

Analisa Mirror Server menggunakan Load Balance pada Jaringan Area Lokal

Alfian Nurdiansyah¹, Nugroho Suharto², Hudiono³

^{1,3} Program Studi Jaringan Telekomunikasi Digital,
Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

² Program Studi Teknik Telekomunikasi,
Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

¹alfiannn13@gmail.com, ²nugroho.suharto@polinema.ac.id, ³hudiono@polinema.ac.id

Abstract— A Server is a system that provides certain services on a computer network. The Server has its own operating system called the network operating system. The Server also controls all access to existing networks. In order to help server tasks, a server mirroring system is set up where the server is duplicating a data sets or an exact clone of a server providing various information. Mirror server or also called Server synchronization is a duplicate of a server. To add performance from the server, a load balancer is required. Load balancing is a technique for distributing two internet connection paths equally. With the application of load balancing the traffic will run more optimally, maximizing throughput, and avoiding overload on the connection path. Iptables is used for IP filtering so the clients could access the server according to the closest server zone, so that the load balance combined with iptables can reduce the server's load. A common problem is when many clients access a server and the server will overload and cause the server performance to be slow because of the heavy traffic. The access Client also get the effect of this, which is slow response time. From the result of research on the combination of load balance and iptables, it was found that the load balance with the round robin algorithm the average delay obtained for server1 was 0.149 seconds and 0.19122. Server2 has an average delay of 0.161 seconds and 0.012 seconds.

Keywords— Load balancing, iptables, mirror server, filter, zone.

Abstrak— Server merupakan sebuah sistem yang memberikan layanan tertentu pada suatu jaringan komputer. Server mempunyai sistem operasi sendiri yang disebut sistem operasi jaringan. Server juga mengontrol semua akses terhadap jaringan yang ada didalamnya. Agar membantu tugas server, dibuatlah sistem *mirroring* server dimana server tersebut menduplikasi sebuah data set atau tiruan persis dari sebuah server yang menyediakan berbagai informasi. Mirror server atau disebut juga sinkronisasi server merupakan duplikat dari suatu server. Untuk menambah kinerja dari server maka dibutuhkan *load balancer*. *Load balancing* adalah teknik untuk mendistribusikan internet dua jalur koneksi secara seimbang. Dengan penerapan *load balancing* trafik akan berjalan lebih optimal, memaksimalkan *throughput* dan menghindari *overload* pada jalur koneksi. Iptables digunakan untuk memfilter IP sehingga *client* mengakses server sesuai dengan zona server yang paling dekat. Sehingga *load balance* yang dipadukan dengan iptables dapat membuat kinerja server menjadi lebih ringan. Masalah yang sering terjadi adalah ketika banyaknya *client* yang mengakses sebuah server maka server akan overload dan mengakibatkan kinerja server menjadi berat karena padatnya trafik. *Client* yang mengakses juga mendapatkan efek dari hal tersebut yaitu akses yang lama. Dari hasil penelitian tentang perpaduan antara *load balance* dan iptables didapati bahwa *load balance* dengan algoritma round robin rata-rata delay yang didapatkan untuk server1 yaitu 0,149 detik dan 0,19122. Server2 rata-rata delay yang didapatkan 0,161 detik dan 0,012 detik.

Kata kunci— Load Balancing, iptables, mirror server, filter, zone

I. PENDAHULUAN

Server merupakan sebuah sistem yang memberikan layanan tertentu pada suatu jaringan komputer. Server mempunyai sistem operasi sendiri yang disebut sistem operasi jaringan. Server juga mengontrol semua akses terhadap jaringan yang ada didalamnya. Server merupakan sistem yang penting untuk suatu jaringan komputer dikarenakan server memiliki sistem backup data yang baik, keamanan relatif lebih tinggi. Untuk membantu kinerja server maka dibutuhkan *mirror* server sebagai penyedia layanan dan *load balance* sebagai pembagi beban. Mirror server atau disebut juga sinkronisasi server merupakan duplikat dari suatu server yang berbeda dengan server

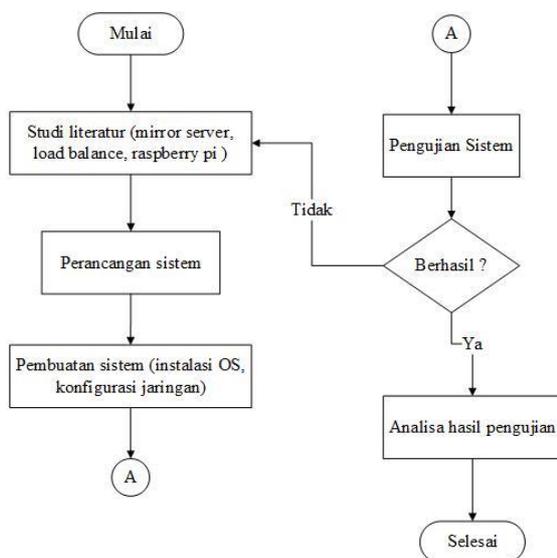
utama. Fungsi mirror adalah untuk mengurangi kemacetan data pada suatu situs. Server utama merupakan server yang paling akurat dan mirror bertugas untuk menyalin isi server utama tersebut [1]. Load Balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan throughput, memperkecil waktu tanggap dan menghindari overload pada salah satu jalur koneksi [2].

Permasalahan yang sering terjadi adalah ketika banyaknya client yang mengakses sebuah server maka server akan overload dan mengakibatkan kinerja server menjadi berat karena padatnya trafik. Client yang

mengakses juga mendapatkan efek dari hal tersebut yaitu akses yang lama. *Mirror server* akan mempermudah pekerjaan server utama karena *client* tidak hanya menuju server utama tetapi dapat mengakses ke server lainnya ditambah dengan *load balance* maka kinerja server akan lebih mudah [3]. Kemudian *load balance* akan dikombinasikan dengan *iptables* dimana bekerja sebagai *filter IP* yang mengakses ke server. *Iptables* akan memfilter IP yang mengakses apakah sesuai dengan zona server yang ada. Hal tersebut juga akan mempermudah kerja server karena jika IP tidak sesuai maka akan ditolak.

II. METODE

Tahapan Penelitian yang akan diambil dalam penentuan langkah selanjutnya dari penyusunan penelitian ini adalah sebagai berikut:



Gambar 1. Tahapan penelitian

- Studi Literatur

Studi Literatur pada penelitian ini yaitu pengumpulan informasi yang bersumber pada jurnal, artikel, dan buku mengenai perancangan sistem pengering serta mencari tahu kelebihan dan kekurangan dari alat dan bahan yang akan digunakan pada sistem.

- Perancangan Sistem

Perancangan sistem pada penelitian ini meliputi perancangan untuk metode load balancer yang digunakan, perancangan topologi.

- Pembuatan Sistem

Pembuatan sistem merupakan realisasi dari proses perancangan sistem load balancer, instalasi server (Web, DNS), instalasi sistem operasi yang digunakan

- Pengujian Sistem

Pengujian dilakukan dengan menjalankan sistem, apakah sistem dapat bekerja dengan baik jika terjadi kemacetan dan dapat menguraikannya dengan berbagai jalur yang ada.

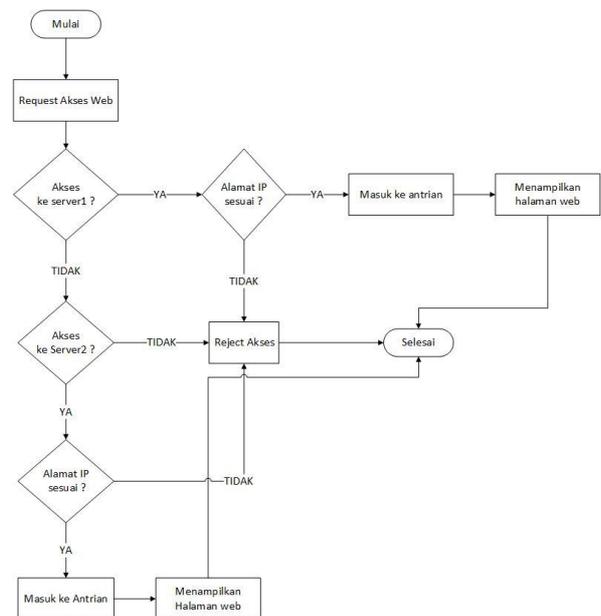
- Pengambilan Data

Data yang akan diambil pada penelitian ini meliputi bandwidth dan delay.

- Analisa Hasil dan pengolahan data

Analisa hasil pengujian sistem dilakukan ketika telah mendapatkan hasil data saat simulasi dan pada kenyataannya dapat berjalan dengan baik. Mengambil Kesimpulan dari penelitian ini yang merupakan jawaban atas masalah yang telah dirumuskan serta saran didapatkan dari melihat kekurangan pada sistem sehingga untuk penelitian berikutnya dapat diperbaiki maupun dikembangkan sesuai dengan kebutuhan dan perkembangan teknologi. Pembuatan laporan dilakukan sebagai bukti bahwa telah dilakukan sebuah penelitian.

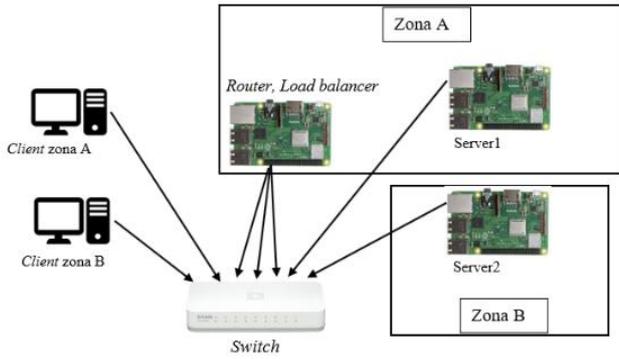
A. Perencanaan Sistem



Gambar 2. Diagram alir sistem

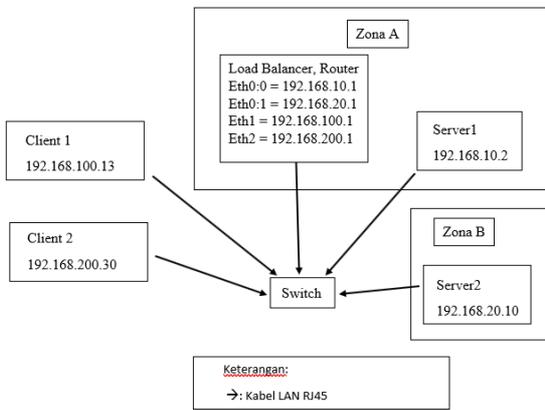
Diagram alir sistem pada penelitian ini pertama yaitu *client* akan *request* untuk akses web apakah meminta ke server 1 jika ya maka akan dilakukan pengecekan lagi apakah IP sesuai dengan zona untuk server1 jika sudah sesuai maka dapat masuk ke dalam antrian dan dapat mengakses halaman web jika alamat IP tidak sesuai maka *request* akan ditolak. Kemudian jika akses ke server2 apakah sesuai dengan alamat IP yang diperbolehkan, jika ya maka akan masuk ke antrian dan dapat mengakses halaman web tetapi jika tidak maka sistem akan menolak *client* tersebut. Antrian disini diatur oleh *load balancer* yang sudah terkonfigurasi pada raspberry pi.

B. Diagram Blok Perangkat



Gambar 3. Sambungan perangkat

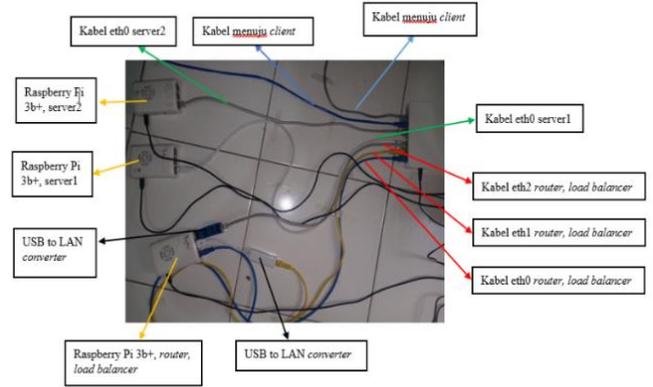
Semua perangkat terhubung melalui switch menggunakan kabel ethernet. Masing-masing kabel memiliki alamat IP dengan berbeda network. Satu raspberry pi berperan sebagai load balancer, dan router. Raspberry pi hanya mempunyai satu interface ethernet maka diperlukan USB to LAN converter pada bagian router untuk menambah interface ethernet. Pada router terdapat 4 interface yang digunakan yaitu eth:0:0, eth0:1, eth1, dan eth2, sehingga terdapat 4 network yang berbeda. Kemudian letak dari server dibuat berbeda zona, webserver1 terletak pada zona A, dan webserver2 terletak pada zona B. Kemudian pada sisi kiri terdapat client yang akan mengakses web, pada client juga dibagi menjadi network yang berbeda. Untuk pengujian pada sisi client juga akan menggunakan virtual client. Berikut merupakan pembagian ip pada masing – masing perangkat:



Gambar 4. Pembagian alamat IP

III. HASIL DAN PEMBAHASAN

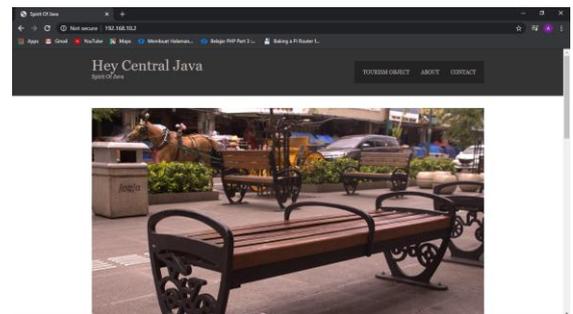
A. Hasil Implementasi Alat



Gambar 5. Hasil implementasi alat

Penambahan interface pada router dikarenakan jumlah interface ethernet yang dibutuhkan yaitu 3. Semua perangkat terhubung melalui switch menggunakan kabel ethernet RJ45. Pada router eth0 dibagi menjadi 2 alamat IP menggunakan IP alias. IP alias yaitu ketika kita membuat 1 interface mempunyai lebih dari 1 IP dengan mendefinisikan seperti eth0:0, eth0:1. Sedangkan untuk server sendiri tidak membutuhkan interface tambahan karena hanya menggunakan 1 interface saja.

B. Hasil implementasi Web



Gambar 6. Tampilan utama halaman web

Gambar 8 adalah halaman utama web yang diakses oleh client. Letak web tersebut berada di server 1 maupun server 2.

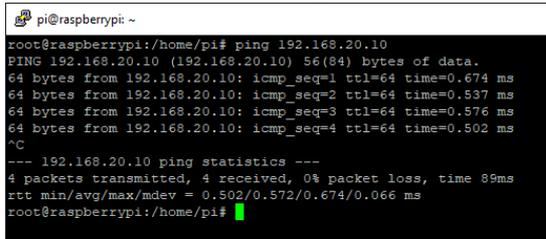
C. Pengujian Alat

1. Cek koneksi router ke server

```

pi@raspberrypi: ~
root@raspberrypi:~/home/pi# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data:
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.664 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.525 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.572 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.537 ms
^C
--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 90ms
rtt min/avg/max/mdev = 0.525/0.574/0.664/0.059 ms
root@raspberrypi:~/home/pi#
    
```

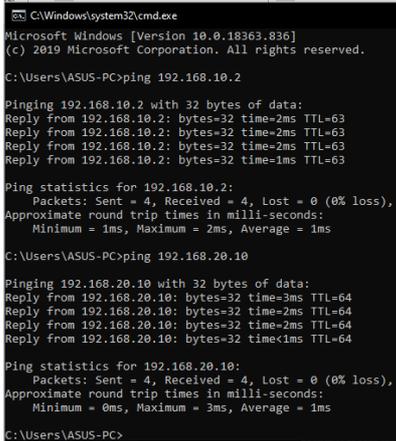
Gambar 7. Koneksi router ke server1



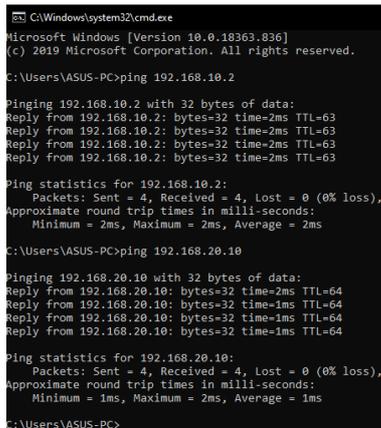
Gambar 8. Koneksi router ke server2

Cek koneksi dari router ke server yang ada berhasil sehingga dapat digunakan.

2. Cek koneksi client ke server



Gambar 9. Koneksi client ke server

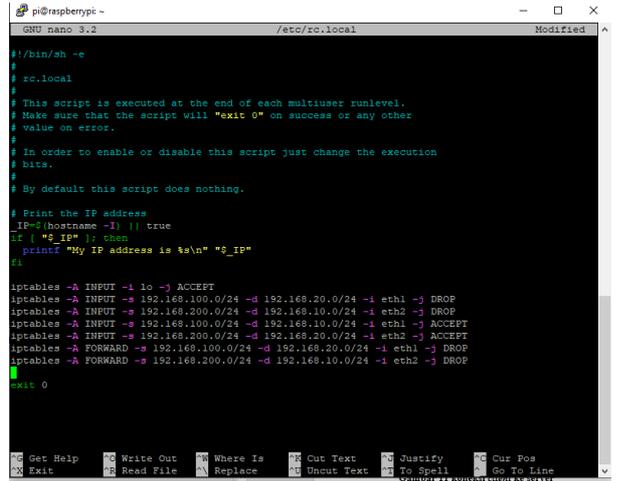


Gambar 10. Koneksi client 2 ke server

Pengecekan koneksi dari client ke server berhasil sehingga client sudah dapat mengakses server dengan baik.

3. Konfigurasi Iptables

Konfigurasi iptables digunakan untuk memfilter IP yang dapat mengakses server, karena yang digunakan adalah berbasis zonasi maka jika alamat IP tidak sesuai dengan zona maka akan ditolak [4]. Berikut adalah konfigurasi iptables yang diletakkan pada “/etc/rc.local”



Gambar 11. Konfigurasi iptables

D. Pengujian Delay dan Throughput Pengiriman Data ke server

TABEL I
HASIL DELAY VIRTUAL CLIENT KE SERVER1

No.	Source	Destination	Time delta from previous displayed (detik)
1.	192.168.10.2	192.168.100.7	0
2.	192.168.10.2	192.168.100.7	0.521020000
3.	192.168.10.2	192.168.100.7	0.006792000
4.	192.168.10.2	192.168.100.7	0.919708000
5.	192.168.10.2	192.168.100.7	0.000833000
6.	192.168.10.2	192.168.100.7	0.000819000
7.	192.168.10.2	192.168.100.7	0.000000000
8.	192.168.10.2	192.168.100.7	0.000851000
9.	192.168.10.2	192.168.100.7	0.022690000
10.	192.168.10.2	192.168.100.7	0.018721000
Rata-rata			0.149

TABEL 2
HASIL DELAY VIRTUAL CLIENT KE SERVER2

No.	Source	Destination	Time delta from previous displayed (detik)
1.	192.168.20.10	192.168.200.4	0
2.	192.168.20.10	192.168.200.4	0.291770000
3.	192.168.20.10	192.168.200.4	0.008567000
4.	192.168.20.10	192.168.200.4	0.934615000
5.	192.168.20.10	192.168.200.4	0.271367000
6.	192.168.20.10	192.168.200.4	0.007762000
7.	192.168.20.10	192.168.200.4	0.005511000
8.	192.168.20.10	192.168.200.4	0.016989000
9.	192.168.20.10	192.168.200.4	0.063340000
10.	192.168.20.10	192.168.200.4	0.005467000
Rata-rata			0.161

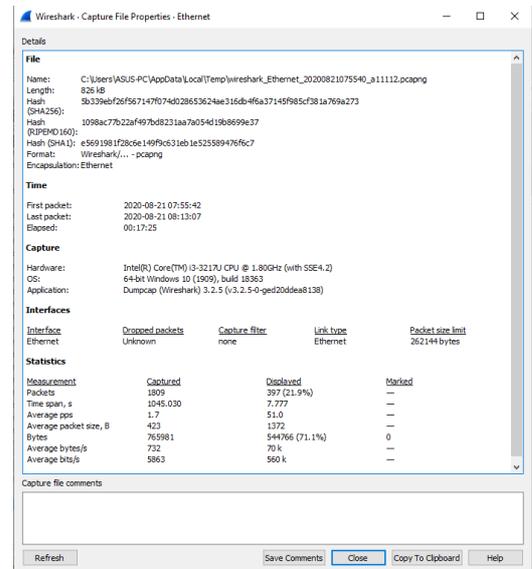
TABEL 3
HASIL DELAY REAL CLIENT KE SERVER1

No.	Source	Destination	Time delta from previous displayed (detik)
1.	192.168.10.2	192.168.100.13	0
2.	192.168.10.2	192.168.100.13	0.000982000
3.	192.168.10.2	192.168.100.13	1.109805000
4.	192.168.10.2	192.168.100.13	0.000902
5.	192.168.10.2	192.168.100.13	0.00622
6.	192.168.10.2	192.168.100.13	0.456796
7.	192.168.10.2	192.168.100.13	0.100758
8.	192.168.10.2	192.168.100.13	0.225318
9.	192.168.10.2	192.168.100.13	0.001866
10.	192.168.10.2	192.168.100.13	0.00963
Rata-rata			0.19222

TABEL 4
HASIL DELAY REAL CLIENT KE SERVER2

No.	Source	Destination	Time delta from previous displayed (detik)
1.	192.168.20.10	192.168.200.13	0
2.	192.168.20.10	192.168.200.13	0.000959000
3.	192.168.20.10	192.168.200.13	0.004422000
4.	192.168.20.10	192.168.200.13	0.007065000
5.	192.168.20.10	192.168.200.13	0.085592000
6.	192.168.20.10	192.168.200.13	0.005950000
7.	192.168.20.10	192.168.200.13	0.000882000
8.	192.168.20.10	192.168.200.13	0.009290000
9.	192.168.20.10	192.168.200.13	0.001661000
10.	192.168.20.10	192.168.200.13	0.002737000
Rata-rata			0.012

Pada tabel 1 – 3 adalah rata-rata delay yang terjadi pada server. *Virtual client* mengakses ke server1 dan server2. Dengan rata-rata *delay virtual client* ke server 1 sebesar 0.149 yang termasuk ke dalam kategori *excellent* sesuai standar ITU-T tentang *delay*. Kemudian untuk *virtual client* ke server2 rata-rata *delay* yang didapatkan sebesar 0,161 detik dimana termasuk dalam kategori *good* pada standar *delay* dari ITU-T. Kemudian untuk dari *real client* ke server 1 rata-rata *delay* sebesar 0.1922 detik dimana termasuk ke dalam kategori *good* pada standar *delay*. Pada Tabel 4 hasil *delay* dari real client ke server2 menunjukkan rata-rata *delay* yang diperoleh sebesar 0.012 dan termasuk kategori *excellent* sesuai standar *delay* dari ITU-T.



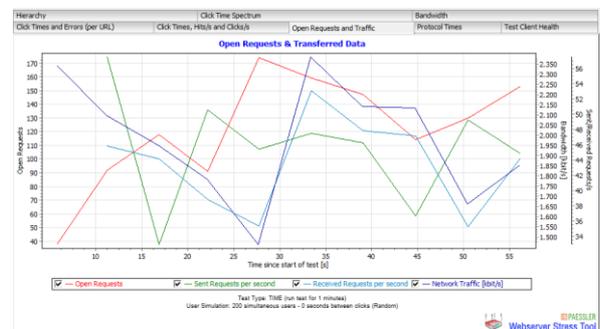
Gambar 12. Hasil capture throughput

E. Pengujian trafik menggunakan Paessler webserver stress tool

TABEL 5
HASIL PENGUJIAN UNTUK PER URL SERVER1

No.	Clicks	Errors	Errors (%)	Time spent (ms)	Avg. click time (ms)
1.	2455	0	0	4.769.395	1.943

Dari pengujian menggunakan webserver *stress tool* untuk didapatkan data *click* sebesar 2455, untuk waktu sebesar 4769395ms dengan *error* sebesar 0% , kemudian untuk rata-rata waktu *click* sebesar 1943ms



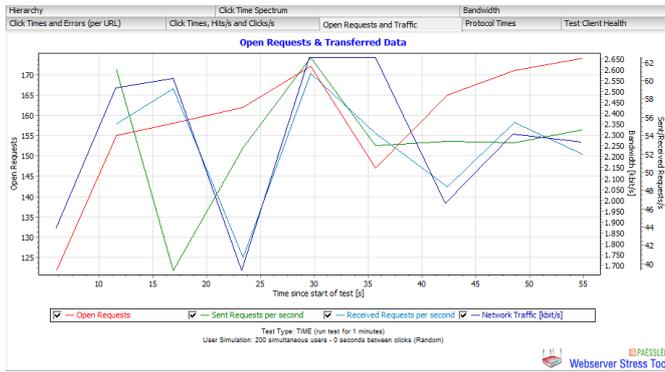
Gambar 13. Grafik trafik dan request pada server 1

Pada gambar 13 merupakan grafik trafik yang ada pada server 1. Trafik paling padat pada jaringan terjadi pada detik antara 33 dan 34 setelah tes dimulai.

TABEL 6
HASIL PENGUJIAN UNTUK PER URL SERVER2

No.	Clicks	Errors	Errors (%)	Time spent (ms)	Avg. click time (ms)
1.	2.761	28	1.01	5.097.892	1.865

Dari pengujian menggunakan *webservice stress tool* untuk didapatkan data *click* sebesar 2761, untuk waktu sebesar 5097892ms dengan *error* sebesar 1.01% , kemudian untuk rata-rata waktu *click* sebesar 1865ms,



Gambar 14. Grafik trafik dan *request* pada server 2

Pada Pada gambar 14 merupakan grafik trafik yang ada pada server 1. Trafik paling padat pada jaringan terjadi pada detik 29 sampai 35 setelah tes dimulai.

IV. KESIMPULAN

1. Pengimplementasian *load balance* pada Raspberry Pi dapat menggunakan aplikasi HAProxy untuk penggunaan pada jaringan area lokal lebih efektif karena telah terdapat *stats* yang dapat digunakan tanpa perlu adanya koneksi internet
2. Server 1 mempunyai rata-rata delay sebesar 0.149 detik atau 149ms untuk server1 ke *virtual client* berdasarkan standar dari ITU-T G.1010 termasuk ke kategori *excellent*. Untuk *real client* ke server1 didapatkan dengan 0.191 detik atau 192 ms dan termasuk dalam kategori *good*. Server2 mempunyai rata-rata delay sebesar 0.161 detik atau 161 ms untuk server2 ke *virtual client* berdasarkan standar ITU-T G.1010 termasuk kategori *good*. Untuk *real client* ke server2 rata-rata delay sebesar 0.012 detik atau 12 ms dan termasuk dalam kategori *excellent*
3. Kinerja server menjadi lebih ringan karena adanya kombinasi dari *load balance* dan *iptables* untuk filter. *Client* mengakses berdasarkan dari zona atau server yang paling setelah itu diantrikan. Sehingga lebih efektif karena rute yang dilewati tidak jauh.

REFERENSI

- [1] E. Wahyu and A. Rahmatulloh, "Optimasi Server SIMAK Menggunakan Memcached dan Mirror Server Untuk Meningkatkan Kecepatan Akses Layanan Akademik Universitas Siliwangi," pp. 69–78.
- [2] R. Dawood, S. F. Qiana, and S. Muchallil, "Kelayakan Raspberry Pi sebagai Web Server: Perbandingan Kinerja Nginx, Apache, dan Lighttpd pada Platform Raspberry Pi," *J. Rekayasa Elektr.*, vol. 11, no. 1, pp. 25–29, 2014, doi: 10.17529/jre.v11i1.1992.

- [3] Y. Arta, "Penerapan Metode Round Robin Pada Jaringan Multihoming Di Computer Cluster," *It J. Res. Dev.*, vol. 1, no. 2, pp. 26–35, 2017, doi: 10.25299/ijrd.2017.vol1(2).677.
- [4] T. Novianti and A. Widiatoro, "Analisa QOS (Quality of Services) pada Implementasi IPV4 dan IPV6 dengan Teknik Tunneling," *Rekayasa*, vol. 9, no. 2, p. 76, 2017, doi: 10.21107/rys.v9i2.334.