

Analisis Performa *Load Balancing* Algoritma *Weighted Round Robin* di Infrastruktur BPBD Provinsi Jawa Timur

Molion Surya Pradana¹, Aditya Prapanca²,

¹Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

²Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

molionpradana@mhs.unesa.ac.id

adityaprapanca@unesa.ac.id

Abstrak— Bencana alam terus terjadi dengan waktu kejadian yang tidak dapat diprediksi. Pada saat bencana terjadi seluruh masyarakat ingin mendapatkan informasi yang benar dan akurat. Sumber informasi tersebut dapat diperoleh pada *website* lembaga pemerintahan non-departemen yang bergerak di bidang social yaitu Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Jawa Timur.

Meningkatnya jumlah pengunjung setiap tahun pada *website* BPBD Jatim membuat kondisi *web server* akan *overload* sehingga tidak semua orang dapat mengakses *website* untuk mencari informasi. Untuk menghindari hal tersebut, dengan memasang aplikasi *load balancer* pada *web server* dapat menambah performa *web server* menjadi lebih cepat dan semua *request* yang masuk akan terkelola dengan baik.

Dari hasil penelitian ini dapat dinyatakan bahwa Algoritma *weighted round robin* dengan rasio (2:3) yang terpasang pada *load balancer* dapat membantu meningkatkan performa parameter *Troughput* sebanyak 113 KB/s, dan *Request error* sebanyak 986 client pada *web server* yang belum menggunakan *load balancer*.

Kata Kunci— Bencana alam, BPBD, Overload, Load Balancer, Web Server, Weighted Round Robin.

I. PENDAHULUAN

Indonesia sebagai negara khatulistiwa yang sebagian besar wilayahnya juga dikelilingi gunung berapi memiliki banyak potensi kekayaan alam karena tanahnya yang subur. Namun selain itu, Indonesia juga dikenal sebagai wilayah cincin api (*ring fire*) sehingga memiliki potensi banyak bencana. Menyadari akan hal itu pemerintah republik Indonesia telah mendirikan Badan Nasional Penanggulangan Bencana (BNPB) yang bertugas sewaktu-waktu ketika mengalami bencana. Bahkan, pemerintah telah membentuk Badan Penanggulangan Bencana Daerah (BPBD) di setiap wilayah provinsi hingga wilayah kabupaten atau kota.

Pusat Pengendalian Operasi Penanggulangan Bencana (Pusdalops-PB) merupakan salah satu unit kerja yang terdapat dalam BPBD Provinsi Jawa Timur yang memiliki tugas sebagai pusat pengumpulan data pada setiap kejadian bencana yang terjadi di daerah Provinsi Jawa Timur. Data yang dimiliki oleh Pusdalops-PB dijadikan sebagai arsip yang nantinya akan menjadi akses layanan informasi bagi petinggi pemerintahan Provinsi Jawa Timur, di antaranya kepala pelaksana, gubernur, pejabat pemerintahan lainnya, dan masyarakat.

Dalam perkembangan teknologi saat ini, banyak orang sudah sangat membutuhkan *website* untuk memudahkan urusannya. Selain itu, banyak orang juga sudah menyadari akan kebutuhan ketersediaan aplikasi berbasis web untuk mempermudah dalam mengelola proses. Saat ini telah banyak

aplikasi berbasis web yang digunakan di instansi-instansi pemerintah. Dengan aplikasi tersebut banyak orang melakukan akses ke dalam *website*.

Load balancing merupakan teknik untuk mendistribusikan beban trafik yang ada pada sekumpulan server atau perangkat jaringan ketika mendapatkan permintaan dari pengguna. Dengan *load balancing*, trafik dapat berjalan secara optimal, *throughput* dapat dimaksimalkan, waktu tanggap dapat diperkecil, dan *overload* pada salah satu server dapat (Lukitasari & Oklilas, 2010). *Load balancing* dapat diterapkan pada lingkungan fisik dan lingkungan virtual dalam hal ini *cloud computing*. Salah satu layanan dari *cloud computing* adalah *Infrastructure as a Service* (IaaS). IaaS adalah sebuah layanan yang “menyewakan” sumberdaya teknologi informasi dasar yang meliputi media penyimpanan, processing power, memory, sistem operasi, kapasitas jaringan dan lain lain, yang dapat digunakan oleh penyewa untuk menjalankan aplikasi yang dimilikinya (Sitorius, 2016).

Karena perkembangan teknologi yang semakin meningkat, kini terdapat algoritma terapan dari *round robin* yang bernama *weighted round robin*. Algoritma *weighted round robin* memiliki konsep yang sama dengan algoritma *round robin*, akan tetapi diberikan tambahan kondisi baru yaitu mempertimbangkan kemampuan sumber daya server dan memberikan jumlah tugas yang lebih tinggi ke server yang mempunyai kapasitas yang lebih tinggi (Devi & Uthariaraj, 2016).

Saat ini BPBD Jatim belum menggunakan *load balancing* pada server yang dimiliki sehingga kinerjanya masih belum optimal untuk pembagian kinerja server. Dalam pertumbuhan dan perkembangan internet, kinerja dari sebuah server semakin padat karena kebutuhan serta fungsi dari internet semakin meningkat dan bertambah. Hal ini mengakibatkan sering terjadinya *overloading* pada server. Untuk mengatasi kondisi *overloading* tersebut, diusulkan penggunaan metode *load balancing* dengan algoritma *weighted round robin*.

II. METODOLOGI PENELITIAN

Penelitian yang dilakukan adalah mengenai hasil akhir perbandingan antara *Load balancing* menggunakan algoritma *weighted round robin* dengan kondisi optimasi server yang saat ini beroperasi di Badan Penanggulangan Bencana Daerah Provinsi Jawa Timur.

Nilai parameter pada penelitian ini yaitu membandingkan *Throughput*, *Response Time*, *Error request*, dan *CPU Utilization*.

A. Rancangan Penelitian

Pada tahap perancangan sistem menggunakan sistem operasi CentOS yang mendukung dalam pembuatan *load balancing* server. Metode pada perancangan topologi ini adalah LVS via *Direct Routing* menggunakan algoritma *weighted round robin*. Penelitian yang dilakukan merupakan simulasi pengukuran kinerja *load balancer* server di lingkungan Badan Penanggulangan Bencana Daerah Provinsi Jawa Timur. Pada table 1 adalah spesifikasi pada masing-masing *hardware*.

Tabel 1 Spesifikasi Hardware

Load Balancer	
Sistem Operasi	Linux CentOS 6.10
Aplikasi	Piranha, IPVSADM
IP Address	10.10.10.21
VIP	10.10.10.20
Web Server	
Sistem Operasi	Linux CentOS 6.10
Aplikasi	Apache Web Server
IP Address Server 1	10.10.10.23
IP Address Server 2	10.10.10.24
VIP	10.10.10.20
Database	
Sistem Operasi	Linux CentOS CLI 7
Aplikasi	MariaDB
IP Address	10.10.10.20
User	
Sistem Operasi	Linux CentOS 6.10
IP Adress	10.10.1.100

B. Teknik Analisis Data

1. Analisis Sistem Lama

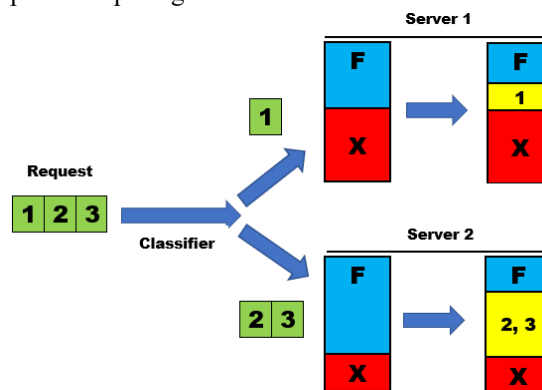
Pada penelitian ini perlu dilakukan Analisis sistem lama atau sistem yang sudah ada. Di Badan Penanggulangan Bencana Daerah Provinsi Jawa Timur sudah memiliki infrastruktur jaringan cukup bagus pada *hardware*. Akan tetapi, dari segi konfigurasi jaringan masih belum menggunakan *load balancing*. Dengan sistem yang lama tersebut sering terjadi server *down* ketika banyak pengguna yang mengakses ke server.

2. Analisis Sistem Baru

Pada penelitian-penelitian yang telah ada telah terbukti bahwa *load balancing* dapat meningkatkan kinerja server secara efisien. Salah satunya menggunakan algoritma *weighted round robin* yang membuat skenario pada pembagian pekerjaan yang akan diproses oleh server secara rata dan bergulir, sehingga akan mengurangi terjadinya *error*, *server loss*, atau pun lambatnya proses.

Pada gambar 1 menerapkan algoritma *weighted round robin*. Dalam algoritma ini, diberikan kondisi tambahan yaitu jika terdapat *request* yang datang secara bersamaan. Maka *request* yang datang akan diclassifier terlebih dahulu dan akan dibagikan ke server secara

satu banding dua (1:2) dikarenakan hanya memiliki 2 server. Request pertama akan diproses ke dalam server 1 kemudian pada server 2 karena memiliki ruang kosong yang dapat melakukan proses request lebih banyak akan diberi 2 request secara bersamaan. Request 1 akan diproses ke dalam server 1, sementara request 2 dan 3 akan diproses ke dalam server 2. Karena pendistribusian request yang akan datang diproses secara berbeda, maka akan membuat server bekerja lebih stabil dan seimbang, sehingga meminimalisir server dari *overload* dan persentase kegagalan melakukan request atau *error request*, seperti alur pada gambar 2.



Gbr. 1 Proses pembagian request weighted round robin.

Keterangan gambar 1:

- Warna hijau : request yang akan diproses
- Warna merah : memori server yang telah terpakai
- Warna biru : memori server yang belum terpakai
- Warna kuning : request baru yang sedang diproses

C. Uji Coba dan Analisa

1. Kebutuhan Proses dan Uji Coba

Tahap kebutuhan proses dan uji coba pada penelitian ini sebagai berikut.

- a) Beban *traffic* yang terhubung pada server Total beban *traffic* yang terhubung pada server diperoleh dari menjumlahkan setiap koneksi yang terhubung pada masing-masing webserver. Data yang diperoleh akan dirata-rata setiap harinya, guna mengetahui berapa banyak rata-rata beban *traffic* yang terhubung dengan menggunakan rumus (1).

$$T = \frac{H_1 + H_2 + H_3 + \dots}{J} \quad (1)$$

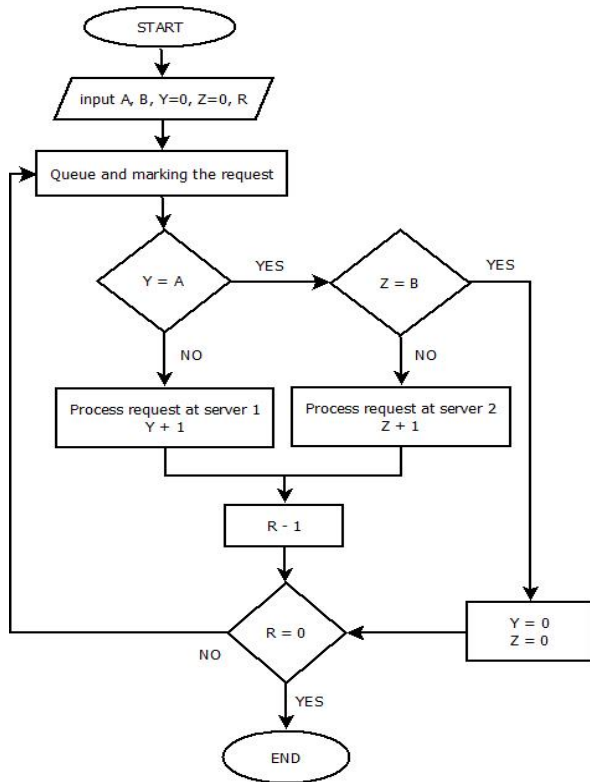
Keterangan formula (1):

T = Rata-rata banyaknya beban *traffic* setiap hari

H = Jumlah beban *traffic* pada setiap harinya

J = Jumlah banyaknya hari yang akan dihitung.

c. (2:3) = 2 request pada VS1, 3 request pada VS2



Gbr. 2 Proses pembagian request round robin.

b) Uji Coba

Aplikasi web pada BPBD akan diclone kedalam virtual server sebagai media uji coba. Angka dari masing-masing nilai parameter akan didapatkan dengan aplikasi Httpperf. Httpperf diujicobakan menggunakan data jumlah beban setiap harinya dengan proses yang ditentukan. Uji coba akan dilakukan sebanyak 5 kali uji coba request yang berbeda, dan masing-masing uji coba request akan dilakukan sebanyak 20 kali percobaan guna mendapatkan nilai tengah atau rata-rata dari nilai parameter yang didapatkan. seperti yang dijelaskan pada table 2.

Tabel 2 Uji Coba Request.

No	Banyaknya Request/5 detik	Diuji sebanyak
1	100 Request	20 kali
2	2500 Request	20 kali
3	5000 Request	20 kali
4	7500 Request	20 kali
5	10000 Request	20 kali

Berdasarkan permintaan request, load balancing akan mencari kombinasi bobot agar kedua virtual server mendapatkan *utilization* yang seimbang. Adapun contoh kombinasi bobot sebagai berikut.

- a. (2:1) = 2 request pada VS1, 1 request pada VS2
- b. (3:1) = 3 request pada VS1, 1 request pada VS2

2. Analisis Hasil Output

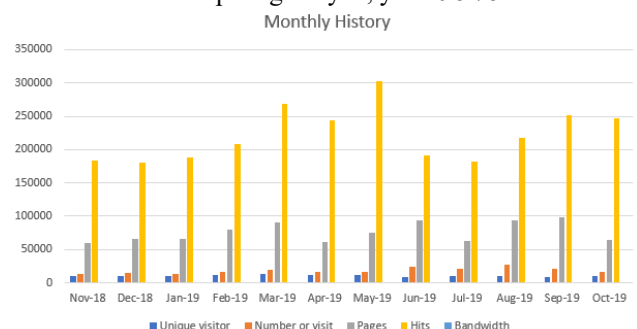
Analisis hasil output merupakan akhir dari penelitian ini. Dari uji coba *load balancing* akan diketahui perbedaan kinerja setiap optimasi *load balancing* tanpa menggunakan algoritma maupun menggunakan algoritma *round robin* dan algoritma *weighted round robin*. Nantinya akan didapatkan nilai-nilai parameter yaitu *Throughput*, *Response Time*, *Error request*, dan *CPU Utilization* diukur menggunakan Htop yang dapat dibuka melalui terminal linux centos. Parameter tersebut kemudian akan digunakan sebagai data pembandingan setiap kinerja *load balancing*.

III. HASIL DAN PEMBAHASAN

A. Kondisi Pemakaian Beban Server di Infrastruktur BPBD Provinsi Jatim

Pusdalops-PB memiliki portal *website* sebagai sumber informasi kejadian bencana yang ditujukan untuk melayani masyarakat. *Website* tersebut sudah aktif sejak 2017. Dari waktu ke waktu hingga 2019 saat ini masyarakat mulai aktif untuk ingin tahu kejadian bencana setiap harinya melalui *website* BPBD Jatim. Jumlah masyarakat yang mengakses *website* dapat dilihat pada gambar 3.

Berdasarkan gambar 3 bulan Mei 2019 adalah bulan yang memiliki jumlah request pada *website* paling banyak. Terdapat 302.758 request, namun hanya memiliki 17.075 user yang mengakses. Hal tersebut terjadi karena dari 17.075 user pada Mei 2019 memiliki rasa ingin tahu yang lebih banyak mengenai informasi kejadian bencana alam. Hal itu berbeda dengan Agustus 2019 Pada Agustus 2019 memiliki 27.198 user, namun hanya memproses 217.690 request. Pemakaian bandwidth terbanyak yaitu pada bulan September 2019, banyaknya bandwidth mencapai 8.25 GB. Tingginya pemakaian bandwidth dipengaruhi oleh banyaknya halaman yang dibuka pada bulan tersebut. Oleh karena itu, bulan September 2019 juga menjadi bulan yang membuka halaman paling banyak, yaitu 98.781 halaman.



Gbr. 3 Riwayat beban server bulanan.

B. Penerapan Algoritma *Weighted Round Robin* dalam Melakukan *Load Balancing*

Sebelum mengimplementasikan algoritma *weighted round robin*, pertama-tama perlu dibuat rangkaian jaringan

seperti pada gambar 4. Rangkaian jaringan tersebut memerlukan 5 buah virtual PC, 2 adaptor dan layanan internet (ISP). Ada 1 virtual diperuntukkan menjadi PC User, 1 virtual diperuntukkan menjadi *Load Balancer*, 2 virtual diperuntukkan menjadi virtual server 1 dan 2, dan 1 virtual diperuntukkan menjadi database. Pada user terdapat 2 ethernet, yaitu ethernet 1 terhubung ke dalam adaptor 1 menggunakan IP static 10.10.1.100 dan ethernet 2 terhubung pada internet menggunakan IP dinamis. Pada load balancer terdapat 2 Ethernet juga, yaitu ethernet 1 terhubung dengan adaptor 1 menggunakan IP static 10.10.1.21 dan ethernet 2 terhubung pada internet menggunakan IP dinamis. Pada virtual server 1 terdapat 3 ethernet, yaitu ethernet 1 terhubung ke adaptor 1 menggunakan IP static 10.10.1.23, ethernet 2 terhubung ke adaptor 2 menggunakan IP static 10.20.1.1, dan ethernet 3 terhubung ke internet menggunakan IP dinamis. Pada virtual server 2 terdapat 3 ethernet juga, yaitu ethernet 1 terhubung ke adaptor 1 menggunakan IP static 10.10.1.24, ethernet 2 terhubung ke adaptor 2 menggunakan IP static 10.20.1.2, dan ethernet 3 terhubung ke internet menggunakan IP dinamis. Pada virtual database terdapat 2 ethernet, yaitu ethernet 1 terhubung pada adaptor 2 menggunakan IP static 10.20.1.3, dan ethernet 2 terhubung pada internet menggunakan IP dinamis. Semua IP static yang terhubung pada rangkaian menggunakan IP dengan prefix 24. Virtual IP (VIP) menggunakan IP 10.10.1.20 yang mana nanti dapat dikonfigurasi di dalam aplikasi load balancer yaitu Piranha.

1. Layanan Multi-Port (*Multi-Port Services*)

Kita akan menggunakan layanan multi-port (HTTP dan HTTPS). Oleh karena itu mark firewall untuk menggabungkan keduanya berbeda, tetapi memerlukan protokol keduanya berhubungan. Namun, tugas menetapkan mark firewall harus dilakukan oleh filter paket jaringan, iptables, dan di luar alat konfigurasi Piranha. Menetapkan mark firewall pada router, virtual load balancer:

```
# iptables -t mangle -A PREROUTING -p tcp -d 10.10.1.20/24 -m multiport --dport 80,443 -j MARK --set-mark 80
```

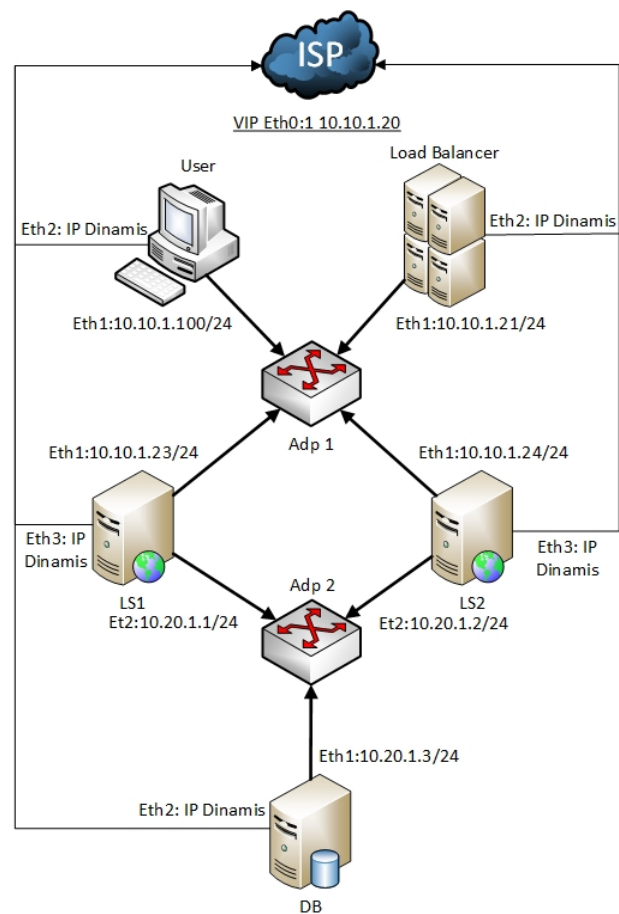
2. Konfigurasi *Node Direct Routing* pada Linux Server.

Pada setiap node linux server (LS1 dan LS2), jalankan perintah berikut pada terminal agar Virtual IP 10.10.1.20 dan kombinasi protokol yang dimaksudkan dilayani pada linux server:

```
# iptables -t nat -A PREROUTING -p tcp -d 10.10.1.20 -m multiport --dport 80,443 -j REDIRECT
```

Perintah di atas akan menyebabkan linux server memproses paket yang ditujukan untuk Virtual IP dan port yang diberikan kepada mereka. Kemudian pastikan

bahwa perubahan firewall disimpan agar setelah restart virtual masih tetap tersimpan.



Gbr. 4 Topologi uji coba jaringan

C. Perbandingan Performa *Load Balancing* Tanpa Menggunakan Algoritma, Menggunakan Algoritma *Round Robin*, dan Algoritma *Weighted Round Robin*

Web server diujicobakan sebanyak 20 kali dalam setiap skenario uji coba. Terdapat 5 skenario yaitu 100 request, 2500 request, 5000 request, 7500 request, dan 10000 request. Semua request dilakukan ujicoba dalam rentang waktu 5 detik. Parameter yang dirasa paling berpengaruh terhadap user yaitu, *Response Time*, *CPU Utilization*, *Throughput*, dan *request Error*. *Response Time* adalah waktu yang dibutuhkan untuk menyelesaikan satu request dan mengirimkannya kembali kepada client. *CPU Utilization* adalah jumlah sumber daya yang dibutuhkan untuk melakukan suatu proses komputerisasi. Topologi perancangan sistem menggunakan sejumlah server dengan spesifikasi yang sama. *Throughput* adalah jumlah data yang diterima dalam satuan waktu. *Request Error* adalah sejumlah request yang mengalami error pada saat penelitian. Pengujian dilakukan dengan perangkat lunak httpperf. Perangkat lunak ini dapat menciptakan banyak request dan koneksi dalam satu waktu. Parameter yang

digunakan pada pengujian menggunakan httpperf dilakukan dengan command pada terminal.

```
# httpperf --hog --server 10.10.1.20 --
num-conn 7500 --timeout 5 --rate 1500
--port 80
```

Nilai dari hasil penelitian tersebut kemudian dihitung rata-rata pada 20 kali penelitian dan dibagi jumlah total request yang diberikan sehingga didapatkan hasil rata-rata *Response Time*, *throughput* dan peluang terjadinya error pada setiap request. Sedangkan perhitungan *CPU Utilization load balancer* server dilakukan dengan cara menganalisa persentase nilai *CPU Utilization* yang diambil dari report *performance server* pada *CPU User*.

1. Analis Parameter *Response Error*

Pada tabel 3 adalah seluruh nilai rata-rata *Response Time* yang dihasilkan dari penelitian ini. implementasi *load balancing* server dengan metode *direct routing* menggunakan algoritma *weighted round robin* 2:3 lebih kecil dari *web server* yang tanpa menggunakan *load balancer* pada pengujian *Response Time* 100 request dan 2500 request. Sedangkan ketika pengujian *Response Time* pada 7500 request dan 10000 request tanpa *load balancer* lebih unggul dikarenakan banyaknya request yang error. *Request error* berpengaruh terhadap nilai *Response Time*.

Tabel 3 Rata-rata *Response Time* seluruh pengujian

Banyak Request	Response Time (millisecond)				
	Tanpa LB	RR	WRR 2:1	WRR 3:1	WRR 2:3
100	5.79	10.3	10.84	3.97	4.39
2500	2.36	2.1	1.94	2.26	1.72
5000	26.44	67.3	74.26	54.29	34.18
7500	352	438.8	420.88	389.70	357.61
10000	374.45	526.2	472.11	433.79	483.69

2. Analisa Parameter *CPU Usage*

Pada tabel 4 adalah seluruh nilai rata-rata *CPU Usage* yang dihasilkan dari penelitian ini. implementasi *load balancing* server dengan metode *direct routing* menggunakan algoritma *round robin* pada pengujian 7500 request dan 10000 request lebih kecil dari *web server* yang tanpa menggunakan *load balancer*. Hasil nilai pada pengujian 100 request, 2500 request, dan 5000 request cenderung stabil dikarenakan *web server* masih belum *overload*. Pengujian *CPU usage* pada algoritma *weighted round robin* 2 banding 3 mendapatkan hasil yang lebih baik kedua pada 7500 request dan 10000 request.

Tabel 4 Rata-rata *CPU Usage* seluruh pengujian

Req	CPU Usage (%)									
	Tanpa LB	RR S1	RR S2	WRR 2:1 S1	WRR 2:1 S2	WRR 3:1 S1	WRR 3:1 S2	WRR 2:3 S1	WRR 2:3 S2	
100	33.3	31.9	31.6	37.3	29.8	41.8	27.4	31.2	35.3	
2500	44.3	38.5	38	42.8	36	48.2	34	36.8	43.2	
5000	52.8	47.1	46.8	51	41.9	52.8	42	46	51.7	
7500	73.9	67.6	68	75.7	58.1	80.3	55.9	62.7	68.8	
10000	90.1	87.5	87.3	88.9	74	94.9	69.7	80	88.9	

3. Analisa Parameter *Troughput*

Pada tabel 5 adalah seluruh nilai rata-rata *throughput* yang dihasilkan dari penelitian ini. Implementasi *load balancing* server dengan metode *direct routing* menggunakan algoritma *weighted round robin* 2 banding 3 pada pengujian 7500 request dan 10000 request lebih besar dari *web server* yang tanpa menggunakan *load balancer* ataupun yang menggunakan algoritma lainnya. Hasil nilai pada pengujian 100 request, 2500 request, dan 5000 request cenderung stabil dikarenakan *web server* masih belum *overload*.

Tabel 5 Rata-rata *Throughput* seluruh pengujian

Banyak Request	Throughput (KB/s)				
	Tanpa LB	RR	WRR 2:1	WRR 3:1	WRR 2:3
100	12.5	12.5	12.5	12.5	12.5
2500	310.1	310.1	308.86	310	310.10
5000	615.2	615.2	566.74	594.72	608.75
7500	427.1	427.1	426.48	420.13	540.90
10000	406.2	406.2	398.64	369.47	444.02

4. Analisa Parameter *Request Error*

Pada tabel 6 adalah seluruh nilai rata-rata *request error* yang dihasilkan dari penelitian ini. implementasi *load balancing* server dengan metode *direct routing* menggunakan algoritma *weighted round robin* 2 banding 3 pada pengujian 7500 request dan 10000 request lebih kecil dari *web server* yang tanpa menggunakan *load balancer* ataupun yang menggunakan algoritma lainnya. Hasil nilai pada pengujian 100 request, 2500 request, dan 5000 request adalah stabil dikarenakan *web server* masih belum *overload*.

Tabel 6 Rata-rata *Throughput* seluruh pengujian

Banyak Request	Request Error				
	Tanpa LB	RR	WRR 2:1	WRR 3:1	WRR 2:3
100	0	0	0	0	0
2500	0	0	0	0	0
5000	0	0	0	0	0
7500	2209	1752	2209	2334	1223
10000	4693	4492	4513	4772	4349

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari keseluruhan proses dan pembahasan penelitian yang telah dijelaskan pada bab sebelumnya adalah sebagai berikut:

1. BPBD Jatim memiliki portal *website* sebagai sumber informasi kejadian bencana yang ditujukan untuk melayani masyarakat. *Website* tersebut belum menggunakan metode *Load Balancing* dan sudah aktif sejak 2017. Dari waktu ke waktu hingga 2019 saat ini kondisi beban server pada Badan Penanggulangan Bencana Daerah Provinsi Jawa Timur selalu meningkat setiap bulannya.
2. Pemasangan *Load Balancing* metode *LVS Direct Routing* menggunakan *Piranha* dapat diimplementasikan secara virtual server atau real server dengan menggunakan beberapa aplikasi pelengkap. *Load balancing* tersebut mudah digunakan dan dapat menyesuaikan algoritma yang ingin digunakan seperti *Round Robin* atau *Weighted Round Robin*.
3. Persentase pemakaian CPU pada *load balancing LVS direct routing* menggunakan algoritma *weighted round robin 2* banding 3 ketika 7500 request dan 10000 request per 5 detik lebih stabil dari semua skenario pengujian. Besarnya nilai *throughput* pada *load balancing LVS direct routing* menggunakan algoritma *weighted round robin 2* banding 3 ketika 7500 request dan 10000 request per 5 detik lebih tinggi dari pada skenario pengujian lainnya. Kecilnya nilai request error ketika pengujian 7500 request dan 10000 request menggunakan *load balancing LVS direct routing* menggunakan algoritma *weighted round robin 2* banding 3 mendapatkan nilai request error yang paling sedikit daripada skenario pengujian lainnya, menjadikan algoritma *weighted round robin 2* banding 3 pilihan terbaik ketika ingin mengatasi request yang error.

UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya kepada Allah SWT yang telah memberikan pertolongannya dalam setiap langkah pengerjaan penelitian ini. Terimakasih pula untuk semua pihak yang telah memberikan dukungan sehingga penelitian ini dapat berjalan dan terselesaikan dengan baik.

REFERENSI

- [1] Afrianto, Y., Sukoco, H., & Wahjuni, S. (2018). Weighted Round Robin Load Balancer to Enhance Web Server Cluster in OpenFlow Networks. TELKOMNIKA, Vol. 16 No. 3.
- [2] Aggarwal, R., & Gupta, L. (2017). Load Balancing in Cloud Computing. International Journal of Computer Science and Mobile Computing, 180–186.
- [3] Ansharullah, K. (2016). Implementasi Sistem Load Balancing dengan Algoritma Round Robin untuk Mengatasi Beban Server di SMK Negeri 2 Kudus. Semarang: Universitas Negeri Semarang.
- [4] Arta, Y. (2016). Implementasi Computer Cluster Berbasis Open Source untuk menyeimbangkan Beban Sistem dan Jaringan Komputer. Jurnal Teknik Informatika dan Sistem Informasi, Vol. 2 No. 1.
- [5] Aziz, A., Febrianto, T., & Saptono, R. (2014). Analisa Load Balancing Server dengan Metode LVS Direct Routing Menggunakan Algoritma Round Robin dan Least Connection. Seminar Ilmiah Ilmu Komputer (pp. 1-13). Bogor: Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor.
- [6] Devi, D. C., & Uthariaraj, V. R. (2016). Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks. The Scientific World Journal Hindawi Publishing Corporation, 14.
- [7] Han, Z., & Pan, Q. (2012). LVS Cluster Technology in the Research and Application of State-owned Asset. The 2nd International Conference on Computer Application and System Modeling, 1-4.
- [8] Irza, I. F., Zuhendra, & Efrizon. (2017). Analisis Perbandingan Kinerja Web Server Apache dan Nginx Menggunakan Httperf Pada Portal Berita. Jurnal Vokasional Teknik Elektronika & Informatika, Vol. 5 No. 2.
- [9] Kumari, P., & Saxena, M. (2015). A Round-Robin based Load balancing approach for Scalable demands and maximized Resource availability. International Journal Of Engineering And Computer Science, Vol. 5.
- [10] Load Balancing. (2019). Educative. <https://www.educative.io>. Retrieved from <https://www.educative.io>
- [11] Lukitasari, D., & Oklilas, A. F. (2010). Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web server Cluster Menggunakan Linux Virtual Server. JURNAL GENERIC, Vol. 5 No. 2.
- [12] Nasser, H., & Witono, T. (2016). Analisis Algoritma Round Robin, Least Connection, dan Ratio pada Load Balancing Menggunakan OPNET Modeler. Bandung: Universitas Kristen Maranatha.
- [13] Pamungkas, E. W., & Putri, D. G. (2015). Load Balancing pada Cloud Computing dengan Sumber Daya Terbatas Menggunakan Penggabungan Algoritma ESCE dan Throttled. Surabaya: Institut Teknologi Sepuluh Nopember.
- [14] Perdana, F. P., Irawan, B., & Latuconsina, R. (2017). ANALISIS PERFORMANSI LOAD BALANCING DENGAN ALGORITMA WEIGHTED ROUND ROBIN PADA SOFTWARE DEFINED NETWORK (SDN). e-Proceeding of Engineering, 4161.
- [15] Rabu, J. A., Purwadi, J., & Raharjo, W. S. (2012). Implementasi Load Balancing Web Server Menggunakan Metode LVS-NAT. INFORMATIKA, Vol. 8 No. 2.
- [16] Rahmatulloh, A., & Firmansyah, M. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi. Jurnal Teknologi dan Sistem Informasi, Vol 3.
- [17] Sitorius, M. (2016). Pemodelan Private Cloud Computing Dengan Metode Software As A Service (Saas). Jakarta: Universitas Satya Negara Indonesia.
- [18] Syaikh, A. (2010). Komputasi Awan (Cloud Computing) Perpustakaan Pertanian. Jurnal Pustakawan Indonesia, Vol. 10 No. 1.
- [19] Triono, G. (2015). Implementasi Load Balancing dengan Menggunakan Algoritma Round Robin pada Kasus Pendaftaran Siswa Baru Sekolah Menengah Pertama LAB School UNESA Surabaya. Seminar Nasional "Inovasi dalam Desain dan Teknologi", 169-176.
- [20] Wibowo, A., Virgono, A. M., & Latuconstina, R. (2018). Load Balancing on Cloud Computing Using Least Connection Method. e-Proceeding of Engineering, 6210.