion Detection System (NIDS). *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, 7(3) (Jul. 2023), pp.368–374. DOI: <https://doi.org/10.35870/jtik.v7i3.845>.

- [24] Saputra, F.W., Aspriyono, H. and Al Akbar, A., 2022. Designing Rt/Rw Net In Implementing Load Balancing Using Per-Connection Classifier Method In Bumi Mas Housing. *Jurnal Komputer*, 1(1), pp.65-72.
- [25] Suherman, S., Aziz, M. and Nababan, E.B., 2019, June. Load balancing algorithm for a local video network. In *Journal of Physics: Conference Series* (Vol. 1235, No. 1, p. 012018). IOP Publishing. DOI: <https://doi.org/10.1088/1742-6596/1235/1/012018>.
- [26] Suryanto, S., Prasetyo, T. and Hikmah, N., 2018. Implementasi load balancing menggunakan metode per connection classifier (PCC) dengan failover berbasis mikrotik router (studi kasus PT. Sumber Rejeki Power). *SNIT* 2018, 1(1), pp.230-238.
- [27] Tanton, A., Zaen, M.T.A. and Mutawalli, L., 2022. Komparasi QoS Load Balancing Pada 4 Line Internet dengan Metode PCC, ECMP dan NTH. *Jurnal Media Informatika Budidarma*, 6(1), pp.110-119. DOI: <https://doi.org/10.30865/mib.v6i1.3436>.
- [28] Yulianto, I., 2022. *Implementasi Reverse Proxy Server Sebagai Load Balancing dan Https Proxy Menggunakan Nginx Pada Google Cloud Platform* (Tugas Akhir, Universitas Mercu Buana).