

# Instalasi Serta Konfigurasi HAproxy Sebagai Load Balancing Web Server

Hafizd Setiawan\*  
Sistem Komputer, Universitas Sriwijaya, Indonesia

\*Korespondensi: hafizdsetiawan21@gmail.com

---

## ARTICLE INFO

### Article History:

- Received 12 January 2023
- Received in revised form 14 January 2023
- Accepted 12 February 2023
- Available online 30 March 2023

---

## ABSTRAK

Internet, sebagai kumpulan komputer yang digunakan oleh pengguna terhubung, memegang peran vital dalam berkomunikasi dan menyimpan data melalui website. Namun, server sering mengalami masalah akibat jumlah client yang tinggi. Penelitian ini menggunakan HAproxy, aplikasi open source untuk load balancing, sebagai solusi. HAproxy tidak hanya mendistribusikan permintaan dari client ke server, tetapi juga mengatasi situasi failover ketika salah satu server tidak tersedia. Hasil studi load balancing menunjukkan bahwa kecepatan respon keseluruhan bergantung pada spesifikasi server dan throughput jaringan. Kualitas jaringan dan spesifikasi server saling berhubungan; ketidakseimbangan keduanya dapat mengakibatkan respons yang kurang tanggap. Penyeimbang beban berhasil mengambil file dari server dan menjalankan tugasnya dengan baik, terutama saat server tidak tersedia. Implementasi load balancing, seperti menggunakan HAproxy, dapat meningkatkan kinerja jaringan, terutama dalam mendukung kebutuhan jaringan kecil.

Kata Kunci: Internet, Server, Komputer, Load Balancing, HAproxy

---

## ABSTRACT

*The Internet, as a collection of computers used by connected users, plays a vital role in communication and data storage through websites. However, servers often encounter issues due to a high number of clients. This study utilizes HAproxy, an open-source application for load balancing, as a solution. HAproxy distributes requests from clients to servers and handles failover situations when one server is unavailable. The results of the load balancing study indicate that the overall response speed depends on server specifications and network throughput. Network quality and server specifications are interrelated; their imbalance can lead to unresponsive responses. The Load Balancer successfully retrieves files from the server and performs its task well, especially when a server is unavailable. Implementing load balancing, such as using HAproxy, can enhance network performance, particularly in supporting the needs of small networks.*

*Keywords: Internet, Server, Computer, Load Balancing, HAproxy.*

---

## 1. PENDAHULUAN

Jauh sebelum media elektronik berkembang menjadi seperti sekarang ini, sebelumnya masyarakat menggunakan media cetak untuk bertukar informasi, cara ini memiliki banyak kelemahan. Salah satu kelemahan media cetak ialah penyebaran informasinya sangat lambat serta hanya bersifat lokal. Jika manusia dapat mengembangkan komputer, dibantu dengan berkembangnya infrastruktur internet, manusia dapat bertukar informasi lebih cepat, lebih luas, serta lebih murah[1]. Internet sangat diperlukan bagi banyak orang saat ini. Internet ialah kumpulan komputer yang digunakan oleh orang-orang yang terhubung serta berbagi aturan umum untuk pertukaran informasi antara pengguna Internet. Untuk terhubung ke internet,

setiap komputer juga harus terhubung dengan network connection provider sebagai penyedia koneksi jaringan. Banyak orang saat ini menggunakan internet untuk chatting, surfing, media perdagangan, berbagi informasi, dll. Saat seseorang ingin berselancar, komputer membutuhkan aplikasi browser, saat seseorang menjelajah, browser mengirimkan permintaan data ke server web, setelah itu server web mengirimkan respons yang diminta. Menampilkan informasi tagihan, menampilkan pesan bahwa Peningkatan jumlah pengguna Internet mengembangkan jumlah kunjungan ke web server. Setiap server web memiliki kapasitas layanan yang dapat melebihi satu hari karena beban pengguna yang tinggi. Untuk mengatasi masalah ini, pemilik jaringan menambahkan server sesuai kebutuhan untuk memaksimalkan layanan. Menambah jumlah server ternyata tidak cukup tanpa pengaturan yang disesuaikan dengan jenis website. Ada banyak jenis situs web seperti belanja online, situs web bisnis, berbagi media, bookmark sosial, penyimpanan cloud, jejaring sosial, serta lainnya[2].

Setiap website memiliki jenis penggunaan yang berbeda-beda, misalnya website jenis cloud storage memiliki pengguna yang membutuhkan bandwidth lebih banyak dibandingkan website korporat karena pengguna cloud storage sering kali menyimpan serta mengunggah data[3]. Anda dapat memaksimalkan kinerja jaringan dengan lebih dari satu server web dengan menggunakan load balancing untuk mendistribusikan permintaan dari klien ke server. Load balancing berguna untuk mencegah server crash karena banyaknya permintaan dari client, ataupun untuk mencegah client mengunjungi web server yang sedang tidak tersedia. Load balancing bekerja dengan mendistribusikan permintaan klien ke server sehingga server tidak memproses permintaan tambahan. Dengan cara ini, permintaan yang diproses oleh masing-masing server dapat disesuaikan dengan kekuatan masing-masing server, metode distribusi yang digunakan, ataupun algoritme yang digunakan. Ada dua jenis load balancer, ialah load balancing software serta load balancing hardware[4].

Load balancing hardware sering dijual di pasaran dengan harga yang cukup mahal. Perangkat lunak load balancing biasanya didistribusikan secara gratis di Internet. Contoh software yang menyediakan load balancing ialah Nginx, Seesaw, LoadMaster dari KEMP, HAProxy, ZEVENET, Neutrino, Balance, Pen, Nginx, dll. HAProxy ialah salah satu dari banyak aplikasi load balancing yang umum digunakan. HAProxy ialah aplikasi open source gratis yang dapat digunakan di sistem operasi Linux[5].

## 2. TINJAUAN PUSTAKA

### A. Jaringan Komputer

Jaringan komputer ialah sekelompok komputer yang terhubung dengan kabel ataupun gelombang radio untuk bertukar data. Pada dasarnya jaringan komputer dibangun dengan menggunakan teknologi informasi serta teknologi komunikasi yang memiliki ukuran yang berbeda-beda mulai dari jaringan LAN hingga jaringan internet. Tanpa kita ketahui, teknologi ini berkembang sangat cepat serta sangat bermanfaat bagi penggunaannya, mulai dari pertukaran informasi hingga pembayaran online. Jaringan komputer juga dapat membantu berbagi sumber daya yang dapat dibagi, mengurangi biaya pembelian peralatan[6].

Berdasarkan kegunaannya, klasifikasi jaringan dapat dibagi menjadi dua jenis ialah jaringan client-server serta jaringan peer-to-peer. Jaringan client-server ialah jaringan yang salah satunya menjadi hub serta membantu komputer lain. Dalam hal ini, jaringan client-server memiliki ciri khas, setiap klien membuat permintaan ke server tentang kebutuhan, kemudian server merespons sesuai dengan permintaan serta sumber daya yang tersedia di server[7]. Jaringan klien-server berskala besar ini dapat menghubungkan klien dengan klien di belahan dunia lain, melalui video streaming ataupun media sosial. Jaringan client-server memiliki beberapa keuntungan, seperti: Kontrol terpusat, dimana server memiliki otoritas untuk mengontrol akses setiap klien. Dibandingkan dengan jaringan peer-to-peer, jaringan client-

server memiliki cakupan yang lebih besar serta luas. Disisi lain terdapat titik lemah pada jaringan client-server, jika terjadi kesalahan pada control center maka tidak semua client dapat terlayani, terlalu banyak user yang memperlambat jaringan client-server[8].

Ketika jaringan client-server menjadikan satu komputer sebagai pembantu, itu berbeda dengan jaringan peer-to-peer, dalam jaringan peer-to-peer semua komputer yang terhubung bekerja sama serta berbagi sumber daya serta layanan untuk menyelesaikan suatu tugas. Setiap komputer dapat menggunakan aplikasi yang berada di komputer lain, misalnya komputer 1 membuka aplikasi pengolah kata di komputer 2 lalu mencetak langsung dari komputer 2[9].

## **B. Teknolgi Local Area Network (LAN)**

LAN (Local Area Network) ialah bentuk jaringan terkecil di antara banyak jenis jaringan lainnya. Dalam praktiknya, LAN terdiri dari beberapa komputer yang terhubung ke satu jaringan ataupun area jangkauan jaringan. Jaringan LAN biasanya dihubungkan dengan switch ataupun hub[10]. Dalam membangun jaringan LAN ada beberapa topologi yang dapat dipilih mulai dari jaringan LAN point to point, jaringan LAN bus, jaringan LAN ring, jaringan LAN Star, jaringan LAN tree. Setiap topologi memiliki kelebihan serta kekurangan masing-masing tergantung kebutuhan Anda. Saat Anda menangani komputer dengan alamat IP pribadi di jaringan LAN, setiap jaringan mewakili beberapa komputer secara bersamaan. Keuntungan menggunakan jaringan LAN ialah ukurannya yang kecil memudahkan untuk memecahkan masalah jika terjadi kegagalan, serta memungkinkan berbagi sumber daya di jaringan dengan mudah. Namun, LAN memiliki kerentanan yang rentan terhadap kecerobohan pengguna, seperti mudah menebak kata sandi komputer ataupun menyebarkan virus ke seluruh jaringan[11].

## **C. Hypertext Transfer Protocol (HTTP)**

Hypertext Transfer Protocol, seperangkat aturan untuk pengiriman file dalam bentuk gambar, teks, gambar, audio, video serta file multimedia lainnya. Protokol HTTP berada di lapisan aplikasi lapisan OSI, yang digunakan oleh browser sebagai klien untuk server sebagai pemilik situs web. Saat client membuka browser, client memasukkan kata kunci ataupun alamat host yang berkunjung [12]. Ketika client memasukkan kata kunci, mesin pencari yang digunakan memproses kata kunci tersebut, kemudian mesin pencari memberikan beberapa alamat email yang direkomendasikan beserta artikel tentang kata kunci tersebut, ketika client mengklik alamat email yang direkomendasikan, browser mengubah alamat itu bagaimana meminta pelanggan ke server tujuan harus dikirim. Jika server yang diinginkan dapat dijangkau, server mengirimkan tanggapan atas permintaan klien, yang kemudian ditampilkan oleh browser web klien[13].

## **D. Load Balancing**

Banyaknya masalah pada server yang diakibatkan oleh overload request dari client menyebabkan server menjadi bad client-server. Sulit dijangkau karena server overload ataupun "lamban" dalam kehidupan sehari-hari[15]. Perangkat berbagi beban dapat beroperasi di beberapa lapisan jaringan komputer, penyeimbang beban menggunakan aliran data dari klien ke server melalui penyeimbang beban. Tujuan utama dari load balancing ialah untuk mendistribusikan lalu lintas permintaan klien sesuai dengan algoritma yang digunakan untuk mencegah kelebihan server. Load balancing memiliki dua sisi ialah depan serta belakang. UI menyertakan lisensi IP server yang terkait dengan penyeimbang muatan. Backend mencakup inisialisasi identitas penyeimbang beban, yang mencakup alamat IP penyeimbang beban, waktu koneksi, serta lainnya[16].

Load balancing ialah jaringan komputer yang menggunakan metode untuk mendistribusikan beban kerja ke dua ataupun bahkan lebih port jaringan secara seimbang, sehingga bekerja secara optimal serta kabel koneksi tidak kelebihan beban (overloaded). Load balancing juga dapat digambarkan sebagai koneksi dari dua ataupun lebih jaringan yang digabungkan menjadi sebuah router serta terhubung ke server serta klien. Dengan bantuan layanan load balancing, sumber daya jaringan dapat didistribusikan ke beberapa host lain secara non-sentral, sehingga kinerja jaringan komputer secara keseluruhan tetap stabil. Ketika pengguna mengakses server, sebenarnya server menerima beban karena harus memproses permintaan yang ditujukan kepada penggunanya. Ketika ada banyak pengguna, ada banyak proses. Sesi komunikasi dibuka oleh server untuk memungkinkan pengguna menerima layanan dari server. Bahkan jika hanya satu server yang kelebihan beban, server tersebut tidak dapat melayani banyak penggunanya karena daya pemrosesan yang terbatas. Solusi ideal ialah mendistribusikan beban yang masuk ke beberapa server. Sehingga yang melayani pengguna tidak hanya fokus pada satu perangkat saja[18].

#### **E. Haproxy**

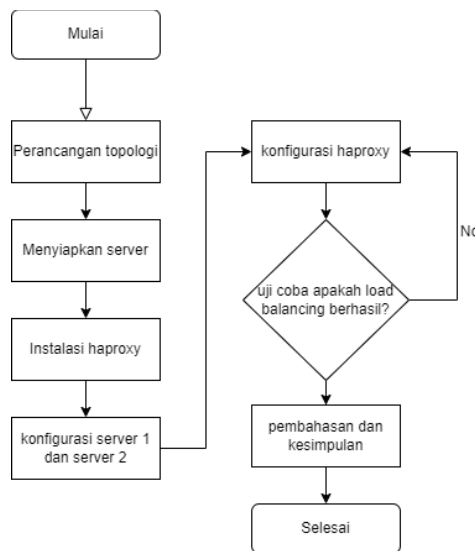
Server proxy ialah komputer ataupun perangkat lunak yang bertindak sebagai perantara antara klien serta server web yang berkunjung. Proksi berisi perintah yang disesuaikan dengan kebutuhan Anda. HAProxy ialah aplikasi open source yang dapat digunakan untuk load balancing. Selain load balancing, haproxy dapat mengatasi kegagalan karena salah satu server tidak dapat menggunakan haproxy yang mendukung banyak algoritma[17].

#### **F. Round Robin**

Algoritma round-robin ialah algoritma yang paling sederhana serta paling banyak digunakan dalam penyeimbang beban. Algoritme round robin bekerja dengan bergantian serta mendistribusikan beban secara berurutan dari satu server ke server lainnya. Ide dasar dari algoritma rangkaian ialah menggunakan pembagian waktu, pada dasarnya algoritma ini memproses antrian secara berurutan. Round Robin ialah salah satu algoritma penjadwalan proses dari sistem operasi. Penulisan melingkar dirancang untuk membagi waktu setiap proses menjadi bagian yang sama serta dalam urutan melingkar. Eksekusi semua proses tanpa prioritas juga dikenal sebagai eksekusi siklik. Perencanaan round robin mudah diterapkan. Penjadwalan round-robin juga dapat diterapkan pada masalah penjadwalan lainnya seperti: penjadwalan paket data dalam jaringan komputer. Round Robin dirancang untuk skema pembagian waktu[19].

### **3. METODELOGI PENELITIAN**

Dalam penelitian ini penulis melakukan observasi dengan cara melakukan pengamatan secara sistematis terhadap masalah yang dihadapi. Lalu penulis melakukan studi pustaka untuk mendapatkan informasi yang diperlukan baik melalui media pembelajaran seperti buku, contoh laporan, serta browsing internet mencari artikel – artikel yang berkenaan dengan tema yang diangkat penulis yakni tentang Load Balancing HAPROXY. Selanjutnya di implementasikan dan hasilnya akan dibahas, dan kesimpulan akan diambil dari penelitian ini. Pada gambar 1 bentuk flowchart dari penelitian

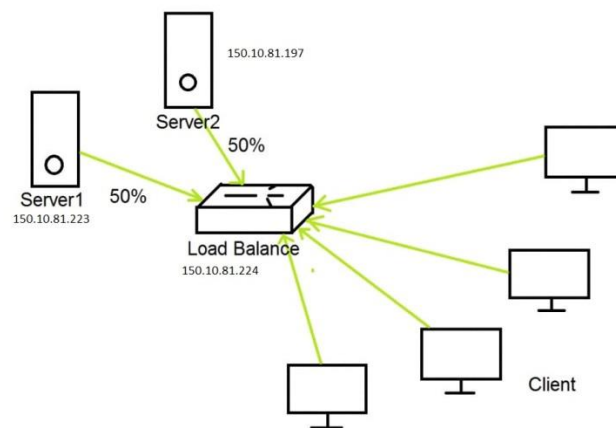


Gambar 1. Flowchart metodologi penelitian

#### 4. HASIL PENELITIAN

##### 1. Topologi

Pada gambar 2 adalah topologi yang saya gunakan pada penelitian kali ini, Dimana ip dari 150.10.81.224 akan menjadi load balancer dari kedua web server yang sudah dipersiapkan. Nantinya client akan mengakses ip dari load balancer dan akan diarahkan ke web server sampai pembagian dari beban web server merata



Gambar 2. Topologi load balancing

##### 2. Menyiapkan Server

Disini saya menyiapkan 3 server,yang pertama sebagai Load Balancer, Kemudian 2 server sebagai Web Server dengan IP yang berbeda-beda

```
File Machine View Input Devices Help
apis@linux:~$ ifconfig
enp0s3: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.18.0.1 netmask 255.255.0.0 broadcast 172.18.255.255
    ether 02:4c:19:15:18:7a5 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:25:13:e4:3e txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 150.10.81.224 netmask 255.255.255.0 broadcast 150.10.81.255
    inet6 fe80::a0027fff:fe72:93bb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:72:a3:bb txqueuelen 1000 (Ethernet)
    RX packets 1158 bytes 105226 (105.2 KB)
    RX errors 0 dropped 54 overruns 0 frame 0
    TX packets 275 bytes 20541 (20.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 154 bytes 12364 (12.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 154 bytes 12364 (12.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

apis@linux:~$
```

Gambar 3. IP load balancer : 150.10.81.224

IP Load Balancer : 150.10.81.224

IP Web Server 1 : 150.10.81.197

IP Web Server 2 : 150.10.81.223

### 3. Instalasi serta Konfigurasi HAProxy pada Server Load Balancing

- Langkah pertama melakukan update pada ubuntu server dengan command “*sudo apt update*”

```
File Machine View Input Devices Help
apis@linux:~$ sudo apt update
[sudo] password for apis:
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:3 https://repo.zabbix.com/zabbix/5.5/ubuntu focal InRelease
Get:4 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 11s (31.7 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
apis@linux:~$
```

Gambar 4. Update Virtual Machine Ubuntu

- Setelah di update lakukan upgrade dengan menetik command “*sudo apt upgrade*”

```
File Machine View Input Devices Help
apis@linux:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0
Use 'sudo apt autoremove' to remove them.
#
# News about significant security updates, features and services will
# appear here to raise awareness and perhaps tease /r/Linux :)
# Use 'pro config set apt_news=false' to hide this and future APT news.
#
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
apis@linux:~$
```

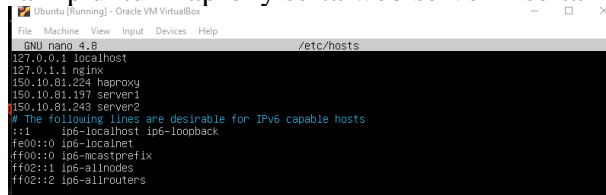
Gambar 5. Upgrade Virtual Machine Ubuntu

- Menginstall HAProxy dengan command “*sudo apt install haproxy*”

```
File Machine View Input Devices Help
apis@linux:~$ sudo apt install haproxy
Reading package lists... Done
Building dependency tree
Reading state information... Done
haproxy is already the newest version (2.0.29-0ubuntu1).
The following packages were automatically installed and are no longer required:
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
apis@linux:~$
```

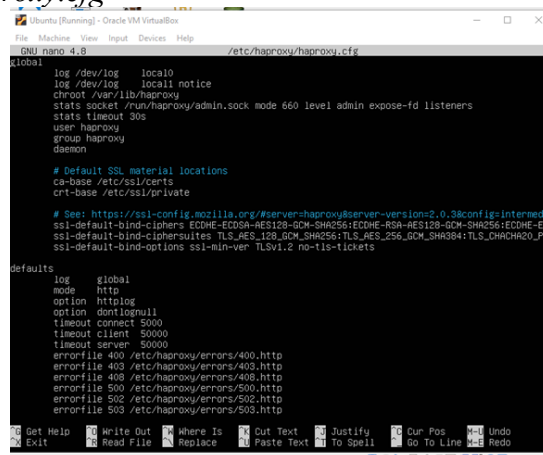
Gambar 7. Install HAProxy

- Langkah selanjutnya kita memberitahukan kepada host ubuntu ip apa saja yang terlibat dalam load balancing ini dengan mengetik perintah “*sudo nano /etc/hosts*”. Kemudian tambahkan ip untuk haproxy serta web server 1 serta 2



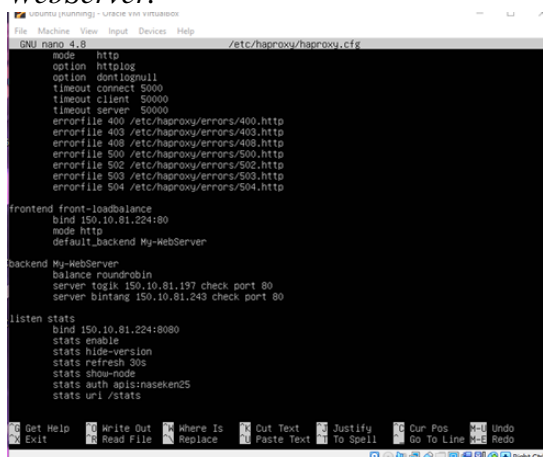
Gambar 8. Membuat Direktori pada setiap Hosts / client

- Selanjutnya konfigurasi frontend,backend dari Haproxy dengan perintah “*sudo nano/etc/haproxy/haproxy.cfg*”



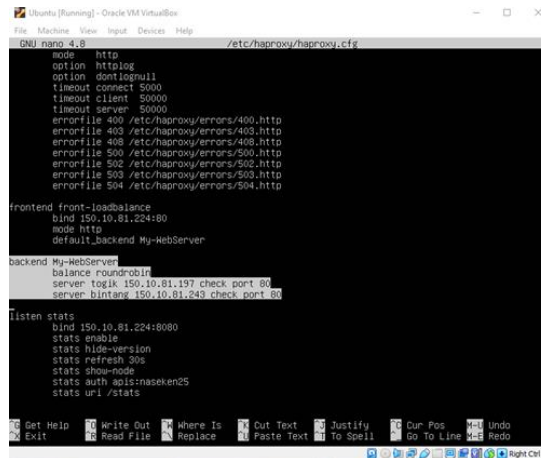
Gambar 9. Konfigurasi Frontend serta Backend pada direktori haproxy.cfg

- Lalu pada bagian bawah menambah beberapa konfigurasi ialah frontend dengan ip yg ingin kita gunakan sebagai load balancer  
*frontend front-loadbalance*  
*bind 150.10.81.224:80*  
*mode http*  
*default\_backend My-WebServer.*



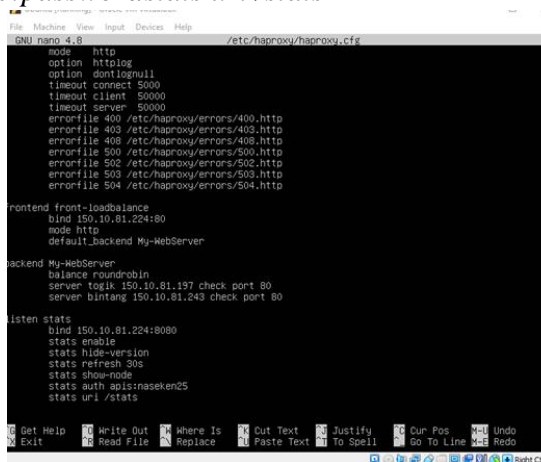
Gambar 10. Menyunting file direktori haproxy.cfg

- Tambahkan konfigurasi backend server pada bagian bawah:  
*backend My-WebServers balance roundrobin*  
*server togik 150.10.81.197 check port 80*  
*server bintang 150.10.81.243 check port 80*



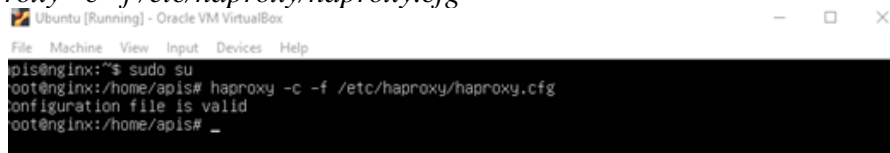
Gambar 11. Membuat backend pada setiap client

- Tambahkan enable stat untuk monitoring status load balancing:  
*listen stats*  
*bind 150.10.81.224:8080*  
*stats hide-version*  
*stats refresh 30s*  
*stats show-node*  
*stats auth username:password*  
*stats uri /stats*



Gambar 12. Mengaktifkan port stat status load balancing

- Save, kemudian cek status HAProxy > *systemctl restart haproxy.service*  
Jika langkah sebelumnya Valid, maka restart HAProxy service menggunakan perintah  
> *haproxy -c -f /etc/haproxy/haproxy.cfg*

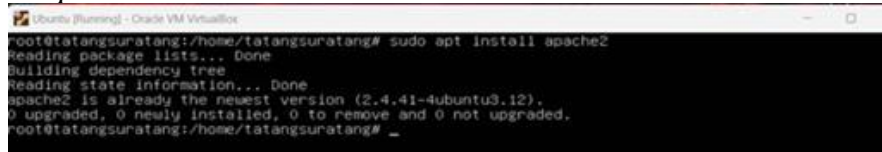


Gambar 13. Restart HAProxy



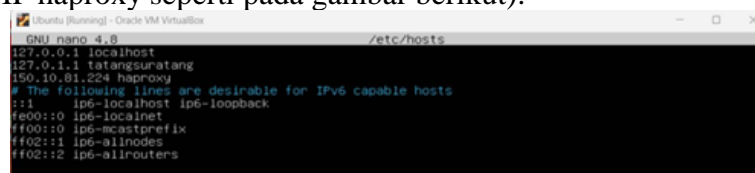
#### 4. Konfigurasi Server 1 serta Server

- Langkah pertama yang dilakukan ialah menginstall apache2 dengan perintah `>sudo apt install apache2`



Gambar 14. Install Apache2

- Setelah apache berhasil terinstall, langkah selanjutnya ialah mengedit file konfigurasi di file host dengan menggunakan perintah `> sudo nano /etc/hosts` (Padafile tersebut masukkan IP haproxy seperti pada gambar berikut):



Gambar 15. Membuat IP pada setiap hosts

- Setelah itu, aktifkan service apache2 dengan perintah `> sudo systemctl enable apache2`



Gambar 16. Mengaktifkan Port Apache2

- Kemudian restart apache2 menggunakan perintah `> sudo systemctl restart apache2`



Gambar 17. Restart Apache2

- Masuk ke directory web host: `> cd /var/www/html`

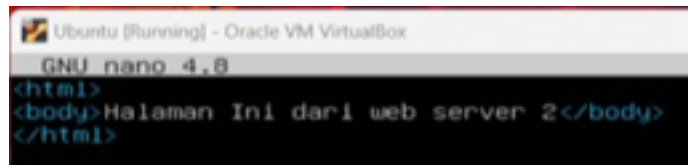


Gambar 18. Masuk dalam kedirektori HTML

- Buat index.html dengan perintah `> sudo touch index.html`
- Buat halaman sederhana pada index.html `> sudo nano index.html`, contohnya:  
`<html>`  
`<body> Ini halaman web dari node web server 1</body>`  
`</html>`

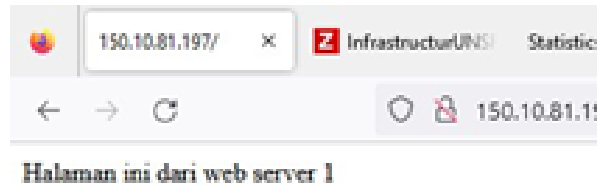


Gambar 19. Membuat HTML pada setiap Hosts

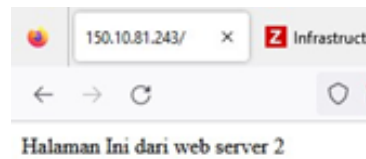


Gambar 20. Membuat HTML pada setiap Hosts

- Selanjutnya uji coba akses melalui browser dengan IP masing masing



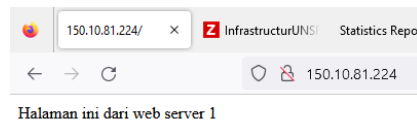
Gambar 21. Pengujian di setiap client / hosts



Gambar 22. Pengujian di setiap client / hosts

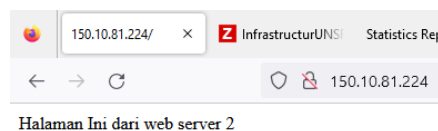
## 5. Uji Coba Load Balancing

- Pengujian pertama dilakukan dengan memasukan IP dari HAProxy maka akan menampilkan halaman web server pertama



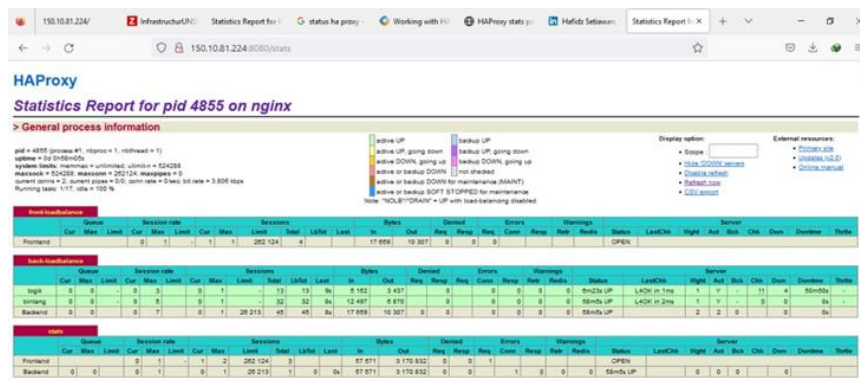
Gambar 23. Pengujian pada web server 1 (Server Togik)

- Jika halamannya di refresh akan menampilkan halaman web server kedua, disinilah algoritma round robin bekerja



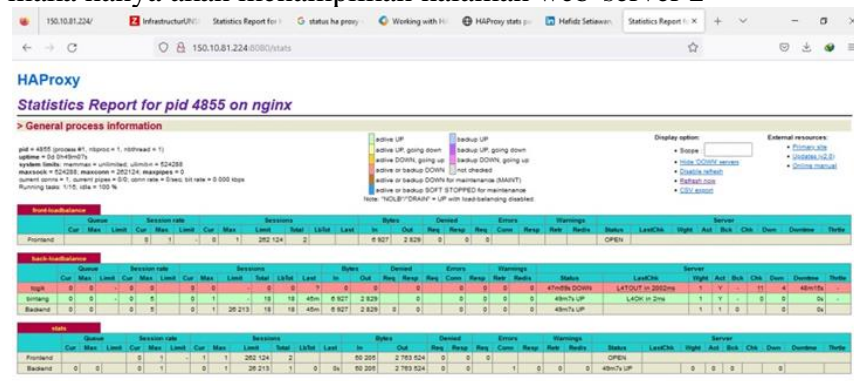
Gambar 24. Pengujian pada web server 2 (Server Bintang)

- Akses ip stats pada ip haproxy contoh URL: <http://150.10.81.224:8080/stats>



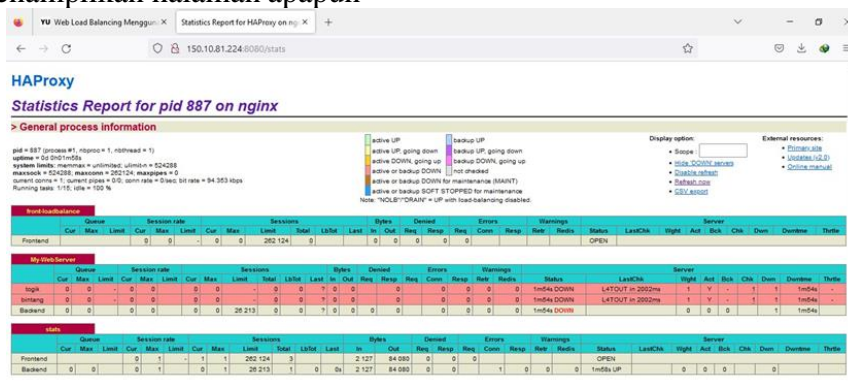
Gambar 25. Melihat Status pada HAProxy

- Melakukan pengujian jika salah satu server ada yang down/mati. disini dilakukan pengujian jika server “togik” mengalami down maka pada stats akan berwarna merah yang mendakan bahwa server tersebut sedang down Jika kita memasukan IP haproxy maka hanya akan menampilkan halaman web server 2



Gambar 26. Monitoring Status pada HAProxy

- Selanjutnya pengujian ketika kedua server mati/down maka pada stats akan menunjukkan kedua server berwarna merah yang artinya kedua server tidak akan menampilkan halaman apapun



Gambar 27. Monitoring Status pada HAProxy

## 5. KESIMPULAN

Kesimpulan yang dapat penulis simpulkan dari penelitian yang berjudul Instalasi Serta Konfigurasi HAProxy Sebagai Load Balancing Web Server ialah secara umum, daya tanggap setiap server bergantung pada spesifikasi server serta throughput jaringan. Kedua faktor ini

saling bergantung serta tidak dapat dipisahkan. Jika client memiliki jaringan yang cepat namun spesifikasi server yang digunakan rendah maka respon server terhadap client akan kurang tanggap. Meskipun server memiliki spesifikasi tinggi serta dipasang di jaringan yang lambat, respon client-server juga kurang. Untuk mencapai kinerja yang baik, kualitas jaringan serta kualitas server diuji sesuai. Penyeimbang beban berhasil menyelesaikan tugasnya mengambil file dari server. Selama upaya browser pertama klien, penyeimbang beban mengambil dari server 1, saat halaman browser klien disegarkan, penyeimbang beban mengirim file dari server lain. Penyeimbang muatan telah berhasil menyelesaikan tugasnya, yang terus mengirimkan file yang sama dari server yang sama jika server tersedia. Jika server yang melayani klien tidak tersedia, penyeimbang beban mengambil file dari server lain yang tersedia. Penerapan load balancing ini dapat diterapkan untuk mendukung kebutuhan jaringan kecil seperti sekolah, perkantoran, rumah sakit, kampus serta lain-lain..

#### DAFTAR PUSTAKA

- [1] G. H. Prathama, D. Andaresta, and K. Darmaastawan, "Instalasi Framework IoT Berbasis Platform Thingsboard di Ubuntu Server," *TIERS Information Technology Journal*, vol. 2, no. 2, pp. 1–9, Dec. 2021, doi: 10.38043/tiers.v2i2.3329.
- [2] A. M. Komaruddin, D. M. Sipitorini, and P. Rispihan, "LOAD BALANCING DENGAN METODE ROUND ROBIN UNTUK PEMBAGIAN BEBAN KERJA WEB SERVER," *Jurnal Siliwangi*, vol. 5, no. 2, 2019.
- [3] A. Nurrohman, A. Ramdan, and J. Informatika, "Implementasi Load Balancing dengan Nginx Menggunakan Algoritma Least Connection dengan Replikasi MySQL XXX-X-XXXX-XXXX-X/XX/\$XX.00 ©20XX IEEE Implementasi Load Balancing dengan Nginx Menggunakan Algoritma Least Connection dengan Replikasi MySQL." [Online]. Available: <https://www.researchgate.net/publication/338187063>
- [4] B. Arifwidodo, V. Metayasha, and S. Ikhwan, "Analisis Kinerja Load Balancing pada Server Web Menggunakan Algoritma Weighted Round Robin pada Proxmox VE," *Jurnal Telekomunikasi serta Komputer*, vol. 11, no. 3, p. 210, Dec. 2021, doi: 10.22441/incomtech.v11i3.11775.
- [5] F. al Isfahani and F. Nugraha, "Implementasi Load Balancing NGINX serta MongoDB Cluster serta Mekanisme Redis Caching Sistem Terdistribusi View project," 2019. [Online]. Available: <https://www.researchgate.net/publication/338126607>
- [6] H. Satria Harefa, J. Triyono, S. Raharjo, J. Kalisahak No, and K. Balapan, "Implementasi Load Balancing Web Server Untuk Optimalisasi Kinerja Web Server Serta Database Server," 2021.
- [7] H. Nurdin and F. Wuryo Handono, "Perancangan N-Clustering High Availability Web Server Dengan Load Balancing Serta Failover", [Online]. Available: <http://nusamandiri.ac.idhttp://nusamandiri.ac.idhttp://bsi.ac.id>
- [8] S. Shidqi, D. A. Sulistyono, and F. Almu'iini Ahda, "Pembuatan Infrastruktur Database Menggunakan Metode Replikasi Untuk Pelanggan Jagoan Hosting," *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 16, no. 1, 2022.
- [9] K. A. Pratama, R. T. Subagio, M. Hatta, and V. Asih, "178~189 Implementasi Load Balancing Pada Web Server Menggunakan Apache Dengan Server Mirror Data Secara Real Time-(Kresna Adi Pratama," 2021.

- [10] P. Studi Teknik Informatika, P. Natalia Evianti, A. Mulyana Wihandar, and A. Kurniawan, "Automation Provisioning Dev-Ops Website Server Menggunakan Ansible Serta Vagrant," vol. 2, no. 2, pp. 72–91, 2021.
- [11] A. S. serta Wini M K, A. Setiawan, and W. Muthia Kansha, "Development of Cluster Database System Using Galera Cluster Application at Vocational School of IPB University," vol. 11, no. 2, pp. 49–59, 2021, doi: 10.29244/jstsv.11.2.49.
- [12] M. Naufal, A. Rizqi, I. Kadek, and D. Nuryana, "Analisis Perbandingan Kinerja Algoritma Weighted Round Robin serta Weighted Least Connection Menggunakan Load Balancing Nginx Pada Virtual Private Server (VPS)," *Journal of Informatics and Computer Science*, vol. 04, 2022.
- [13] G. P. Sajati and B. T. Handoko, "Implementasi Sistem Terdistribusi Load Balancing HAProxy serta Replikasi MySQL Implementasi Sistem Terdistribusi Load Balancing HAProxy serta Replikasi MySQL View project Implementasi Algoritma RSA serta kriptografi Quantum pada Sistem Login View project." [Online]. Available: <https://www.researchgate.net/publication/338138607>
- [14] H. Alamsyah, "Analisa Serta Perancangan Load Balancing Web Server Menggunakan HAProxy Analysis And Design Of Web Server Load Balancing Using HAProxy," 2021.
- [15] A. R. Sofyan, S. Dwi, and Y. Kusuma, "Implementasi Load Balancing Web Server menggunakan HAProxy pada Virtual Server Direktorat SMK Kemendikbudristek".
- [16] K. Wibowo, I. Fitri, and D. Hidayatullah, "Implementasi Load Balancing Web Server Menggunakan Apache di Ubuntu 16.04.," *SISFOTENIKA*, vol. 10, no. 1, p. 50, Jan. 2020, doi: 10.30700/jst.v10i1.773.
- [17] E. P. Cynthia *et al.*, "Rancang Bangun Server HAProxy Load Balancing Master to Master MySQL (Replication) Berbasis Cloud Computing."
- [18] R. A. Putra and H. F. Lutfi, "Implementasi Load Balancing Algoritma Round Robin Pada F5 Big-IP Local Traffic Manager." [Online]. Available: <https://www.researchgate.net/publication/338187273>
- [19] M. Kusumawardani, N. Suharto, and M. Nanak Zakaria, "Pembagian Beban Trafik pada Cluster Server," 2022. [Online]. Available: <https://jurnaleccis.ub.ac.id/>