Jurnal Riset Inovasi Bidang Informatika Dan Pendidikan Informatika (KERNEL)
Vol. 1, No. 2, December 2020
ISSN: 2774-4345

IMPLEMENTASI *LOAD BALANCING* MENGGUNAKAN ANTRIAN *ROUND ROBIN* DENGAN STUDI KASUS *E-SHOP*

Danang Haryo Sulaksono[1], Arnold Reza Giovanni[2],
[1,2]Jurusan Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi
Institut Teknologi Adhi tama Surabaya
Email : danang_h_s@itats.ac.id, arnoldreza27@gmail.com

ABSTRACT

Along with technological developments, the number of uses of web services is increasing. This causes popular websites to have high amounts of traffic. One of the popular websites is e-shop. The problem faced is when certain activities or events in e-shops can also cause an increase in the amount of web traffic of an organization. One solution to deal with these problems is to use a reliable server with high performance. Another solution that can be applied is load balancing technology. Round robin algorithm is the simplest and most widely used algorithm by load balancing devices. Therefore a web was created to implement load balancing using round robin queues with e-shop case studies. The testing process in this study was calculated using QoS testing consisting of four parameters. The first, throughput is calculated and produces a value of 0.00241841 kbps. Second, the delay obtained from the calculation results is 6.086161 ms, 0.010741 ms, 6.069903 ms, and 6.069903 ms classified into very good categories for the use of transactions on web e-shops. Third, jitter obtained from the calculation result is -0,000541933 ms which is classified into a very good category for the use of transactions on the e-shop web. Finally, packet loss obtained from the calculation result of 0% is classified into the very good category for the use of transactions on e-shop webs.

Article History

Received 2020-08-14
Revised 2020-08-26
Accepted 2020-09-02

Key words

Load Balancing, Round Robin Algorithm, QoS, E-Shop

ABSTRAK

Seiring dengan perkembangan teknologi, jumlah penggunaan layanan web semakin meningkat. Hal ini menyebabkan situs-situs web popular memiliki jumlah traffic yang tinggi. Salah satu web yang populer itu adalah e-shop. Masalah yang dihadapi adalah pada saat ada kegiatan atau acara tertentu dalam e-shop juga dapat menyebabkan naiknya jumlah traffic web suatu organisasi. Salah satu solusi untuk menangani masalah tersebut adalah dengan menggunakan sebuah server yang handal dengan performa yang tinggi. Solusi lain yang dapat diterapkan adalah teknologi load balancing. Algoritma Round robin merupakan algoritma yang paling sederhana dan paling banyak digunakan oleh perangkat load balancing. Oleh karena itu dibuatlah sebuah web untuk mengimplementasikan load balancing menggunakan antrian round robin dengan studi kasus e-shop. Proses pengujian pada penelitian ini dihitung menggunakan pengujian QoS terdiri dari empat parameter. Yang pertama, throughput yang dihitung dan menghasilkan nilai 0,00241841 kbps. Kedua, delay yang didapat dari hasil perhitungan yaitu 6,086161 ms, 0,010741 ms, 6,069903 ms, dan 6,069903 ms digolongkan kedalam kategori sangat baik untuk penggunaan transaksi pada web e-shop. Ketiga, jitter yang didapat dari hasil perhitungan yaitu -0,000541933 ms yang digolongkan kedalam kategori sangat baik untuk penggunaan transaksi pada web e-shop. Terakhir, packet loss yang didapat dari hasil perhitungan yaitu 0% digolongkan kedalam kategori sangat baik untuk penggunaan transaksi pada web e-shop.

PENDAHULUAN

Seiring dengan perkembangan teknologi, jumlah penggunaan layanan web semakin meningkat. Hal ini menyebabkan situs-situs web popular memiliki jumlah traffic yang tinggi. Salah satu web yang populer itu adalah e-shop. E-shop merupakan sarana penjualan, pembelian, dan pemasaran barang atau jasa melalui sistem elektronik seperti web.

Masalah yang dihadapi adalah pada saat ada kegiatan atau acara tertentu dalam e-shop juga dapat menyebabkan naiknya jumlah traffic web suatu organisasi. Peningkatan jumlah traffic menyebabkan kerja server yang melayani permintaan menjadi semakin berat. Akibatnya performa

server menurun dan sering terjadi gangguan pada layanan-layanan web tersebut. Jika tidak di tangani, hal ini dapat mengakibatkan sistem atau situs web tersebut mati/down.

Salah satu solusi untuk menangani masalah tersebut adalah dengan menggunakan sebuah server yang handal dengan performa yang tinggi. Solusi lain yang dapat diterapkan adalah teknologi load balancing. Teknologi load balancing dapat membagi beban permintaan ke beberapa web server sehingga teknologi ini memiliki peranan penting dalam mencapai penggunaan sumber daya bersama yang efektif. Algoritma Round robin merupakan algoritma yang paling sederhana dan paling banyak digunakan oleh perangkat load balancing. Algoritma Round robin bekerja dengan cara membagi beban secara bergiliran dan berurutan dari satu server ke server lainnya.

Berdasarkan uraian diatas maka dibuatlah sebuah web untuk mengimplementasikan load balancing menggunakan antrian round robin dengan studi kasus e-shop.

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat dirumuskan permasalahan. Yaitu bagaimana mengimplementasikan metode load balancing round robin terhadap web e-shop dan bagaimana kinerja server sebelum dan sesudah diimplementasikan metode load balancing.

Berdasarkan rumusan masalah diatas maka tujuan penelitian ini adalah untuk dapat mengimplementasikan metode load balancing round robin terhadap web e-shop. Dan untuk mengetahui kinerja dari server sebelum dan sesudah diterapkan metode load balancing round robin. Manfaat yang diharapkan dapat membantu beban kerja server yang digunakan dalam pengaksesan web dengan kapasitas yang cukup tinggi dan traffic yang padat sehingga web dapat diakses dengan lancar.

TINJAUAN PUSTAKA

Jaringan Komputer

Jaringan Komputer adalah suatu sistem telekomunikasi yang didalamnya terdiri dari dua atau lebih perangkat komputer yang dirancang untuk dapat berkerja secara bersama-sama dengan tujuan dapat berkomunikasi, mengakses informasi, meminta serta memberikan layanan atau *service* antara komputer satu dengan yang lainnya.

Web Server

Web server adalah sebuah software yang memberikan layanan berbasis data dengan menggunakan protokol HTTP atau HTTPS dari client menggunkan aplikasi *web browser* untuk *request* data dan *server* akan mengirim data dalam bentuk halaman *web* dan pada umumnya berbentuk dokumen *HTML*. Halaman *web* yang diminta bisa terdiri dari berkas teks, video, gambar, *file* dan banyak lagi.

Load Balancing

Load balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. *Load balancing* digunakan pada saat sebuah server telah memiliki jumlah user yang telah melebihi maksimal kapasitasnya. *Load balancing* juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, link jaringan, CPU, *hard drive*, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal.

Data *Server*

Database Server adalah aplikasi komputer yang menyediakan layanan data kekomputer atau program komputer dan memiliki fungsi sebagai tempat penyimpanan data, seperti yang ditetapkan oleh model klien-server. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program server *database*. Salah satu contoh aplikasi database adalah MySQL, Oracle, MariaDB dan lain sebagainya.

Algoritma Round Robin

Algoritma *Round Robin* merupakan algoritma paling sederhana dan banyak digunakan oleh perangkat load balancing. Algoritma ini membagi beban dengan cara bergiliran dan berurutan dari satu server ke server lainnya sehingga membentuk perputaran.

Pengujian

Pengujian yang digunakan pada program ini adalah QoS (Quality of Services) atau yang lebih dikenal sebagai manajemen bandwidth. QoS merupakan suatu metode pengukuran seberapa baik jaringan. Qos digunakan untuk mengukur sekumpulan atribut kinerja suatu program yang telah dispesifikasikan dan diasosiasikan dengan suatu servis. Model monitoring QoS terdiri dari parameter, sebagai berikut:

1. *Throughput*

Throughput yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps (*bit per second*). Troughput adalah jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

$$throughput = \frac{paket\ data\ diterima}{lama\ pengamatan} \quad \text{.....(1)}$$

2. *Packet Loss*

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena collision dan congestion pada jaringan.

$$packet\ loss = \frac{(paket\ data\ dikirim - paket\ data\ diterima) \times 100\%}{paket\ data\ yang\ dikirim} \quad \text{.....(2)}$$

3. *Delay (Latency)*

Delay (letency) merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, congesti atau juga waktu proses yang lama.

$$delay = \frac{packet\ length}{link\ bandwidth} \quad \text{.....(3)}$$

4. Jitter

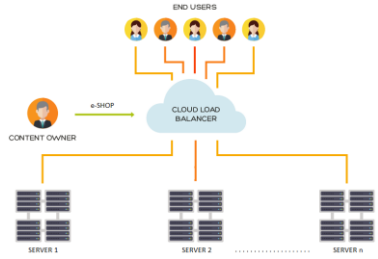
Jitter atau variasi kedatangan paket diakibatkan oleh variasi – variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket – paket diakhir perjalanan jitter. Jitter lazimnya disebut variasi delay, berhubungan erat dengan latency, yang menunjukkan banyaknya variasi delay pada transmisi data di jaringan.

$$jitter = \frac{total\ \ variasi\ delay}{total\ paket\ yang\ diterima} \quad \text{.....(4)}$$

METODE

Gambaran Umum

Dalam penelitian ini diterapkan sistem menggunakan algoritma *Round Robin*. Dimana hal ini berguna untuk membantu beban kerja server yang digunakan dalam pengaksesan web dengan kapasitas yang cukup tinggi dan traffic yang padat sehingga web dapat diakses dengan lancar.

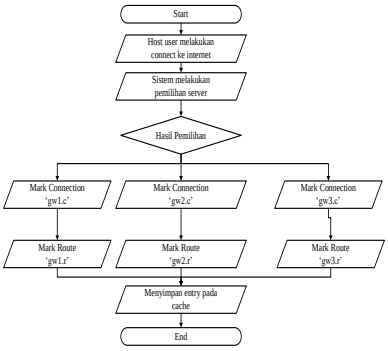


Gambar 1. Skema Perancangan Aplikasi

Gambar 1 diatas menjelaskan tentang skema rancangan *load balancing round robin*. Dimana *client* 1, *client* 2, *client* 3 dan seterusnya yang memiliki IP *address* dan ID pemain melakukan *request* ke *server*. Maka sistem *load balancing* akan secara otomatis untu membuat antrian secara merata kepada *server* yang ada dalam hal ini adalah *server* 1, *server* 2, *server* 3, dan seterusnya. Hal ini dapat memungkinkan masing – masing *server* untuk menanggung beban yang lebih ringan. *Load balancing* adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, dan menghindari *overload* pada salah satu jalur koneksi. Dapat dilakukan dengan cara memaksimalkan beberapa parameter yaitu *delay*, *packet loss*, *jitter*, *availability* dan *throughput* menggunakan software *winbox*.

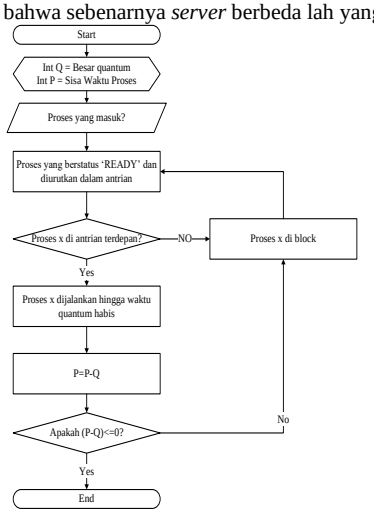
Rancangan dan Pembangunan

Tahap ini menjelaskan tentang prosedur dan proses apa saja yang akan dilakukan oleh aplikasi, alur proses, serta tampilan dasar aplikasi. Alur proses dalam hal ini berbentuk flowchart dan perhitungan manual dari metode yang digunakan. *Flowchart* dari sistem yang dibangun dapat dilihat pada gambar 3.2.



Gambar 2. Arsitektur Load Balancing

Gambar 2 diatas menjelaskan tentang rancangan arsitektur *Load Balancing* yang akan digunakan. Pertama pengguna diwajibkan untuk terhubung dengan *internet*. Lalu ketika pengguna mengirimkan *request* pada *server* maka sistem akan memilih *server* mana yang akan menerima. Kemudian *server* yang menerima *request* tersebut akan memproses permintaan pengguna tersebut hingga seluruh *request* yang diminta selesai. Maka saat pengguna pertama masih melakukan transaksi dan muncul pengguna lain maka sistem akan secara otomatis memilih *server* yang lain yang tidak sedang memproses *request* dari pengguna. Sehingga pengguna dapat mengakses *web* secara bersamaan tanpa mengetahui bahwa sebenarnya *server* berbeda lah yang menanggapi mereka.



Gambar 3 *Flowchart Round Robin*

Gambar 3 menjelaskan tentang alur program dari algoritma *round robin*. Dimana diketahui Q adalah besar quantum dengan satuan *integer* dan P adalah sisa waktu proses dengan satuan *integer*. Ketika proses masuk maka proses akan akan diurutkan dalam antrian (*ready*), jika antrian sudah mencapai antrian terdepan maka proses akan dijalankan sampai waktu quantum habis, jika antrian belum mencapai terdepan maka antrian akan di block dan akan diurutkan kembali ke dalam antrian. Setelah proses antrian berjalan sampai waktu *quantum* habis maka nilai R(waktu proses) = R(waktu proses)

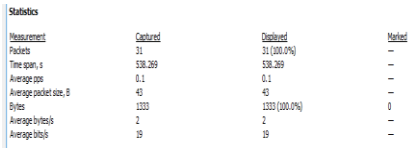
– Q(waktu quantum), jika nilai (R-Q) kurang dari atau sama dengan 0 makan proses antrian selesai. Akan tetapi, jika nilai (R-Q) lebih dari atau sama dengan 0 maka proses akan di *block* dan akan di urutkan kembali ke antrian.Untuk mengetahui bagaimana cara menghitung penjadwalan *Round robin.* Berikut ini merupakan contoh penjadwalan *Round robin*.

HASIL DAN PEMBAHASAN

Pengujian sistem pada implementasi *Load Balancing* menggunakan antrian *Round Robin* dengan studi kasus *e-shop* ini dilakukan menggunakan pengujian QoS. Adapun parameter QoS yang digunakan dalam pengukuran meliputi *Throughput*, *packet Loss*, *Delay*, dan *Jitter*. Adapun metode pengambilan dataset yaitu :

1. Waktu pengambilan data dibatasi kurang dari 1 menit.
2. Perangkat lunak yang digunakan adalah *wireshark.*
3. Pengukuran QoS dilakukan pada parameter *delay*, *jitter*, *throughput*, dan *packet loss*.
4. Pengukuran dilakukan dari sisi *client*.

Hasil pengujian yang dilakukan pada *client* melalui perangkat lunak *wireshark* adalah sebagai berikut:

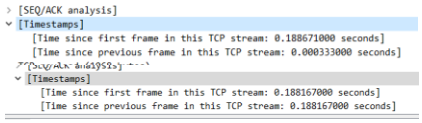


Statistics

Measurement	Captured	Displayed	Marked
Packets	31	31 (100.0%)	–
Time span, s	538.269	538.269	–
Average pps	0.1	0.1	–
Average packet size, B	43	43	–
Bytes	1333	1333 (100.0%)	0
Average bytes/s	2	2	–
Average bits/s	19	19	–

Gambar 4 Hasil *Capture Wireshark*

Gambar 4 merupakan hasil *screenshoot* dari perangkat lunak *wireshark* yang menjadi dasar untuk perhitungan parameter pengujian QoS.



Gambar 5 Hasil *Capture Delay* pada *Wireshark*

Gambar 5 merupakan hasil *screenshoot* dari perangkat lunak *wireshark* untuk menghitung variasi *delay*. Terdapat lebih dari satu waktu *delay* dikarenakan *client* dapat melakukan transaksi lebih dari sekali pada satu pengujian. Dengan syarat pengujian tersebut dilakukan saat waktu *quantum* yang ditentukan pada algoritma *Round Robin* belum habis.

Pengujian *Throughput*

Throughput dapat dihitung dengan rumus :

$$Throughput = \frac{paket\ data\ yang\ diterima}{lama\ pengamatan}$$

Berdasarkan gambar 4 maka didapatkan nilai paket data yang diterima sebanyak $1333\ byte$ dan lama pengamatan $538{,}269\ s$. Maka nilai throughput yang dapat dihitung dengan cara perhitungan sebagai berikut :

$$Throughput = \frac{1333\ byte}{538{,}269\ s} = 2{,}47645693882\ byte/s$$

$$Throughput = \frac{2{,}47645693882\ byte/s}{1024} = 0{,}00241841\ Kbps$$

Sehingga didapatkan nilai throughput dari capture gambar 4 yaitu 0,00241841 kbps.

Commented [SA1]: Sebaiknya ini dihapus, dibuat dalam satuan bps saja dan angkanya dibuat 2 digit saja di belakang koma

Commented [SA2]: Dihilangkan saja

Pengujian *Delay*

Untuk menghitung rata – rata *delay* digunakan rumus :

$$Delay = \frac{Total\ Delay}{Total\ packet\ yang\ diterima}$$

Berdasarkan gambar 5 maka didapatkan empat waktu *delay* yaitu 0,188671000 s, 0,000333000 s, 0,188167000 s, dan 0,188167000 s. Sedangkan untuk total *packet* yang diterima terdapat pada gambar 4 yaitu 31. Berikut ini adalah perhitungan delay :

$$Rata\ delay\ 1 = \frac{0{,}188671000}{31} = 0{,}00608616129\ s = 6{,}086161\ ms$$

$$Rata\ \ delay\ 2 = \frac{0{,}000333000}{31} = 0{,}00001074194\ s = 0{,}010741\ ms$$

$$Rata\ \ delay\ 3 = \frac{0{,}188167000}{31} = 0{,}00606990323\ s = 6{,}069903\ ms$$

$$Rata\ \ delay\ 4 = \frac{0{,}188167000}{31} = 0{,}00606990323\ s = 6{,}069903\ ms$$

Dari hasil rata – rata *delay* maka rata – rata *delay* yang didapatkan dapat digolongkan kedalam kategori sangat baik pada penggunaan transaksi *web e-shop*.

Commented [SA3]: Penggunaan kata-kata yang membingungkan. Jadi delay rata-rata yang mana?

Commented [SA4]: Menurut standar apa?

Commented [SA5]: spasi

Pengujian *Jitter*

Untuk menghitung *jitter* menggunakan rumus :

$$Jitter = \frac{Total\ variasi\ delay}{Total\ packet\ yang\ diterima - 1}$$

Commented [SA6]: persamaan ini beda dengan persamaan 4. Rumus hanya ditulis sekali saja di persamaan 4

Total variasi *delay* diperoleh dari penjumlahan :

$Total\ variasi\ delay = (delay\ 2 - delay 1) + \cdots + (delay\ n - delay(n-1))$

Berdasarkan gambar 4 didapatkan nilai total paket yang diterima yaitu 31. Sebelum menghitung nilai *jitter* maka hal yang perlu dihitung terlebih dahulu adalah total variasi *delay* :

$Total\ variasi\ delay = (0{,}010741\ ms - 6{,}086161\ ms) + (6{,}069903\ ms - 0{,}010741\ ms) + (6{,}069903\ ms - 6{,}069903\ ms)$

$$Total\ variasi\ delay = (-6{,}07542) + 6{,}059162 + 0 = -0{,}016258\ ms$$

Dari gambar 5 maka didapatkan *jitter* dengan cara perhitungan sebagai berikut:

$$Jitter = \frac{-0{,}016258\ ms}{31 - 1} = -0{,}000541933$$

Dari hasil *jitter* yaitu -0,000541933 maka nilai *jitter* yang didapatkan dapat digolongkan kedalam kategori sangat baik padapenggunaan transaksi *web e-shop*.

Commented [SA7]: standarnya apa?

Pengujian *Packet Loss*

Packet loss adalah jumlah paket data yang hilang per detik. *Packet loss* dapat dihitung dengan rumus :

$$Packet\ Loss = \frac{paket\ data\ yg\ dikirm - paket\ data\ yg\ diterima}{paket\ data\ yang\ dikirim} \times 100\%$$

Commented [SA8]: Rumus ini sudah ditulis di atas jadi tidak perlu ditulis lagi. Kalimatnya bisa diganti “dapat dihitung dengan persamaan 2”

Berdasarkan gambar 4 maka didapatkan paket data yang dikirim adalah 31 dan paket data yang diterima adalah 31. Maka didapatkan nilai packet loss dengan cara perhitungan sebagai berikut:

$$Packet\ Loss = \frac{31 - 31}{31} \times 100\% = 0\%$$

Dari hasil *packet loss* yaitu 0% maka nilai *p* yang didapatkan dapat digolongkan kedalam kategori sangat baik pada penggunaan transaksi *web e-shop*.

KESIMPULAN

Dari hasil analisa yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Dari pengujian yang dilakukan membuktikan bahwa *load balancing* yang diterapkan pada aplikasi yang dibuat berhasil untuk mengoptimasi pada *e – shop*. Aplikasi berhasil dijalankan dan memberikan *output* berupa hasil pemesanan yang cepat menggunakan algoritma *Round Robin*.
2. Pengujian sistem yang dilakukan dihitung menggunakan pengujian QoS terdiri dari empat parameter. Yang pertama, *throughput* yang dihitung dan menghasilkan nilai 0,00241841 kbps. Kedua, *delay* yang didapat dari hasil perhitungan yaitu 6,086161 ms, 0,010741 ms, 6,069903 ms, dan 6,069903 ms digolongkan kedalam kategori sangat baik untuk penggunaan

transaksi pada *web e-shop*. Ketiga, *jitter* yang didapat dari hasil perhitungan yaitu $-0{,}000541933\ ms$ yang digolongkan kedalam kategori sangat baik untuk penggunaan transaksi pada *web e-shop*. Terakhir, *packet loss* yang didapat dari hasil perhitungan yaitu 0% digolongkan kedalam kategori sangat baik untuk penggunaan transaksi pada *web e-shop*.

DAFTAR PUSTAKA

[1] Ansharullah, Khairul. (2016). "Implementasi Sistem Load Balancing Dengan Algoritma Round Robin Untuk Mengatasi Beban Server Di SMK Negeri 2 Kudus", Buku Skripsi Pendidikan Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang. Semarang.

[2] Gea, Asaziduhu. (2015)."Optimasi Turn Around Time Pada Penjadwalan Round Robin Dengan Mencari Quantum Time Optimal Menggunakan Algoritma Simulated Annealing", Jurnal Methodika, Vol. 1, No. 1.

[3] Gunawan. (2018). "Implementasi Algoritma Round Robin Pada Sistem Penjadwalan Mata Kuliah (Studi Kasus: Universitas Muhammadiyah Bengkulu)", Universitas Muhammadiyah Bengkulu

[4] P., Yogie Ariyanto. Wicaksono, Soetam Rizky. (2017). "Sistem Pendukung Keputusan Penjadwalan Pengemudi Dengan Menggunakan Algoritma Round Robin (Studi Kasus: Zena Ttravel)", Jurusan Sistem Informasi, Fakultas Sains dan Teknologi. Vol. 6, No. 1. P-ISSN: 2303-3142. E-ISSN: 2548-8570.

[5] Putra, Rifaldi Adi. Lutfi, Helmi Faisal. (2019)."Implementasi Load Balancing Algoritma Round Robin Pada F5 Big-IP Local Traffic Manager", Siliwangi University.

[6] Rahmana, Dany. Rakhmadhany Primananda. Widhi Yahya. (2018). "Analisis Load Balancing Pada Web Server Menggunakan Algoritme Weighted Least Connection". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 3, Hal. 915-920. e-ISSN : 2548-964x.

[7] Rahmatulloh, Alam. Firmansyah MSN. (2017). "Implementasi Load Balancing Web Server Menggunakan Haproxy dan Sinkronisasi File Pada Sistem Informasi Akademik Universitas Siliwangi". Jurnal Nasional Teknologi dan Sistem Informasi, Vol. 03, No. 02. ISSN (Print): 2460-3465 ISSN (Online) : 2476-8812

[8] Riskiono, Sampurna Dadi. Pasha, Donaya. (2020). "Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning". Jurnal TeknoInfo, Vol. 14, No. 1, Hal. 22-26. ISSN : 2615-224x.

[9] Triono, Gaguk. (2015). "Implementasi Load Balancing Dengan Menggunakan Algoritma Round Robin Pada Kasus Pendaftaran Siswa Baru Sekolah Menengah Pertama Labschool Unesa Surabaya". Seminar Nasional "Inovasi dalam Desain dan Teknologi."-IDeaTech. ISSN: 2089-1121.