

Analisis Perbandingan Kinerja Algoritma Weighted Round Robin dan Weighted Least Connection Menggunakan Load Balancing Nginx Pada Virtual Private Server (VPS)

Muhammad Naufal Ammar Rizqi¹, I Kadek Dwi Nuryana²

^{1,2} Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

¹muhammadnaufal.18049@unesa.ac.id

²dwinuryana@unesa.ac.id

Abstrak— Saat ini, kemajuan teknologi yang pesat memudahkan untuk mendapatkan informasi dan menyelesaikan pekerjaan dengan cepat. Salah satunya adalah penggunaan teknologi web. Oleh karena itu, peran web server sangat penting saat memproses permintaan dalam jumlah besar. Jenis penelitian ini merupakan penelitian eksperimental yang bertujuan untuk membandingkan performansi algoritma weighted round robin (WRR) dan algoritma weighted least connection (WLC) menggunakan nginx pada Virtual Private Server (VPS). Pada pengujian throughput rata-rata dari WLC adalah 150.9 thread/sec yang dimana hasil tersebut lebih unggul 38.31 thread/sec dari pada algoritma WRR. Untuk pengujian Response time menggunakan WLC rata-rata yang didapatkan 31.77 sec lebih cepat dibandingkan yang menggunakan WRR. Dan untuk pengujian Error Req untuk WRR mendapatkan rata-rata 39.66% lebih besar dibandingkan dengan WLC. Sedangkan untuk pengujian CPU dapat diambil kesimpulan bahwa konfigurasi bobot pada setiap server dan method mempengaruhi koneksi yang diterima setiap server. Dari keempat resource yang diuji membuktikan bahwa algoritma weighted least connection (WLC) lebih efisien dalam membagi traffic dikarenakan karakteristik dari algoritma weighted least connection (WLC) adalah membagi traffic tidak hanya dari bobot namun juga dari koneksi yang tersambung pada masing-masing node.

Kata Kunci— Load Balancing, Weighted Round Robin, Weighted Least Connection, VPS, Nginx.

I. PENDAHULUAN

Saat ini, kemajuan teknologi yang pesat memudahkan untuk mendapatkan informasi dan menyelesaikan pekerjaan dengan cepat. Salah satunya adalah penggunaan teknologi web. Banyak Institusi dan organisasi yang menggunakan teknologi jaringan. Aplikasi berbasis web dirancang dan dirancang untuk dapat diakses oleh banyak orang. Oleh karena itu, peran web server sangat penting saat memproses permintaan dalam jumlah besar.

Menurut hasil survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) tahun 2019-2020, pengguna internet di Indonesia telah mencapai 196,71 juta dari 266,91 juta penduduk Indonesia. Menurut hasil survei yang dilakukan Asosiasi Penyelenggara Jasa Internet Indonesia

(APJII) pada 2019-2020, dibandingkan hasil survei 2018, jumlah pengguna Internet meningkat 8,9% menjadi hanya 171,17 juta orang. (APJII, 2019) Pengguna Internet yang berkembang pesat dapat mempengaruhi beban lalu lintas. Seringkali, masalahnya adalah pengelola server tidak mengelola server web dengan benar. Akibatnya, server downtime dan overload sering terjadi karena akses simultan oleh sejumlah besar pengguna. Karena menggunakan satu server untuk menampung banyak permintaan. (Wibowo et al., 2020).

Web server menjadi bagian penting dari internet saat ini infrastruktur. Aplikasi internet terbesar seperti Google, YouTube, dan Gmail berjalan di atas aplikasi web. Itulah mengapa membangun infrastruktur web server yang andal dan highly available adalah sangat penting (Bella et al., 2018). Menurut MADCOMS (2016) Web Server adalah program komputer yang bertanggung jawab atau bertugas menerima permintaan HTTP dari komputer klien (disebut browser web) dan melayani mereka dengan memberikan tanggapan HTTP dalam bentuk konten data. Sehingga perlu digunakan sistem load balancing untuk mengatasi terjadinya server downtime. Ada berbagai macam web server seperti Nginx.

Nginx adalah server web ringan dengan kinerja cepat. Nginx dapat menangani banyak permintaan dengan baik. Nginx dapat diakses oleh banyak klien secara bersamaan, yang berarti server Nginx dapat menerima banyak lalu lintas dengan menggunakan Load Balancing Nginx. (Apriliansyah et al., 2020) Dari penjelasan diatas untuk mengatasi masalah dari terlalu banyaknya request sehingga dapat membebani web server dan dapat membuat server down. Maka sistem load balancing dapat digunakan untuk mengurangi beban pada web server.

Menurut dari hasil penelitian dari Rahmatulloh & MSN (2017) Sistem Load Balancing bekerja dengan baik ketika permintaan dari klien berhasil didistribusikan oleh penyeimbang ke setiap cluster node. Dengan cara ini server tidak kelebihan beban dan server web memiliki kapasitas untuk melayani 10.000 permintaan tanpa kesalahan permintaan. Dengan teknologi load balancing, manfaat seperti memastikan keandalan layanan, ketersediaan, dan skalabilitas jaringan dapat diperoleh. (Prasetyocloa & Hikmah, 2018).

Banyak algoritma yang dapat dipakai dalam Load Balancing seperti Weighted Round Robin (WRR) dan Weighted Least Connection (WLC) yang akan digunakan dalam pengujian ini. Weighted Round Robin (WRR) adalah algoritma yang diturunkan dari round robin menggunakan bobot untuk

merepresentasikan tingkat kemampuan layanan mesin dan melakukan penjadwalan sesuai dengan bobot, untuk memastikan bahwa server berkinerja tinggi dapat memperoleh tingkat penggunaan yang tinggi. Ini, sampai batas tertentu, menghindari kelebihan relatif dari server berkinerja rendah dan meningkatkan tingkat penggunaan server berkinerja tinggi.(Yang et al., 2018). Ada beberapa penelitian yang dilakukan menggunakan algoritma Weighted Round Robin(WRR) untuk balancing antara lain : Analisis Performa Load Balancing Algoritma Weighted Round Robin di Infrastruktur BPBD Provinsi Jawa Timur(Pradana & Prapanca, 2020) dan Implementasi Load Balancing Dengan Algoritma Penjadwalan Weighted Round Robin Dalam Mengatasi Beban Webserver(Hanafiah, 2021).

Hampir sama dengan Weighted Round Robin(WRR). Algoritma weighted least connection(WLC) membagi beban lalu lintas sesuai dengan bobot kinerja masing-masing server. Nilai bobot memiliki dampak besar pada jumlah koneksi yang dapat ditampung server. Server dengan spek yang unggul diberikan nilai bobot yang lebih besar sehingga dapat menampung lebih banyak koneksi.(Rahmana et al., 2018) Untuk Algoritma Weighted Least Connection (WLC) ada juga beberapa penelitian yang menggunakannya sebagai algoritma load balancing antara lain : Analisis Load Balancing Pada Web Server Menggunakan Algoritme Weighted Least Connection(Rahmana et al., 2018) dan Research and optimization of the cluster server load balancing technology based on centos 7(Ding et al., 2018).

Dan untuk Virtual Private Server sendiri adalah virtual server yang dibagi dari server fisik. Virtual Private Server (VPS) memungkinkan akses pengguna ke sistem operasi dari sumber daya perangkat lunak dan perangkat keras pribadi. Banyak berbagai faktor dapat mempengaruhi kinerja VPS terutama termasuk spesifikasi perangkat keras server fisik, sistem operasi yang dipilih, lapisan virtualisasi, dan infrastruktur jaringan. Oleh karena itu, sangat penting untuk pilih host VPS dengan benar yang memenuhi pengguna, aplikasi dan persyaratan sumber daya dan kinerja layanan. Virtual Private Server (VPS) menawarkan biaya yang hemat dengan cara menyediakan spesifikasi yang kuat dan fleksibel server. Dari merender video, hingga menghosting server game dan situs web, VPS memiliki beragam kemungkinan penggunaan dan digunakan oleh banyak pengguna yang berbeda untuk memenuhi pemrosesan pribadi dan profesional yang berkembang, komputasi, dan persyaratan dan kebutuhan penyimpanan. (Hariyadi et al., 2020).

Dari penelitian penelitian diatas terdapat beberapa penelitian yang menggunakan Algoritma Weighted Round Robin(WRR) sebagai bahan penelitiannya dan terdapat juga penelitian yang menggunakan Algoritma Weighted Least Connection(WLC) untuk penelitian. Oleh sebab itu penelitian ini dilakukan agar dapat mengetahui Algoritma Load Balancing mana yang terbaik dalam hal mengatasi beban kerja yang sangat besar. Dalam penelitian akan menggunakan metode stress test yang dimana masing masing load balancing akan di test dengan request 5000 , 10000 , 15000 , 20000 dengan masing masing menggunakan 3 buah server dan dalam test ini

request akan dilakukan pada cloud storage untuk mengakses foto , video , dan pdf. Berbeda dengan penelitian sebelumnya, penelitian ini menggunakan load balancer Nginx di VPS untuk melakukan uji perbandingan kinerja antara algoritma weighted round robin(WRR) dengan algoritma weighted least connection(WLC). Di sini, penyimpanan cloud menggunakan gambar, video, dan file untuk menafsirkan sebuah web yang terdapat gambar , video dan file.

II. METODE PENELITIAN

Jenis penelitian ini adalah penelitian eksperimental yang bertujuan untuk membandingkan kinerja algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC) menggunakan nginx pada VPS. Metode adalah langkah-langkah yang dikembangkan dan diterapkan untuk mencapai tujuan tertentu. Ada beberapa fase yang terlibat dalam desain server dan pengembangan aplikasi. antara lain:



Gbr 1. Metode penelitian

A. Analisa Kebutuhan

Tahap pertama penelitian adalah analisis kebutuhan yang akan digunakan untuk percobaan. Persyaratan ini akan digunakan sebagai modalitas untuk menyelesaikan analisis komparatif dari load balancing yang menggunakan algoritma weighted round robin(WRR) dan algoritma weighted least

connection(WLC) dan single server(SS). Analisis kebutuhan dapat dibagi menjadi beberapa bagian, yaitu:

1. Kebutuhan Data

Data yang dibutuhkan untuk penelitian ini diperoleh dari berbagai referensi. Referensi berasal dari Jurnal internasional dan Nasional. Dengan menggunakan data yang lengkap dan akurat, hasil penelitian ini dapat lengkap dan relevan. Pengumpulan data untuk penelitian ini berasal dari dua sumber yaitu studi pustaka dan observasi.

a. Studi literatur

Dalam penelitian ini, kami mengumpulkan berbagai referensi dari beberapa literatur yang mendukung analisis perbandingan dari algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC). Untuk mempelajari lebih lanjut, literatur diambil dari sumber Internet seperti laporan, jurnal nasional, jurnal internasional, makalah, tesis, situs web resmi, dll.

b. Observasi

Penelitian ini juga melakukan observasi dengan mengunjungi beberapa situs referensi Load Balancing dan juga situs resmi dokumentasi nginx dan situs resmi dokumentasi jmeter yang mana nantinya akan digunakan sebagai alat pengujian.

2. Kebutuhan Alat

Spesifikasi perangkat yang digunakan untuk melakukan penelitian yang berupa perancangan dan analisis adalah :

- Processor AMD A8-5550M
 - RAM 8 GB
 - SSD 512 GB
 - Sistem Operasi Windows 8.1 Pro 64-bit
- Adapun perangkat lunak yang digunakan untuk mendukung penelitian ini adalah :
- Putty SSH untuk mengakses VPS
 - JMeter Apache untuk pengujian
 - Visual Studio Code memiliki fungsi sebagai alat untuk menulis kode program

B. Perancangan Sistem

Perancangan sistem adalah tahap mempersiapkan kebutuhan sistem untuk melakukan implementasi penelitian. Yaitu system single server(SS) , load balancer (LB) , vps :

1. Perancangan Single Server(SS)

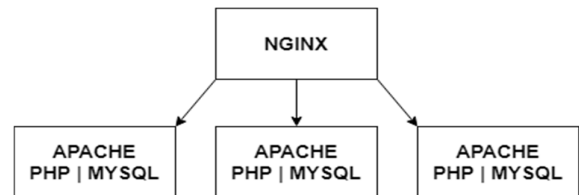


Gbr 2. Perancangan Single Server

Pada gambar 2 adalah gambar yang menunjukkan arsitektur yang akan digunakan pada single server(SS) dan digunakan pada server yang berfungsi sebagai node dari load balancer. Dalam single server (SS) maupun node Web Server yang digunakan merupakan Apache dengan script PHP dan database mysql pada root folder Apache 'var/www/html/cloud' untuk menampilkan web cloud storage yang dijalankan pada OS Ubuntu dalam virtual private server.

2. Perancangan Load Balancer

Berikut ini perancangan load balancer yang akan digunakan untuk analisis perbandingan algoritma weighted round robin(WRR) dan algoritma weighted least connection:



Gbr 3. Desain Load Balancer

Pada gambar 3 terlihat Web server nginx sebagai load balancer yang berfungsi membagi traffic kepada node node yang dibawahnya.

3. Perancangan VPS

Pada penelitian untuk uji coba nya menggunakan VPS(Virtual Private Server) yang dimana akan digunakan sebagai load balancer dan node 1 , node 2 , node 2. Pada tabel dibawah ini adalah spesifikasi spesifikasi dari VPS VPS yang akan digunakan nanti

Tabel 1 Spesifikasi Hardware

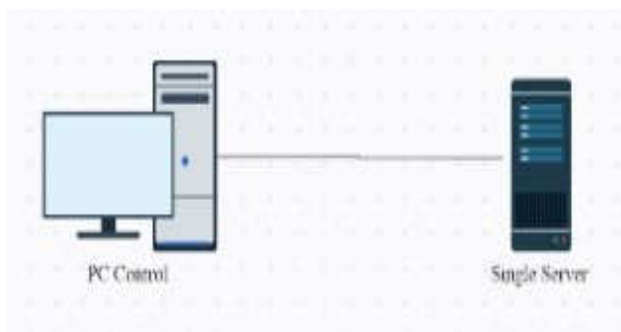
Load Balancer	
CPU	2 vCPU
Memory	4 GB
Storage	10 GB
Sistem Operasi	Linux Ubuntu
Aplikasi	NGINX
IP Address	35.239.176.8
Node 1	
CPU	2 vCPU
Memory	1 GB
Storage	10 GB
Sistem Operasi	Linux Ubuntu
Aplikasi	Apache Web Server, php ,mysql

IP Address	34.121.23.142
Node 2	
CPU	2 vCPU
Memory	2 GB
Storage	10 GB
Sistem Operasi	Linux Ubuntu
Aplikasi	Apache Web Server, php ,mysql
IP Address	104.198.224.173
Node 3	
CPU	2 vCPU
Memory	4 GB
Storage	10 GB
Sistem Operasi	Linux Ubuntu
Aplikasi	Apache Web Server, php ,mysql
IP Address	34.69.79.208

Pada tabel 1 terlihat bahwa spesifikasi dari load balancer dan node 1 , node 2 , node 3 berbeda ini dikarenakan pada algoritma Weighted Round Robin(WRR) dan algoritma Weighted Least Connection(WLC) membagi traffic menggunakan bobot yang dimana bobot ini dapat disesuaikan berdasarkan spesifikasi dari server tersebut. Maka dari itu jika server node tersebut lebih memiliki spesifikasi yang lebih bagus dapat diberi bobot lebih besar, agar server tersebut mendapatkan traffic lebih banyak daripada server lainnya.

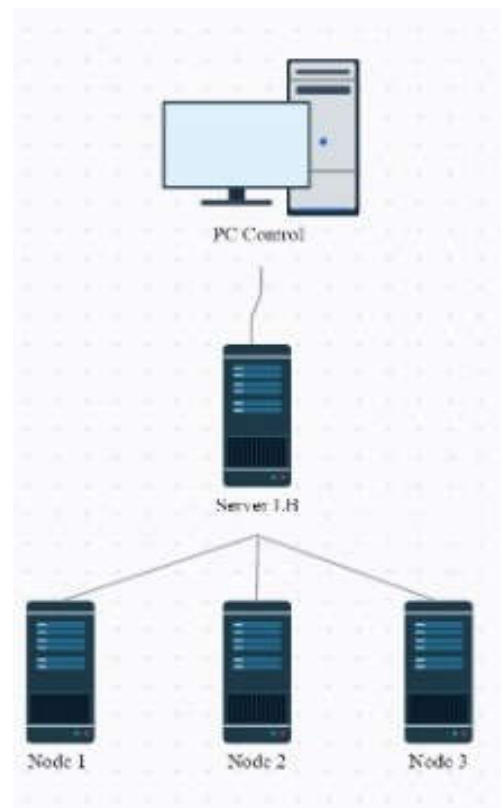
C. Perancangan Topologi Jaringan

Perancangan topologi jaringan digunakan untuk memudahkan perancangan server yang tanpa menggunakan Load Balancing dan juga menggunakan Load Balancing sebagai objek pengujian.



Gbr 3. Perancangan Topologi Single Server

Pada gambar 3. PC Control melakukan pengujian dengan tool jmeter terhadap Web Server yang dijalankan pada VPS dengan IP 34.121.23.142.



Gbr 4. Perancangan Topologi Load Balancer

Pada Gambar 4. PC User melakukan pengujian dengan tool Jmeter terhadap server Load balancer dengan IP 35.239.176.8 yang dimana nantinya server load balancer akan membagi traffic kepada node 1 , node 2 , node 3 sesuai bobot yang telah ditentukan.

D. Implementasi Sistem

Setelah menganalisis data dan merancang sistem. Langkah selanjutnya adalah menerapkan rancangan dan hasil desain yang dibuat seperti mendeploy web ke single server (SS) maupun node node yang telah disiapkan ,install web server nginx dan apache, install php dan mysql dan konfigurasi nginx dengan algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC) untuk menjalankan uji benchmark.

E. Pengujian Sistem

Pada tahap ini, setelah tahap implementasi selesai kemudian dilakukan pengujian. Pengujian ini dilakukan untuk mendapatkan hasil dari pengujian tersebut, yang dimana nantinya data dari hasil pengujian tersebut akan di bandingkan agar mengetahui algoritma mana yang lebih baik antara algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC). Dari studi literatur yang dilakukan pada penelitian penelitian terdahulu tentang algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC) didapatkan cara kerja dari algoritma weighted round robin(WRR) dan

algoritma weighted least connection(WLC). Untuk algoritma weighted round robin(WRR) sendiri cara kerjanya adalah dengan membagi traffic yang datang dibagi hanya sesuai bobot jadi jika traffic yang datang adalah 10 request dan pembagian bobot adalah 2:3:5 maka node 1 akan mendapatkan 2 traffic , node 2 akan mendapatkan 3 traffic dan node 3 akan mendapatkan 5 traffic. Sedangkan algoritma weighted least connection(WLC) pembagian traffiknya tidak hanya dari bobot namun juga melihat dari koneksi jadi jika traffic yang datang adalah 10 dan pembagian bobot adalah 2:3:5 maka sebelum membagi traffic lb akan mengecek setiap node terlebih dahulu apakah koneksi yang terdapat pada node itu sudah memenuhi bobot atau belum jadi jika pada node 1 telah terpenuhi 2 request dan 1 request telah tidak terkoneksi maka node 1 akan diisi terlebih dahulu baru node lainnya berbeda dengan algoritma weighted round robin (WRR) yang tidak melakukan pengecekan terlebih dahulu dan melakukan pembagian secara rata sesuai dengan bobotnya.

F. Analisis Hasil Pengujian

Analisis hasil tes merupakan akhir dari penelitian. Dari percobaan load balancing dapat dilihat perbedaan performa dari setiap optimasi load balancing menggunakan algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC). Setelah itu akan didapatkan nilai parameter yaitu :

1. Throughput

Kebutuhan resource throughput yang diukur adalah jumlah data yang diterima dalam satuan waktu dari setiap pengujian.

2. Response Time

Kebutuhan resource response time yang diukur adalah waktu yang dibutuhkan untuk menyelesaikan satu request dan mengirimkannya kembali kepada client dari setiap pengujian

3. Error request

Kebutuhan resource error request yang diukur adalah sejumlah request yang mengalami error pada saat penelitian dari setiap pengujian.

4. CPU

Kebutuhan resource CPU yang diukur adalah penggunaan CPU dari setiap pengujian sistem web server Apache yang sedang berjalan. Data yang diambil dari statistik penggunaan sumber daya web server yang dilakukan selama pengujian.

Parameter tersebut kemudian digunakan sebagai data untuk membandingkan kinerja masing-masing load balancer.

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan diambil setelah perancangan system dan Analisa kebutuhan selesai diimplementasikan. Hal yang akan dibahas pada hasil dan pembahasan adalah Analisis Perbandingan Kinerja Algoritma Weighted Round Robin(WRR) dan Weighted Least Connection(WLC) Menggunakan Load Balancing Nginx Pada VPS.

A. Deployment Cloud Storage pada VPS

Setelah membuat VPS dengan spesifikasi yang ditentukan tahap selanjutnya ada mendeploy web cloud storage pada server tersebut. Untuk dapat mendeploy web cloud storage harus menginstal PHP , Apache sebagai web server dan Mysql sebagai database servernya.

```
mnaufalamarr_15@node-1:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease [94.5 kB]
```

Gbr 5. Update Packet

Sebelum melakukan instalasi web server apache update terlebih dulu packet yang di server dengan cara seperti yang ada di gambar 5.

```
mnaufalamarr_15@node-1:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libatasmart4 libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-part2 libblockdev-swap2
  libblockdev-util2 libblockdev2 libbtrfs-glib4 libbtrfs-proxy libbtrfs-glib0 libnspr4 libnss3 libnuma1
  libparted-fs-resize0 libqemu-glib5 libqemu-proxy librados2-0 usb-modeswitch usb-modeswitch-data
Use 'dpkg --get-selections --purge' to remove these packages.
0 upgraded, 1 installed, 0 removed, 0 not installed, 0 not upgraded.
```

Gbr 6. Instalasi Apache2

Setelah update packet tahap selanjutnya adalah menginstal apache web server dengan cara menuliskan perintah yang ada di gambar 6. Setelah server menginstall apache web server tahap selanjutnya adalah mengijinkan firewall apache seperti pada gambar 7.

```
mnaufalamarr_15@node-1:~$ sudo ufw allow apache
Rules updated
Rules updated (v6)
```

Gbr 7. Ijinkan Firewall

Setelah mengijinkan firewall akses ip maka akan memunculkan page awal dari apache untuk menggantinya agar menampilkan web cloud storage ubah DocumentRoot yang ada pada '/etc/apache2/sites-available/000-default.conf'.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/cloud1/
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Gbr 8. Konfigurasi 000-default.conf

Konfigurasi apache dilakukan untuk mengkonfigurasi DocumentRoot apache. Dalam konfigurasi '000-default.conf' menampilkan server admin , DocumentRoot yang merujuk pada '/var/www/html/cloud' yang berisi file web cloud storage. 'ErrorLog' berfungsi untuk menyimpan error log pada apache.

Setelah mengubah DocumentRoot agar dapat menampilkan web tersebut pada browser install php terlebih dahulu.

```
mnaufalamarr 15@node-1:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-part2 libblockdev-swap2
libblockdev-util2 libblockdev2 libblock-glib4 libblock-proxy libm-glib2 libnfs4 libnfs3
libparted-fs-resize2 libpart-glib2 libpart-proxy libpart2-0 sub-mode-switch sub-mode-switch-data
Use 'dpkg --get-selections' to see the status of installed packages.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

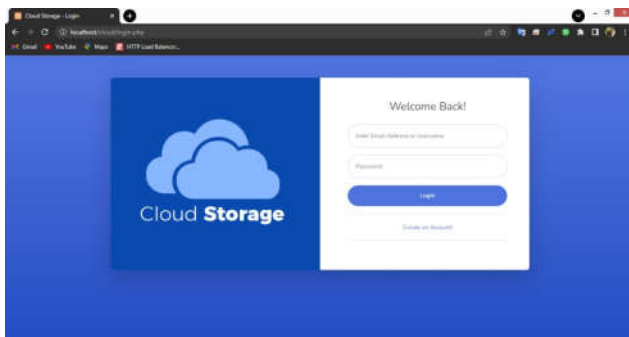
Gbr 9. Konfigurasi 000-default.conf

Setelah install php buka browser dan masukan ip maka web cloud storage akan tampil di browser namun agar web dapat berjalan dengan semestinya mysql harus terinstall di server.

```
mnaufalamarr 15@node-1:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-part2 libblockdev-swap2
libblockdev-util2 libblockdev2 libblock-glib4 libblock-proxy libm-glib2 libnfs4 libnfs3
libparted-fs-resize2 libpart-glib2 libpart-proxy libpart2-0 sub-mode-switch sub-mode-switch-data
Use 'dpkg --get-selections' to see the status of installed packages.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Gbr 10. Instalasi Mysql

Setelah menginstall mysql web akan dapat berjalan dengan semestinya karena telah terdapat database servernya. Jadi dapat membuat database, mengimport database, membuat table, memasukkan data, dll



Gbr 11. Tampilan Web Cloud Storage

B. Instalasi Load Balancer pada VPS

Untuk membuat server agar menjadi load balancer server harus terinstall beberapa software seperti nginx untuk menjadi load balancernya. Untuk melakukan instalasi nginx pertama update packetnya terlebih dahulu.

```
mnaufalamarr 15@node-1:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
```

Gbr 12. Update Packet

Setelah melakukan update packet yang ada di server tahap selanjutnya melakukan instalasi nginx web server agar dapat melakukan pemecahan traffic ke beberapa server.

```
mnaufalamarr 15@load-balancer:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
nginx is already the newest version (1:1.18.0-ubuntu1.3).
The following packages were automatically installed and are no longer required:
libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-part2 libblockdev-swap2
libblockdev-util2 libblockdev2 libblock-glib4 libblock-proxy libm-glib2 libnfs4 libnfs3
libparted-fs-resize2 libpart-glib2 libpart-proxy libpart2-0 sub-mode-switch sub-mode-switch-data
Use 'dpkg --get-selections' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Gbr 13. Instalasi Nginx

Setelah melakukan instalasi nginx sebelum melakukan konfigurasi terlebih dahulu mengijinkan firewall nginx dan mengaktifkan firewallnya.

```
mnaufalamarr 15@load-balancer:~$ sudo ufw allow "Nginx Full"
Rule updated
Rule updated (v6)
```

Gbr 13. Instalasi Nginx

Tahap selanjutnya adalah mengedit konfigurasi untuk mengatur load balance agar dapat memecah traffic ke IP yang telah ditentukan. Buka `/etc/nginx/sites-available/default` dan edit menjadi seperti gambar 14.

```
upstream backend {
    least_conn;
    server 34.121.23.142 weight=2;
    server 104.198.224.173 weight=3;
    server 34.69.79.208 weight=5;
}

server {
    location / {
        proxy_pass http://backend;
    }
}
```

Gbr 14. Konfigurasi default.conf

Pada gambar 14 berfungsi untuk membuat group upstream dengan nama backend, group upstream tersebut berisi 3 server dengan IP yang telah ditentukan dan 3 server yang telah ditentukan memiliki bobot yang berbeda-beda sesuai dengan spesifikasi yang telah dibuat. Server 1 memiliki bobot terkecil yaitu 2 dikarenakan server 1 memiliki spesifikasi yang paling rendah. Dan dikarenakan server 2 memiliki spesifikasi menengah dengan 2 vCPU dan 2 GB memory maka server 2 memiliki bobot yang lebih berat dibandingkan dengan server 1. Sedangkan server 3 yang memiliki spesifikasi yang paling tinggi dia mendapatkan bobot paling tinggi juga yaitu 3.

Pada menu upstream selain dapat menentukan ip tujuan dari server node terdapat pilihan lanjutan yang dapat diatur seperti pengaturan method load balancing dan pengaturan weight. Untuk method load balancing sendiri terdapat banyak method namun pada pengujian kali ini hanya menggunakan Round Robin dan Least Connection yang dimana dikombinasikan dengan weight maka jadi lah algoritma weighted round robin(WRR) dan weighted least connection(WLC).

Sedangkan pada menu server terdapat menu lagi yaitu location yang berisi proxy_pass <http://backend>; Proxy_pass <http://backend>; ini berguna untuk memungkinkan permintaan yang datang melalui proxy reverse Nginx untuk meneruskan ke group upstream yang telah dibuat yaitu backend.

C. Hasil Pengujian Throughput

Penggunaan kebutuhan throughput yang diambil dengan menjalankan beberapa penelitian seperti single server (SS), Weighted Round Robin (WRR) dan Weighted Least Connection(WLC). Data yang diambil dari pengujian menggunakan JMeter.

Untuk hasil pengujian penggunaan throughput pada penelitian ini dapat dilihat pada table 2.

Tabel 2. Hasil Pengujian Throughput (Sec)

Jumlah HTTP Request		5000	10000	15000	20000
PHP & MYSQL	SS	11.1	20	83.2	56.1
	WRR	13.9	22.2	35.19	41.3
	WLC	22.2	35	41.4	52.3
PHP & MYSQL & GAMBAR	SS	34	41	56.4	61
	WRR	14.1	26.1	41.1	56.9
	WLC	20.1	36.8	43.8	57.8
PHP & MYSQL & VIDEO	SS	30	35.7	58.2	60.7
	WRR	15.9	32.2	54.4	61.2
	WLC	26.1	39.1	63	68.7
PHP & MYSQL & FILE	SS	21.2	43.27	47	38.2
	WRR	40	47.9	52.3	65.8
	WLC	41.16	51.4	54.4	67.6

Pada tabel hasil pengujian penggunaan kebutuhan throughput menunjukkan bahwa throughput yang dihasilkan dari menggunakan algoritma weighted least connection(WLC) menunjukkan performa throughput yang lebih baik jika dilihat pada tabel dibandingkan dengan algoritma weighted round robin(WRR). Jika pada pengujian php & mysql dapat dilihat rata – rata dari pengujian WLC lebih unggul dengan 38.31 thread/sec yang dimana pada pengujian throughput yang lebih besar lah yang lebih baik.

D. Hasil Pengujian Response Time

Pengujian response time yang diambil dengan menjalankan beberapa penelitian seperti single server (SS) , Weighted Round Robin (WRR) dan Weighted Least Connection(WLC). Data yang diambil dari pengujian menggunakan JMeter.

Untuk hasil pengujian response time pada penelitian ini dapat dilihat pada table 3.

Tabel 3. Hasil Pengujian Response Time(Sec)

Jumlah HTTP Request		5000	10000	15000	20000
PHP & MYSQL	SS	22.6	23	26.8	28
	WRR	23.6	63.7	77.36	82.81
	WLC	18.8	59	61.9	72.6
PHP & MYSQL & GAMBAR	SS	23.6	34.7	36	43.4
	WRR	40	52.4	73.75	81
	WLC	34.45	43.9	53.25	61.4
PHP & MYSQL & VIDEO	SS	26.25	38.9	44.37	48.2
	WRR	50	62.9	71.6	77.48
	WLC	37.2	54.21	64.5	74.3
PHP & MYSQL & FILE	SS	21.4	30	39.7	44
	WRR	49	56	66.7	75.2
	WLC	40.2	48.2	58.7	68.9

Pada tabel hasil pengujian penggunaan kebutuhan response time menunjukkan bahwa response yang dihasilkan dari menggunakan algoritma weighted least connection(WLC) menunjukkan performa dari algoritma weighted least connection(WLC) yang lebih cepat jika dilihat pada tabel

dibandingkan dengan algoritma weighted round robin(WRR). Jika dari table pada pengujian video, WRR mendapatkan rata – rata 261.98 sec sedangkan WLC mendapatkan rata – rata 230.21 dari sini dapat dilihat bahwa rata – rata yang menggunakan WLC lebih 31.77 sec. WLC memiliki hasil response time lebih cepat karena request yang datang dialihkan node yang memiliki koneksi tersedikit.

E. Hasil Pengujian Error Request

Pengujian error request yang diambil dengan menjalankan beberapa penelitian seperti single server (SS) , Weighted Round Robin (WRR) dan Weighted Least Connection(WLC). Data yang diambil dari pengujian menggunakan JMeter.

Untuk hasil pengujian penggunaan error request pada penelitian ini dapat dilihat pada table 4.

Tabel 4. Hasil Pengujian error request (%)

Jumlah HTTP Request		5000	10000	15000	20000
PHP & MYSQL	SS	25.5	52.59	57.63	66.78
	WRR	14.48	30.3	47.28	57.7
	WLC	9.34	22.71	41.4	56.9
PHP & MYSQL & GAMBAR	SS	34.14	48.8	62.94	70
	WRR	27.5	38.24	58.91	64.86
	WLC	20.7	28.5	47.9	52.75
PHP & MYSQL & VIDEO	SS	37.7	59.3	61.57	72.13
	WRR	29.6	43.1	55.19	59.41
	WLC	20.88	34.55	51.7	56.69
PHP & MYSQL & FILE	SS	44.5	54.23	57.82	70.38
	WRR	15.4	36	54.81	64.82
	WLC	11.66	30.59	51.4	64

Pada tabel hasil pengujian kebutuhan error request menunjukkan bahwa error yang dihasilkan dari menggunakan algoritma weighted least connection(WLC) menunjukkan performa dari algoritma weighted least connection yang lebih sedikit errornya jika dilihat pada tabel dibandingkan dengan algoritma weighted round robin(WRR). Jika dari table pada pengujian gambar dapat dilihat WRR memiliki error request yang cukup besar yang dimana memiliki rata – rata 189.51% jika dibandingkan dengan yang menggunakan WLC yang hanya memiliki rata -rata 149.85%, disini dapat dilihat antara kedua algoritma tersebut memiliki selisih rata – rata yang cukup besar yaitu 39.66% yang berarti algoritma WLC lebih baik dalam membagi request yang datang sehingga dapat meminimalisir error yang dapat terjadi.

F. Hasil Pengujian Penggunaan CPU

Penggunaan kebutuhan CPU yang diambil dengan menjalankan beberapa penelitian seperti single server (SS) , Weighted Round Robin (WRR) dan Weighted Least Connection(WLC). Data yang diambil dari pengujian menggunakan Console yang telah disediakan oleh Google Cloud Platform.

Untuk hasil pengujian penggunaan CPU pada penelitian ini dapat dilihat pada table 5.

Tabel 5. Hasil Penggunaan CPU(%)

Nama CPU		CPU 1	CPU 2	CPU 3	CPU LB
PHP & MYSQL	SS	49.6			
	WRR	10.83	6.84	3.99	5.15
	WLC	13.87	7.24	4.2	6.23
PHP & MYSQL & GAMBAR	SS	57.5			
	WRR	13.59	7.76	4.32	6.04
	WLC	14.15	7.61	2.78	4.71
PHP & MYSQL & VIDEO	SS	61.14			
	WRR	11.47	6.88	4.63	6.08
	WLC	13.51	6.73	2.97	4.48
PHP & MYSQL & FILE	SS	46.6			
	WRR	9.8	3.9	4.6	6.3
	WLC	6.03	7.07	3.71	5.21

Pada tabel hasil pengujian penggunaan kebutuhan CPU menunjukkan bahwa CPU untuk dua algoritma sama-sama sesuai dengan bobot yang telah diimplementasikan melalui konfigurasi nginx, pada pengaturan konfigurasi nginx bobot dibagi menjadi 2:3:5. Dikarenakan node 1 memiliki lebih kecil spesifikasi dari yang lain maka pada table pengujian terlihat bahwa cpu dari node 1 lebih besar.

IV. KESIMPULAN

Berdasarkan dari hasil penelitian yang telah dilakukan dapat disimpulkan bahwa algoritma weighted round robin(WRR) dan algoritma weighted least connection(WLC) telah berhasil diimplementasikan pada web server nginx pada Virtual Private Server(VPS). dan analisis dari kedua algoritma tersebut berhasil dilakukan. Hasil analisis yang telah didapat dari pengujian kali ini adalah algoritma weighted least connection (WLC) lebih unggul dari pada algoritma weighted round robin(WRR) baik pada pengujian throughput, response time, maupun error request. Pada pengujian throughput rata – rata dari WLC adalah 150.9 thread/sec yang dimana rata – rata tersebut lebih unggul 38.31 thread/sec dari pada algoritma WRR. Untuk pengujian Response time menggunakan WLC hasil rata – rata yang didapatkan 31.77 sec lebih cepat dibandingkan rata – rata yang menggunakan WRR. Dan untuk pengujian Error Req untuk WRR sendiri mendapatkan rata – rata 39.66% lebih besar dibandingkan dengan yang menggunakan WLC. Sedangkan untuk pengujian CPU dapat diambil kesimpulan bahwa konfigurasi bobot pada setiap server dan method mempengaruhi koneksi yang diterima setiap server. Dari keempat resource yang diuji membuktikan bahwa algoritma weighted least connection(WLC) lebih efisien dalam membagi traffic dikarenakan karakteristik dari algoritma weighted least connection(WLC) adalah membagi traffic tidak hanya dari bobot namun juga dari koneksi yang tersambung pada masing masing node.

Saran yang diajukan untuk penelitian lebih lanjut, sebaiknya menggunakan Perangkat keras yang nyata dalam pengujian, karena dapat meningkatkan akurasi data uji. Untuk algoritme, analisis dapat dilakukan dengan menggunakan algoritme lain yang dapat memperhitungkan kapasitas atau spesifikasi perangkat dengan menetapkan bobot secara dinamis.

UCAPAN TERIMA KASIH

Rasa terima kasih dan ucapan rasa syukur peneliti sampaikan kepada :

1. Allah SWT yang Maha Esa atas segala berkah, rahmat dan hidayahnya, sehingga penulis dapat menyelesaikan penelitian ini.
2. Bapak Budi Sudiro dan Ibu Andayanti selaku orang tua saya yang selalu mendukung serta menyemangati.
3. Trianto, Andayani, Biyan, Dendro, Febriansyah, Athalla selaku keluarga yang selalu memberikan motivasi dan bimbingan.
4. Bapak I Kadek Dwi Nuryana selaku dosen pembimbing dengan sabar memberi arahan dan membantu penelitian ini dari awal hingga akhir.
5. Hanif A, Alfian Fahrudi, Farry C, Dimas S, Alfian M, Aris S, Romli, Fachriza, Adharis, Edo yang sering memberi wawasan dan refrensi untuk dapat menyelesaikan hambatan dalam penelitian.
6. Rizky Pramesti E, Anastasia A, Wanda W, Imam, Sahrul, Hafizh, Tegar, Dhoni, A Fadli dan teman-teman serta sahabat yang selalu memberikan energi positif.

REFERENSI

- Apriliansyah, F., Fitri, I., Iskandar, A., & Artikel, R. (2020). Jurnal Teknologi dan Manajemen Informatika Implementasi Load Balancing Pada Web Server Menggunakan Nginx Info Artikel ABSTRAK. *Tahun*, 6(1). <http://http://jurnal.unmer.ac.id/index.php/jtmi>
- Ding, L., Zheng, W., Liu, S., & Han, Z. (2018). Research and optimization of the cluster server load balancing technology based on centos 7. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10745 LNCS, 201–207. https://doi.org/10.1007/978-3-319-74521-3_23
- Hanafiah, A. (2021). Implementasi Load Balancing Dengan Algoritma Penjadwalan Weighted Round Robin Dalam Mengatasi Beban Webserver. *IT Journal Research and Development*, 5(2), 226–233. [https://doi.org/10.25299/itjrd.2021.vol5\(2\).5795](https://doi.org/10.25299/itjrd.2021.vol5(2).5795)
- Pradana, M. S., & Prapanca, A. (2020). Analisis Performa Load Balancing Algoritma Weighted Round Robin di Infrastruktur BPBD Provinsi Jawa Timur. *Journal of Informatics and Computer Science (JINACS)*, 1(02), 109–114. <https://doi.org/10.26740/jinacs.v1n02.p109-114>
- Prasetyo, T., & Hikmah, N. (2018). Implementasi Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Dengan Failover Berbasis Mikrotik Router (Studi Kasus PT. Sumber Rejeki Power). In *Seminar Nasional Inovasi dan Tren (SNIT)*.
- Rahmana, D., Primananda, R., & Yahya, W. (2018). Analisis Load Balancing Pada Web Server Menggunakan Algoritma Weighted Least Connection (Vol. 2, Issue 3). <http://j-ptiik.ub.ac.id>
- Rahmatulloh, A., & MSN, F. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 3(2), 241–248. <https://doi.org/10.25077/teknosi.v3i2.2017.241-248>
- Wibowo, K., Fitri, I., & Hidayatullah, D. (2020). Implementasi Load Balancing Web Server Menggunakan Apache di Ubuntu 16.04. *SISFOTENIKA*, 10(1), 50. <https://doi.org/10.30700/jst.v10i1.773>
- Yang, Z., Zhang, S., Ji, X., & Liu, X. (2018). Research on cloud service quality control implementation based on improved load balance algorithm. *Journal of Computational Methods in Sciences and Engineering*, 18(3), 793–800. <https://doi.org/10.3233/JCM-180830>

- Tim Madcoms. (2016). Pemrograman PHP dan MySQL untuk pemula / Tim Madcoms. Yogyakarta :: Andi Offset.
- Bella, M. R. M., Data, M., & Yahya, W. (2018). Web Server Load Balancing Based On Memory Utilization Using Docker Swarm. 2018 International Conference on Sustainable Information Engineering and Technology (SIET). doi:10.1109/siet.2018.8693212
- APJII, "Laporan Survei Internet APJII 2019 - 2020 [Q2],"2019. [Online]. Available: <https://apjii.or.id/survei>. [Accessed: 19-Mei-2022]
- S. Alotaibi, H. Alomair and M. Elhussein, "Comparing Performance of Commercial Cloud Storage Systems: The Case of Dropbox and One Drive," 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-5, doi: 10.1109/ICCISci.2019.8716385.
- Hariyadi, I. P., & Marzuki, K. (2020). Implementation Of Configuration Management Virtual Private Server Using Ansible. MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, 19(2), 347-357.