

ANALISIS KINERJA PENYEIMBANG BEBAN SERVER MENGUNAKAN ALGORITMA *LEAST CONNECTION* UNTUK LAYANAN TRIPLE PLAY

ANALYSIS PERFORMANCE LOAD BALANCING ON SERVER USING *LEAST CONNECTION* ALGORITHM FOR TRIPLE PLAY SERVICE

Fidel Alessandro Metekohy¹, Tengku Ahmad Riza, S.T., M.T.², Asep Mulyana, S.T., M.T.³

¹ Prodi D3 Teknologi Telekomunikasi, ²Fakultas Ilmu Terapan, ³Universitas Telkom

Jln. Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia

fidelmetekohy02@gmail.com,

tengkuriza@telkomuniversity.ac.id

@asepmulyana@telkomuniversity.ac.id

ABSTRAK

Aplikasi internet yang paling banyak dikunjungi adalah *website*. konten yang ada pada *website* yang bervariasi sehingga membuat masyarakat sangat tertarik untuk terus membuka suatu *website*. Pertumbuhan manusia semakin besar yang mengakibatkan pengguna aplikasi internet semakin bertambah, hal ini dapat menyebabkan trafik pada *website* meningkat dan beban kerja pada server bertambah untuk melayani request dari pengguna. Sedangkan pengguna ingin mengakses sebuah *website* dengan kecepatan yang maksimal.

Oleh karena itu, penyeimbang beban atau *load balancing* adalah solusinya karena dapat digunakan untuk peningkatan kinerja sebuah server dengan tujuan agar server tidak mengalami *overload request*.

Pada proyek akhir ini telah diimplementasikan sistem penyeimbang beban dengan menggunakan algoritma *Least Connection* terhadap tiga layanan yaitu web server, FTP server, dan VoIP server. Didapatkan nilai rata-rata dengan koneksi antara *client* dan *load balancer* menggunakan *wireline* yaitu pada web server untuk *throughput* sebesar 105,03 KB/s, *response time* 6,501 detik, *request loss* sebesar 1,97%. Pada FTP server untuk *throughput* sebesar 120,70 KB/s, *response time* 33,97 detik, *request loss* sebesar 0,39% dan pada VoIP server untuk nilai *block call* sebesar 0,9%. Sedangkan dengan koneksi antara *client* dan *load balancer* menggunakan *wireless* yaitu pada web server untuk *throughput* sebesar 108,38 KB/s, *response time* 7,52 detik, *request loss* sebesar 2,07%. Pada FTP server untuk *throughput* sebesar 122,64 KB/s, *response time* 33,19 detik, *request loss* sebesar 0,27% dan pada VoIP server untuk nilai *block call* sebesar 0,32%.

Kata kunci: *load balancing, least connection, server.*

ABSTRACT

The most visited internet application is a website. the content on a the website varies so that people are very interested in continuing to open a website. The greater human growth that results in increased internet application users, this can cause traffic on the website to increase and the workload on servers increases to serve requests from users. While users want to access the website with maximum speed.

Therefore, load balancing is the solution because it can be used to improve server performance with the aim that the server does not experience overload requests..

In this final project, load balancing system was implemented using the least connection algorithm for three types of services, namely a web server, a FTP server, and a VoIP server. Obtained an average value with the connection between the client and the load balancer using wireline that is on the web server for throughput of 105,03 KB/s, response time of 6.501 seconds, request loss of 1.97%. On FTP server for throughput of 120,70 KB/s, response time of 33,97 seconds, request loss of 0,39% and on VoIP servers for block call values of 0,9%. Whereas with the connection between the client and the load balancer using wireless that is on the web server for throughput of 108,38 KB/s, response time of 7,52 seconds, the request loss of 2,07%. On FTP server for throughput of 122,64 KB/s. response time of 33,19 seconds, request loss of 0,27% and on VoIP servers for block call values of 0,32%.

Keywords: *load balancing, least connection, server.*

1. Pendahuluan

Di era digital saat ini, penggunaan internet sudah menjadi kebutuhan bagi kehidupan masyarakat. melalui internet masyarakat bisa mendapatkan informasi terbaru dari negara-negara di eropa bahkan amerika. Salah satu aplikasi internet yang sering diakses setiap hari adalah *website*. *website* memiliki sangat banyak kegunaan untuk mencari referensi, media sosial, *blogging*, *commerce*, dll. Meningkatnya penggunaan *website* mengakibatkan beban kerja pada server ikut meningkat dengan cepat sehingga server menjadi kelebihan beban dalam waktu yang singkat. masalah lain yang muncul juga, ketika penyedia jasa web hanya bergantung pada server tunggal.

Berdasarkan penelitian sebelumnya yang berjudul “Analisis Kinerja Penyeimbang Beban Server Web dengan menggunakan Algoritme Rasio Dinamis” yang membahas tentang *Load Balancer* yang dapat membaca ketersediaan *resource* pada server sehingga pembagian *request* dilakukan dengan mempertimbangkan *resource server* seperti CPU, *memory*, dan *disk* [4] , dan penelitian dengan judul “ Analisa Kinerja Load Balancing Menggunakan Linux Virutal Server pada Layanan HTTP” dengan pembahasan tentang kinerja dari web server dengan menggunakan Linux Virtual Server Load Balancer. Parameter yang digunakan sebagai pedoman dalam menganalisis adalah *throughput*, *request/sec*, dan *request loss* [3].

Pada penelitian proyek akhir ini yang berjudul “ Analisis Kinerja Penyeimbang Beban Server menggunakan Algoritma *Least Connection* Untuk Layanan Triple Play” akan dibangun suatu perancangan sistem penyeimbang beban yang dapat menghindari terjadinya *overload* pada sebuah server. Terdapat tiga layanan yaitu web server, FTP server dan VoIP server sehingga penulis akan menganalisis web server, FTP server, dan VoIP server dengan menggunakan penyeimbang beban F5 BIG-IP LTM serta algoritma yang digunakan adalah *Least Connection* dan parameter yang diperhatikan untuk menganalisis adalah *throughput*, *response time*, dan *request loss*.

2. Dasar Teori

2.1. Penyeimbang Beban

Load balancing Server (SLB) dapat diartikan sebagai proses pendistribusian trafik dari suatu alamat ke dalam beberapa server menggunakan sebuah perangkat jaringan. Perangkat ini bertugas mengalihkan trafik yang ditujukan untuk sebuah alamat ke server berbeda-beda. Proses load balancing ini benar-benar transparan bagi user. Biasanya terdapat puluhan atau bahkan ratusan server yang bekerja pada suatu URL.[1]

2.2. Web Server

Web server adalah perangkat lunak yang menjadi tulang belakang dari world wide web (www). Web server menunggu permintaan dari client yang menggunakan browser seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar, disebut dengan format standar general markup language (SGML). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser tersebut [2].

2.3. FTP Server

FTP atau *File Transfer Protocol* merupakan suatu aplikasi yang dapat melakukan pertukaran transfer data antara dua computer. Dua computer tersebut adalah *client* dan *server* . pertukaran data bisa terjadi pada computer dengan menggunakan jaringan local ataupun melalui internet. Terdapat dua buah port pada saat koneksi FTP terjadi, yaitu *port 20* dan *port 21*. Kedua *port* tersebut berjalan melalui TCP bukan UDP. *Port 21* akan didengar oleh FTP *server* untuk koneksi yang akan datang dari FTP *client*. *Port 20* dan *port 21* masing-masing berfungsi sebagai data *port* dan *command port*.

2.4. Session Initiation Protocol (SIP)

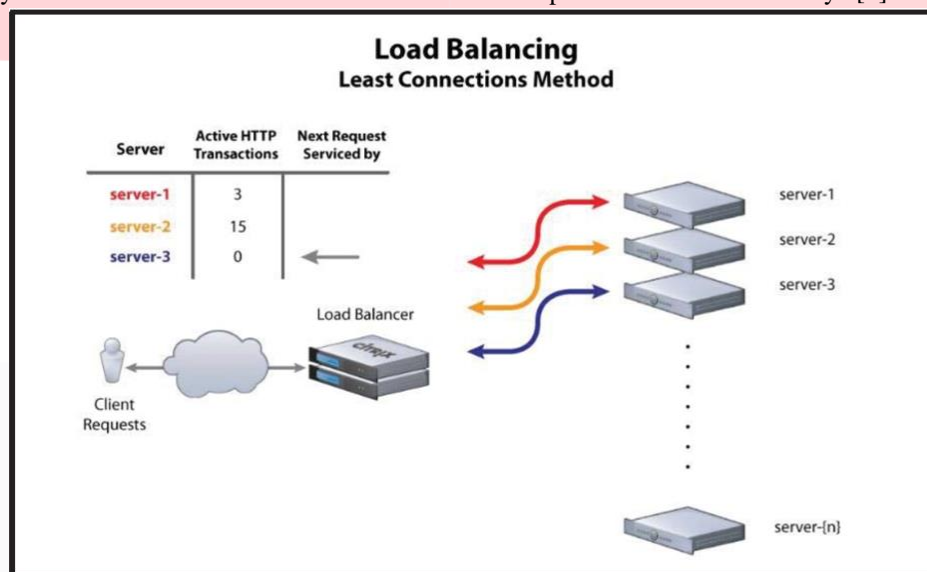
SIP merupakan salah satu protocol yang bekerja pada layer aplikasi. SIP dapat melakukan proses awal, modifikasi, dan pemutusan pada sesi multimedia. SIP dapat dikatakan memiliki karakteristik sama dengan FTP server yaitu *client-server*, yang artinya *request* akan diberikan oleh *client* dan dikirimkan ke server. Kemudian, server mengolah *request* tersebut dan memberikan tanggapan terhadap *request* ke *client*. Proses

itu disebut transaksi SIP. Selain itu, SIP juga disebut sebagai protocol yang *text-based* seperti HTTP dan SMTP.

2.5. Algoritma *Least Connection*

Least connection merupakan algoritme penjadwalan yang mengarahkan koneksi jaringan pada server aktif dengan jumlah koneksi yang paling sedikit. Penjadwalan ini termasuk salah satu algoritme penjadwalan dinamik, karena memerlukan perhitungan koneksi aktif untuk masing-masing real server secara dinamik. Metode penjadwalan ini baik digunakan untuk melancarkan pendistribusian ketika request yang datang banyak [5].

Sebagai contoh terdapat dua *service-HTTP* yaitu *service HTTP-1* (terdapat 3 *active HTTP transaction*) dan *service HTTP 2* (terdapat 1 *active HTTP transaction*), maka *service HTTP-2* akan menerima *request* selanjutnya dikarenakan *service HTTP-1* > *Service HTTP-2* pada nilai transaksi aktifnya [6].

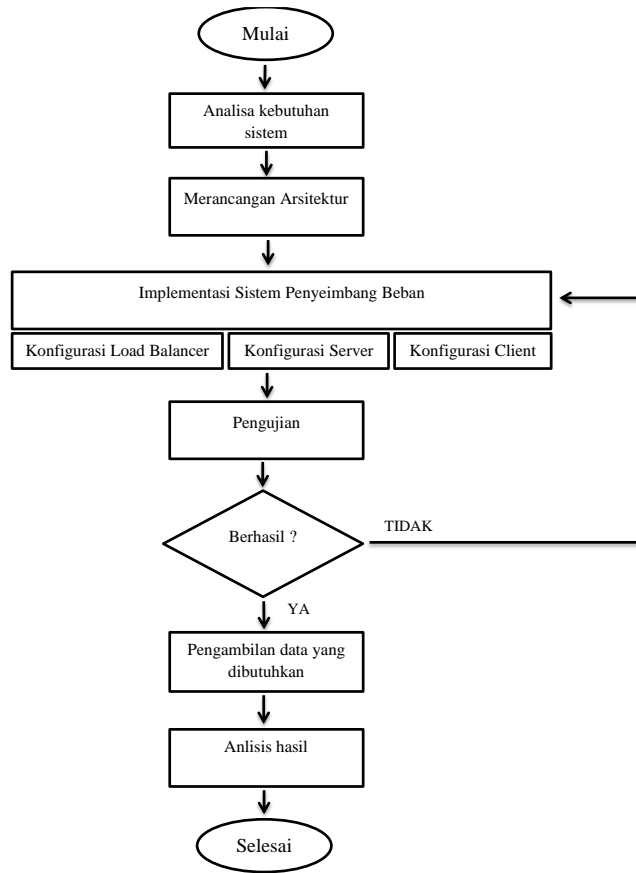


Gambar 3.1 Proses *Least Connection*^[6]

3. Perancangan dan Implementasi Sistem

3.1. Perancangan Sistem

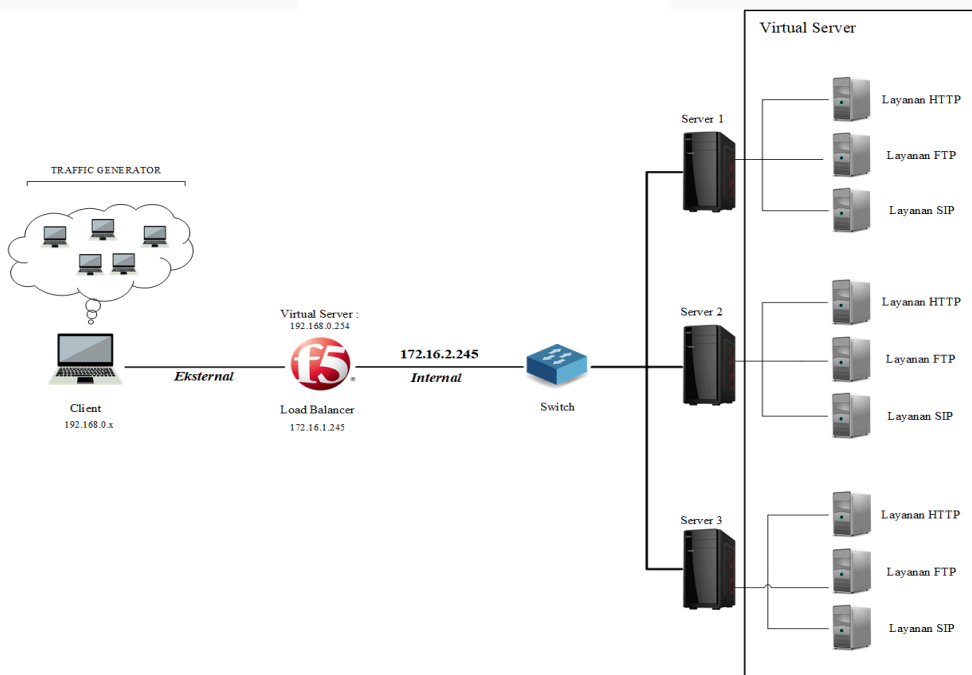
Berdasarkan perancangan system proyek akhir, teknologi yang digunakan sebagai *load balancer* adalah F5 BIG-IP LTM *Load Balancer*. Tugas *load balancer* adalah membagi beban berupa *request* yang diberikan oleh *client* kepada server. Algoritma yang digunakan dalam proyek akhir ini adalah *Least Connection* dan layanan pada sistem berupa web server, FTP server, dan VoIP server.



Gambar 3.2 Diagram Alir Pengerjaan

3.2. Model Sistem

Model sistem Proyek Akhir dapat dilihat pada Gambar 3.3.



Gambar 3.3 Model Sistem Load Balancing

Berdasarkan arsitektur, sistem mempunyai dua model jaringan yaitu jaringan *eksternal* dan jaringan *internal*. Jaringan *eksternal* terdiri *client* dan jaringan *internal* terdiri dari tiga server yang terdapat tiga *virtual* server pada setiap server dengan menggunakan protocol HTTP, TCP, dan UDP. *Virtual* server ini akan diakses oleh *client* pada jaringan *eksternal*. *Client* ini memberikan *request* ke server untuk menguji sejauh mana tingkat kinerja dari *load balancer*.

3.3. Perangkat Keras dan Perangkat Lunak

Perangkat keras yang digunakan untuk penelitian proyek akhir ini adalah sebagai berikut:

Tabel 3.1 Spesifikasi Komputer Load Balancer

Instance	Processor	RAM	HDD	OS
Load Balancer	2	4	42	BIG-IP-13.1.3-0.0.6 LTM

Tabel 3.2 Spesifikasi Komputer Server

Instance	Processor	RAM	HDD	OS
Web Server 1	1	1	20	Ubuntu 12
Web Server 2	1	1	20	Ubuntu 12
Web Server 3	1	1	20	Ubuntu 12
FTP Server 1	1	1	20	Ubuntu 12
FTP Server 2	1	1	20	Ubuntu 12
FTP Server 3	1	1	20	Ubuntu 12
VoIP Server 1	1	1	20	Ubuntu 16
VoIP Server 2	1	1	20	Ubuntu 16
VoIP Server 3	1	1	20	Ubuntu 16

Perangkat lunak yang digunakan untuk penelitian proyek akhir ini adalah sebagai berikut:

1. BIGIP 13.1.3.1 0.0.6.LTM sebagai *virtual load balancer*.
2. VMware sebagai perangkat lunak untuk *virtual machine* yang menjalankan banyak sistem operasi dalam satu perangkat keras.
3. *Web Server Apache* sebagai *software* pada web server.
4. Proftpd sebagai *software* Ftp Server.
5. Asterisk sebagai *software* Sip Server.
6. Httperf sebagai *software* untuk membangkitkan *request* pada layanan web server
7. Apache Jmeter sebagai *software* untuk membangkitkan trafik pada layanan FTP server.
8. SIPp berfungsi untuk membangkitkan jumlah panggilan pada layanan VoIP server.

3.4. Skenario Pengujian

Pada proyek akhir ini akan dilakukan pengujian *load balancer* terhadap tiga layanan yaitu web server, FTP server, dan VoIP server dengan menggunakan algoritma *Least Connection*. Pengujian ini dilakukan untuk mendapatkan parameter *throughput*, *response time*, dan *request loss* pada layanan web server dan FTP server. Sedangkan layanan VoIP server dilakukan pengujian untuk mendapatkan parameter *block call*. Terdapat dua skenario pengujian pada proyek akhir ini yaitu hubungan *load balancer* dengan jaringan *eksternal* menggunakan *wireless* dan *wireline*.

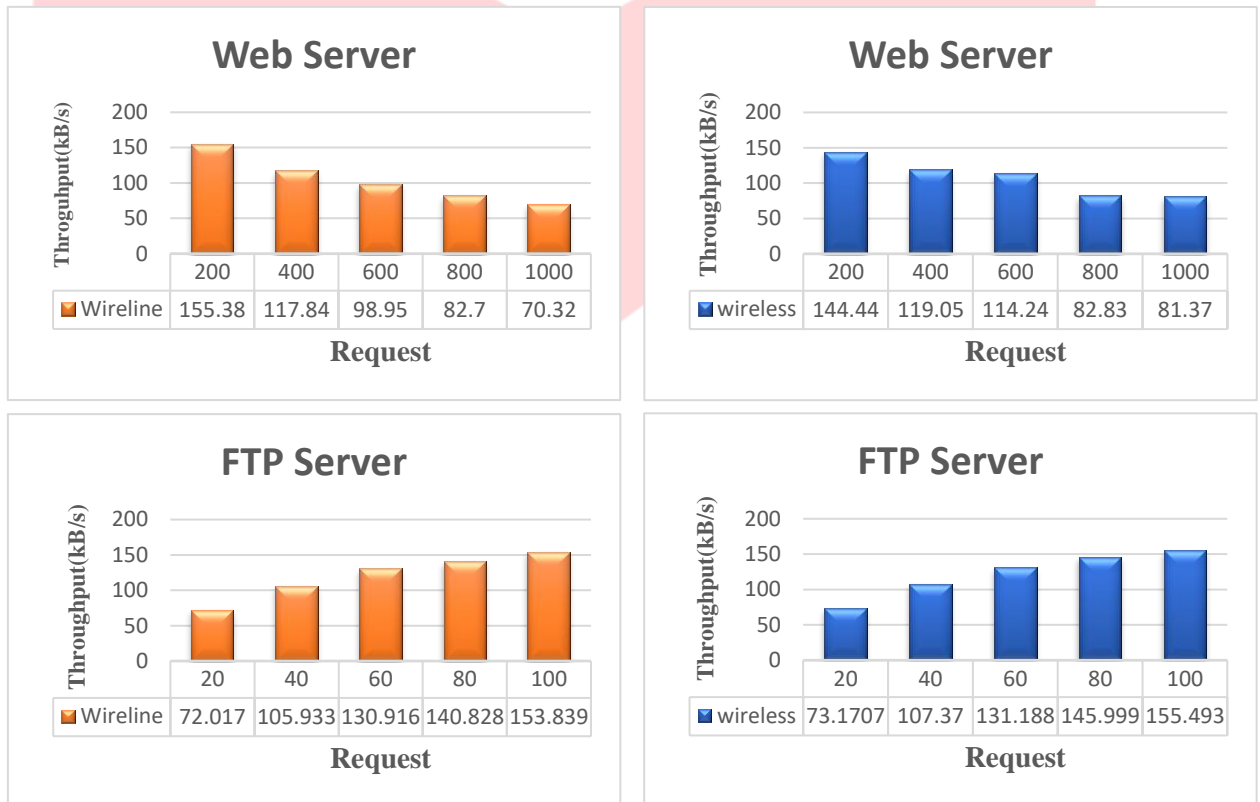
Pada setiap layanan akan diberikan beban berupa *request* dari *clinet* menggunakan *software* pembangkit *request*. Untuk layanan web server dilakukan pengujian dengan memberikan beban ke server sebanyak 200 *request*, 400 *request*, 600 *request*, 800 *request*, dan 1000 *request* dengan menggunakan *software* httperf. Pada layanan FTP terdapat *file transfer upload* dan *file transfer download* dengan ukuran file sebesar 160 KB dilakukan pengujian dengan jumlah *request* sebanyak 20 *file transfer*, 40 *file transfer*, 60 *file transfer*, 80 *file transfer*, dan 100 *file transfer* menggunakan *software* Apache Jmeter. Pada layanan VoIP server dilakukan pengujian dengan melakukan panggilan sebanyak 10 cps, 15 cps, 20 cps, 25 cps, dan 30 cps selama satu menit menggunakan *software* SIPp. Pengujian terhadap *load balancer* pada ketiga server dilakukan

secara bersamaan. Dalam pengujian ini dilakukan pengambilan data sebanyak 10 kali untuk setiap *request* yang diberikan untuk mendapat hasil yang maksimal.

4. Hasil Pengujian dan Analisis

Berikut adalah hasil pengujian dan analisis setiap parameter yang diuji.

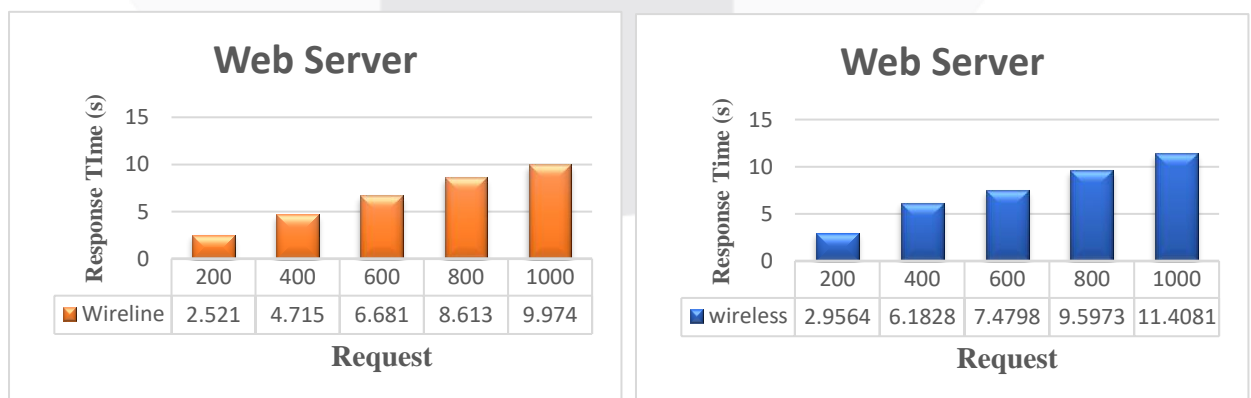
4.1 Throughput

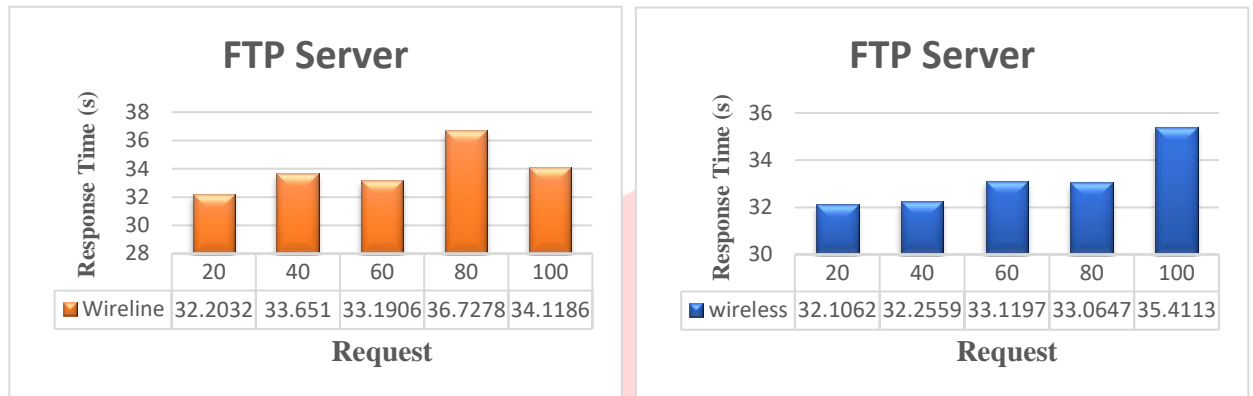


Gambar 4.1 Grafik Hasil Pengujian *Throughput* pada Web Server dan FTP Server

Pada Gambar 4.1 menunjukkan nilai *throughput* dari algoritma *least connection* pada web server dan FTP server. Semakin besar nilai *throughput* maka semakin baik juga kinerja dari *load balancer* yang digunakan pada proyek akhir ini. Grafik diatas menunjukkan nilai *throughput* saat pengujian pada web server cenderung menurun seiring dengan bertambahnya *request* yang diberikan, sedangkan pada pengujian FTP server nilai *throughput* mengalami kenaikan seiring dengan bertambahnya jumlah *file transfer*. Hal ini disebabkan karena waktu yang dibutuhkan untuk melayani *request file transfer* lebih stabil dan konstan dibandingkan *throughput* pada web server. Nilai rata-rata *throughput* paling besar pada kedua jenis layanan adalah pada saat koneksi antara *load balancer* dan *client* menggunakan *wireless* yaitu web server sebesar 108,38 KB/s dan FTP server sebesar 122,64 KB/s. sehingga lebih optimal dalam mendistribusikan *request* dengan menggunakan koneksi *wireless* antara *load balancer* dan *client*.

4.2 Response Time

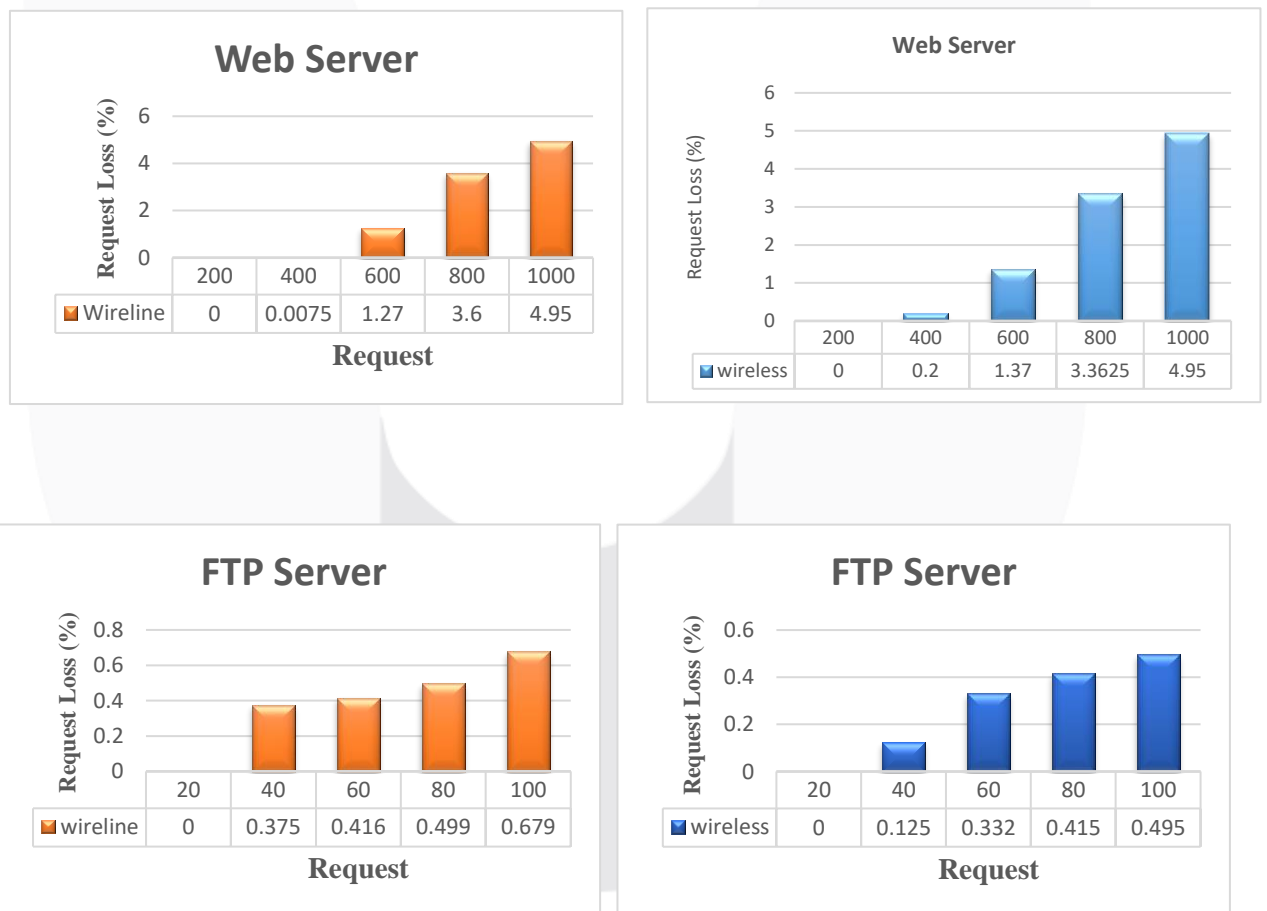




Gambar 4.2 Grafik Hasil Pengujian *Response Time* pada Web Server dan FTP Server

Pada Gambar 4.2 menunjukkan nilai *response time* dari algoritma *least connection* pada web server dan FTP server. Semakin kecil nilai *response time* maka semakin cepat suatu server untuk melayani *request* yang diberikan dari *client*. Berdasarkan grafik nilai *response time* cenderung mengalami kenaikan seiring dengan bertambahnya jumlah *request* pada web server dan *file transfer* pada FTP server. Hal ini membuktikan bahwa bertambahnya jumlah *request* dan *file transfer* maka semakin lama waktu yang dibutuhkan untuk server melayani *request* tersebut. Hasil pengujian parameter *response time* menunjukan nilai rata-rata *response time* terbaik pada saat web server menggunakan koneksi *wireline* dan FTP menggunakan koneksi *wireless* antara *load balancer* dan *client* yaitu 6,501 detik dan 33,19 detik.

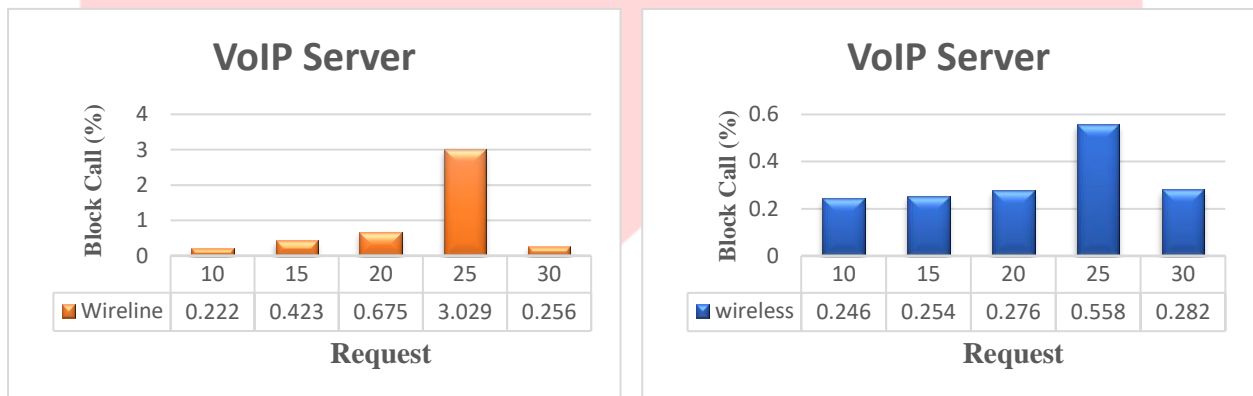
4.3 Request Loss



Gambar 4.3 Grafik Hasil Pengujian *Request Loss* pada Web Server dan FTP Server

Pada Gambar 4.3 menunjukkan nilai *request loss* dari algoritma *least connection* pada web server dan FTP server. Parameter *request loss* digunakan untuk menunjukkan jumlah kegagalan *request* yang terjadi saat server tidak dapat melayani *request* tersebut. Semakin kecil nilai *request loss* maka semakin baik kinerja server dalam melayani permintaan dari *client*. Pada layanan web server memiliki nilai *request loss* terkecil saat menggunakan koneksi *wireline* antara *load balancer* dan *client* yaitu sebesar 1,97% dan pada layanan FTP server saat menggunakan koneksi *wireless* yaitu sebesar 0,27%. Berdasarkan standar nilai *request loss* menunjukkan bahwa nilai *request loss* pada kedua layanan dapat dikategorikan sangat baik pada sistem penyeimbang beban ini.

4.4 Block Call



Gambar 4.4 Grafik Hasil Pengujian *Block Call* pada VoIP Server

Pada Gambar 4.4 menunjukkan nilai *block call* dari algoritma *least connection* pada VoIP server. Semakin kecil nilai *block call* maka semakin baik kinerja dari sistem penyeimbang beban dalam mendistribusikan beban untuk kedua server. Berdasarkan grafik diatas nilai rata-rata *block call* terkecil terdapat pada saat koneksi antara *load balancer* dan *client* menggunakan *wireless* yaitu 0,32% dari total kegagalan.

5. Kesimpulan

Berdasarkan hasil penelitian selama pengerjaan proyek akhir ini dapat ditarik kesimpulan bahwa:

- Berhasil melakukan perancangan penyeimbang beban menggunakan F5 BIG-IP LTM terhadap ketiga layanan yaitu web server, FTP server, dan VoIP server.
- Berhasil melakukan perancangan penyeimbang beban menggunakan F5 BIG-IP LTM dengan dua skenario yaitu menggunakan *wireline* dan *wireless* pada koneksi antara *load balancer* dan *client*.
- Algoritma *least connection* dapat dijalankan dengan F5 BIG-IP LTM.
- Hasil pengujian *throughput* pada skenario menggunakan *wireless* antara *load balancer* dan *client* memiliki nilai *throughput* tertinggi yaitu pada layanan web server sebesar 108,38 KB/s dan FTP server yaitu 122,64 KB/s. sehingga lebih optimal dalam mendistribusikan *request* dengan menggunakan koneksi *wireless* antara *load balancer* dan *client*.
- Hasil pengujian *response time* saat koneksi *wireline* memiliki nilai *response time* tercepat pada layanan web server yaitu 6,501 detik dan pada saat koneksi *wireless* memiliki nilai *response time* tercepat pada layanan FTP server yaitu 33,19 detik.
- Hasil pengujian *request loss* pada saat koneksi *wireline* memiliki nilai *request loss* paling sedikit pada layanan web server yaitu 1,97% dan pada saat koneksi *wireless* memiliki nilai *request loss* paling sedikit pada layanan FTP Server yaitu 0,27%
- Dengan menggunakan koneksi *wireless* antara *load balancer* dan *client* dapat meminimalkan *block call* pada VoIP server yaitu dengan nilai sebesar 0,32% kegagalan panggilan.

DAFTAR PUSTAKA

- [1] Bourke, Tony. 2011. *Server Load Balancing*, Beijing “O’Reilly”.
- [2] Lukitasari D, Oklilas AF. 2010. *Analisis perbandingan load balancing web server tunggal dengan web server cluster menggunakan linux virtual server*. Jurnal Generic. 5(2):31-34. doi:1907-4093/2087-9814.
- [3] Martasurianagara, Satria. 2016. *Analisa Kinerja Load Balancing menggunakan Linux Virtual Server pada layanan Http.Bandung “universitas Telkom”*.
- [4] Munandar, Aris. 2014. *Analisis Kinerja penyeimbang Beban Server Web dengan Algoritme Rasio Dinamis*.Bogor “Institut Pertanian Bogor”.
- [5] Nasution AH. 2011. *Komparasi algoritma penjadwalan pada layanan terdistribusi load balancing LVS via NAT*. Surabaya “Institut Sepuluh November”.
- [6] Nasser H., Witono T. 2016. *Analisis Algoritma Round Robin, Least Connection, dan Ratio pada Load Balancing Menggunakan OPNET Modeler*. Bandung. Universitas Kristen Maranatha.