

ANALISIS PERBANDINGAN KINERJA WEB SERVER TUNGGAL DENGAN LOAD BALANCING WEB SERVER SEBAGAI SOLUSI MENGATASI BEBAN KERJA WEB SERVER

Mahmud

*Teknik Informatika STMIK PalComTech
Jl. Basuki Rahmat No. 05, Palembang 30129, Indonesia
e-mail: m4h86mud@gmail.com*

Abstrak – *Load Balancing Web Server* merupakan salah satu cara yang digunakan untuk meningkatkan kinerja dan tingkat ketersediaan *Web Server*, yaitu dengan membagi *request* yang datang ke beberapa *server* sekaligus, sehingga beban yang ditanggung oleh masing-masing lebih ringan. Jadi dikarenakan mendapatkan beban kerja yang lebih ringan maka secara langsung mempengaruhi performa *Web Server* yang semakin responsif, tingkat ketersediaan *Web server* juga dapat lebih terjaga. Dengan menggunakan teknologi *Load Balancing*, maka akan ada perbedaan pada kinerja *Web Server Load Balancing* dengan *Web Server* Tunggal, yang dapat dijadikan perbandingan, sehingga dapat menjadi solusi bagi pengelola *Web Server*.

Keyword – *Web Server, Load Balancing, Tunggal, Kinerja*.

I. PENDAHULUAN

Perkembangan teknologi informasi hingga saat ini berkembang sangat pesat ini, perkembangan ini dapat terlihat dari penggunaan-penggunaan *website* sebagai penyedia informasi. *Web server* adalah komponen utama dalam pembangunan sebuah *website*, jadi dibutuhkanlah sebuah *web server* yang handal dan memiliki kehandalan, ketersediaan dan skalabilitas yang baik. Sebuah layanan penyedia informasi dalam hal ini adalah sebuah *website* yang diakses oleh jutaan pengguna informasi maka setiap *request* ke *web server* akan menjadi beban, dan makin banyak *request* yang dilayani maka semakin meningkat juga beban sebuah *web server*, ini apabila *web server* mengalami *overload* maka akan menyebabkan *webserver* menjadi *down*. Ini tentu saja secara langsung menjadi sebuah kerugian sebagai penyedia informasi dan pengguna informasi.

Load Balancing Web Server merupakan salah satu cara yang digunakan untuk meningkatkan kinerja dan tingkat ketersediaan *Web Server*, yaitu dengan membagi *request* yang datang ke beberapa *server* sekaligus, sehingga beban yang ditanggung oleh masing-masing lebih ringan. Jadi dikarenakan mendapatkan beban kerja yang lebih ringan maka secara langsung mempengaruhi performa *Web Server* yang semakin responsif. Tingkat Ketersediaan *Web server* juga dapat lebih terjaga. Dengan penggunaan teknologi *Load Balancing* ini, yaitu ketika salah satu *Web Server* tidak dapat melayani pengguna, maka otomatis *Web Server* yang lain akan langsung

menggantikan kerjanya, sehingga pengguna tidak akan dapat mengetahui bahwa *Web Server* mengalami kegagalan layanan.

Berdasarkan hasil latar belakang diatas, maka Penulis tertarik untuk menganalisis bagaimana perbedaan kinerja antara *Web Server* tunggal dengan *Web Server* yang menggunakan teknologi *Load Balancing* sehingga dapat memberikan kejelasan perbedaan mengapa *Load Balancing* dapat menjadi solusi bagi sebuah *Web Server* yang memiliki beban kerja yang tinggi.

II. METODE PENELITIAN

Dalam penelitian ini Penulis menggunakan jenis penelitian eksperimen (*Experimental Research*). Penelitian eksperimen pada prinsipnya dapat didefinisikan sebagai metode sistematis guna membangun, hubungan yang mengandung fenomena sebab akibat (*Causal-Effect Relationship*) [1]. Metode eksperimen adalah metode penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dalam kondisi terkendali [2]. Berdasarkan sumber-sumber diatas dapat disimpulkan bahwa penelitian eksperimental atau metode eksperimental merupakan suatu bentuk penelitian yang digunakan untuk mengetahui suatu bentuk pengaruh hasil pemberian suatu treatment atau perlakuan pada suatu objek penelitian.

A. Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian ini adalah data penggunaan beban CPU, data penggunaan RAM dan data penggunaan SWAP serta *response time* dan *fail request* dengan melakukan pengujian kinerja dengan dibebani kerja berupa jumlah *request* dan *bandwidth* dengan menggunakan perangkat lunak yaitu Apache JMeter.

B. Alat Analisis

Alat analisis yang digunakan adalah berupa aplikasi *Resource Monitor* yang telah disediakan oleh FreeBSD untuk memberikan informasi berupa penggunaan CPU, RAM dan SWAP. Untuk simulasi beban *request* menggunakan perangkat lunak Apache JMeter untuk melakukan *request* secara bersamaan. Lalu, untuk pembatasan *bandwidth* menggunakan konfigurasi *packet filter* pada Sistem Operasi *FreeBSD*, konfigurasi ini membatasi besarnya *bandwidth* yang dilewati oleh *client* ke *server*. Dan

untuk pengukuran *response time* dan *fail request* juga menggunakan perangkat lunak Apache JMeter.

C. Metodologi

Metodologi yang digunakan adalah deskriptif kualitatif untuk mengidentifikasi masalah, harus dilakukan analisis perbandingan terhadap kinerja *Web Server* Tunggal dan *Web Server Load Balancing* yang diukur berdasarkan beban penggunaan CPU, penggunaan RAM, penggunaan SWAP, *response time* dan *fail request*.

D. Variabel Penelitian

Variabel penelitian yang Penulis gunakan dalam penelitian ini adalah perbandingan kinerja *Web Server* Tunggal dan *Load Balancing Web Server*. Adapun variabel-variabel tersebut dapat diuraikan sebagai berikut:

Tabel 1. Variabel Penelitian

Variabel	Indikator
1. Request	1. Penggunaan CPU
2. Bandwidth	2. Penggunaan RAM
	3. Penggunaan SWAP
	4. Response Time
	5. Fail Request

Kelima indikator pengukuran pada Tabel 1 diambil karena setiap request dari client kepada *Web Server* adalah merupakan beban bagi *Web Server* sehingga semakin banyak *request* yang datang maka akan semakin membebani kinerja *Web Server* seperti pada CPU, RAM dan SWAP. Untuk *Response Time* dan *fail request* diukur dikarenakan *Web Server* bekerja pada jaringan komputer baik lokal maupun publik dan ini berpengaruh terhadap besar kecilnya *bandwidth* yang tersedia.

E. Metode Pengumpulan Data

Metode yang Penulis gunakan dalam penelitian ini adalah dengan metode Observasi. Metode ini dilakukan Penulis dengan cara mengamati dan menganalisa dan melakukan pencatatan informasi yang sesuai dengan konteks penelitian dan yang berhubungan dengan ruang lingkup penelitian yang dilakukan oleh Penulis. Penulis melakukan pencatatan atas hasil dari pengujian-pengujian yang dilakukan serta mempelajarinya. Dan Penulis perbandingan terhadap data yang telah didapatkan agar dapat membandingkan hasil pengukuran yang telah di dapat dengan dokumentasi yang ada, sehingga dapat ditarik kesimpulan dari hasil penelitian ini.

F. Teknik Pengujian dan Analisa Data

Teknik pengujian ini akan dilaksanakan dengan melakukan pengujian kinerja terhadap kedua *Web Server* yaitu *Web Server Tunggal* dan *Load Balancing Web Server*. Teknik pengujian dilakukan secara bergantian dengan membebani masing *Web Server* dan setelah itu dilihat *resource monitor* sebagai alat ukur untuk kemampuan kedua *Web Server* tersebut.

Untuk pengujian *response time* dan *fail request* dilakukan dengan cara yaitu pada saat *Web Server* itu

dibebani dengan jumlah *request* dengan menggunakan aplikasi Apache JMeter yang dieksekusi di client maka akan didapatkan nilai *response time* dan *fail request* dari setiap user sesuai dengan jumlah *request*, setelah didapatkan hasil data tersebut maka selanjutnya dibuat rata-rata dari setiap user untuk *response time* dan *fail request*.

Pengujian penelitian ini dilakukan terhadap *Web Server* Tunggal dan *Load Balancing Web Server* dengan melakukan pembatasan *bandwidth* masing-masing yaitu 512 Kbps, 2 Mbps, 5 Mbps dan 10 Mbps. Hal ini dilakukan untuk melihat perbedaan dari masing-masing *Web Server* berikut berdasarkan jumlah *bandwidth*-nya, apakah besar kecilnya *bandwidth* berpengaruh dengan *resource* dari *Web Server* dari masing-masing tersebut.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Web Server Tunggal dan Load Balancing Web Server

Pengujian terhadap *Web Server* Tunggal dan *Web Server Load Balancing* yang dilakukan dengan sebanyak 4 kali dengan masing-masing dibebani simulasi *request* dengan menggunakan aplikasi Apache JMeter yang dieksekusi dari komputer *client*. Adapun masing-masing pengujian dibebani *request* sejumlah 50, 100, 150, 200, 250, 300, 350, 400, 450 dan 500. Pengujian ini juga dilakukan dengan membatasi lebar *Bandwidth* yang sudah ditentukan yaitu masing-masing 512 Kbps, 2 Mbps, 5 Mbps dan 10 Mbps.

1) Pengujian Bandwidth 512 Kbps: Untuk hasil pengujian yang pertama adalah dengan melakukan pengukuran penggunaan CPU, RAM dan SWAP. Lalu setelah itu dilakukan pengukuran selanjutnya yaitu dengan *Average Response Time* dan *Fail Request*.

Tabel 2. Hasil Pengujian Beban CPU, RAM dan SWAP (512 Kbps)

Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	CPU (%)	69,8	89,9	91,7	95,7	100	100	100	100	100	100
	RAM (M)	611	1028	1164	1365	1605	1651	1831	1858	1898	1936
	SWAP (M)	0	0	0	0	0	0	0	0	0	0
Load Balancing 1	CPU (%)	34,1	67,2	75,4	89,9	89,9	100	100	100	100	100
	RAM (M)	442	815	996	1024	1354	1387	1403	1484	1599	1644
	SWAP (M)	0	0	0	0	0	0	0	0	0	0
Load Balancing 2	CPU (%)	37,4	66,3	76,4	87,9	90,0	100	100	100	100	100
	RAM (M)	444	831	891	1111	1272	1331	1500	1551	1589	1774
	SWAP (M)	0	0	0	0	0	0	0	0	0	0

Dari tabel 2 dapat dilihat beban yang diterima oleh masing *Web Server* dalam bentuk penggunaan CPU, penggunaan RAM dan penggunaan SWAP, pada saat *Web Server* menerima *request* dari client dan dibatasi lebar *bandwidth*-nya maksimal dengan 512 Kbps. Dari tabel 2 tersebut juga dapat disimpulkan bahwa jumlah *request* yang masuk akan sangat berpengaruh dengan beban CPU dan RAM, namun untuk penggunaan SWAP masih nol atau digunakan. Selanjutnya adalah data hasil pengujian untuk mengetahui rata-rata waktu

respon (Average Response Time) dan kegagalan layanan (Fail Request), seperti pada tabel di bawah ini.

Tabel 3. Hasil Pengujian Response Time dan Fail Request (512 Kbps)

Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	Ave. Resp. Time(ms)	3682	8417	13693	15420	19288	20892	23355	27423	23569	22467
	Fail Request (%)	0	0	0	0	12,4	21,0	30,6	38,33	50,22	51,0
Load Balancing	Ave. Resp. Time(ms)	3988	10013	15148	17848	25941	30488	32417	33395	34148	32417
	Fail Request (%)	0	0	0	0	0	0	0	16,75	17,67	18,40

Dari Tabel 3 dapat dilihat bahwa semakin banyak jumlah *request* yang masuk maka akan berdampak kepada semakin besarnya *response time* dan pada saat mencapai jumlah 500 *request* maka *Web Server* mengalami kegagalan layanan (*Fail Request*).

2) Pengujian Bandwidth 2 Mbps: Untuk hasil pengujian yang kedua yaitu dengan membatasi lebar bandwidth sebesar 2 Mbps adalah dengan melakukan pengukuran penggunaan CPU, RAM dan SWAP. Lalu setelah itu dilakukan pengukuran selanjutnya yaitu dengan Average Response Time dan Fail Request.

Tabel 4. Hasil Pengujian Beban CPU, RAM dan SWAP (2 Mbps)

Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	CPU (%)	97,0	100	100	100	100	100	100	100	100	100
	RAM (M)	729	1083	1134	1469	1607	1653	1735	1790	1808	1815
	SWAP (M)	0	0	0	0	0	0	0	0	0	0
Load Balancing 1	CPU (%)	46,7	70,1	85,9	90,9	93,0	100	100	100	100	100
	RAM (M)	545	903	996	1064	1333	1389	1413	1501	1660	1662
	SWAP (M)	0	0	0	0	0	0	0	0	0	0
Load Balancing 2	CPU (%)	52,0	69,9	86,6	93,9	95,7	100	100	100	100	100
	RAM (M)	768	948	1024	1229	1248	1321	1500	1651	1580	1766
	SWAP (M)	0	0	0	0	0	0	0	0	0	0

Dari Tabel 4 dapat dilihat beban yang diterima oleh masing *Web Server* dalam bentuk penggunaan CPU, penggunaan RAM dan penggunaan SWAP, pada saat *Web Server* menerima *request* dari client dan dibatasi lebar *bandwidth*-nya maksimal dengan 2 Mbps. Dari Tabel IV tersebut juga dapat disimpulkan bahwa penggunaan CPU dan RAM meningkat dari pengujian sebelumnya dan untuk penggunaan SWAP masih nol atau digunakan. Selanjutnya adalah data hasil pengujian untuk mengetahui rata-rata waktu respon (*Average Response Time*) dan kegagalan layanan (*Fail Request*), seperti pada tabel di bawah ini.

Tabel 5. Hasil Pengujian Response Time dan Fail Request (2 Mbps)

Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	Ave. Resp. Time(ms)	2761	4801	5511	6662	6882	7007	6677	6551	4877	4367
	Fail Request (%)	0	0	0	0	16,8	28	37,71	45,75	52,0	52,0
Load Balancing	Ave. Resp. Time(ms)	1301	2142	2726	3880	5653	7611	7683	9053	9106	8658
	Fail Request (%)	0	0	0	0	0	0	0	0	1,78	7,8

Dari Tabel 5 dapat dilihat bahwa dibanding pada pengujian sebelumnya, ada peningkatan jumlah *Fail Request* pada *Web Server* Tunggal dan juga peningkatan response time pada setiap pengujian.

B. Perbandingan Web Server Tunggal dan Load Balancing Web Server

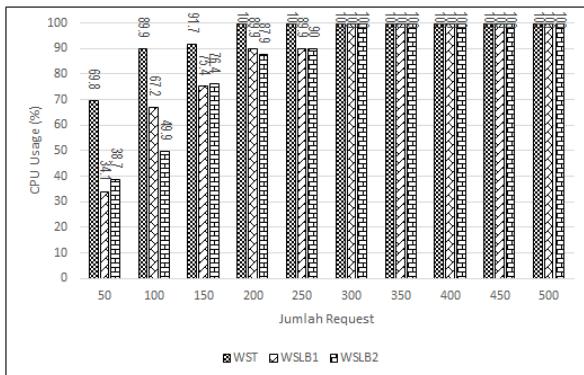
Setelah hasil pengujian dilakukan maka telah didapatkan data-data yang dapat selanjutnya dianalisa untuk dilakukan perbandingan antara *Web Server* Tunggal dan *Load Balancing Web Server* dengan memperhatikan beberapa variabel-variabel yang sudah diukur. Dari hasil pengujian yang telah dilakukan, untuk pengukuran penggunaan SWAP tidak akan di lakukan perbandingan, ini dikarenakan selama hasil pengujian data hasil yang didapat menunjukkan angka '0' yang artinya tidak ada penggunaan SWAP selama proses pengujian dilakukan, sehingga Penulis memutuskan tidak akan menggunakan variabel penggunaan SWAP untuk dilakukan perbandingan dengan variabel lainnya. Seluruh data hasil pengujian akan dituangkan dalam bentuk grafik sehingga akan terlihat perbandingan dengan lebih jelas.

1) Hasil Perbandingan Bandwidth 512 Kbps: Hasil Perbandingan dari pengujian Beban CPU dan RAM dengan lebar bandwidth 512 Kbps, dapat dilihat pada tabel 6 dibawah ini.

Tabel 6 Perbandingan Beban CPU dan RAM (512 Kbps)

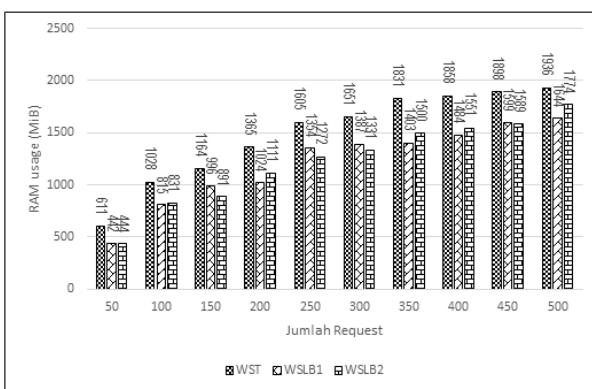
Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	CPU (%)	69,8	89,9	91,7	95,7	100	100	100	100	100	100
	RAM (M)	611	1028	1164	1365	1605	1651	1831	1858	1898	1936
Load Balancing 1	CPU (%)	34,1	67,2	75,4	89,9	100	100	100	100	100	100
	RAM (M)	442	815	996	1024	1354	1387	1403	1484	1599	1644
Load Balancing 2	CPU (%)	37,4	66,3	76,4	93,9	100	100	100	100	100	100
	RAM (M)	444	831	891	1111	1272	1331	1500	1551	1589	1774

Selanjutnya dari tabel 6 dapat dibuatkan grafik dari masing-masing parameter CPU dan RAM sesuai dengan data diatas, dapat dilihat pada Gambar 1.



Gambar 10. Perbandingan Penggunaan CPU Web Server Tunggal dan Load Balancing (512 Kbps)

Pada Gambar 1, untuk penggunaan CPU Web Server Tunggal dalam melayani 50 request maka didapatkan beban CPU sebesar 69,8%, dibanding dengan beban CPU yang diterima oleh Web Server Load Balancing yang didapatkan masing-masing 34,1% dan 38,7%. Pada saat beban request yang diterima sebanyak 150 request, Web Server Tunggal sudah mulai mendekati beban puncak karena beban CPU yang bekerja senilai 91,7%, berbeda dengan Web Server Load Balancing dengan beban CPU pada masing-masing beban Web Server berkisar di 75%. Pada saat beban request yang sebanyak 200 request, Web Server Tunggal sudah menemui beban puncak karena beban CPU sudah mencapai 100% dan beban CPU Web Server Load Balancing pada masing-masing Web Server adalah 89,9% dan 87,9. Sedangkan untuk Load Balancing Web Server mendapatkan beban puncak CPU dimulai dari 300 request.



Gambar 11. Perbandingan Penggunaan RAM Web Server Tunggal dan Load Balancing (512 Kbps)

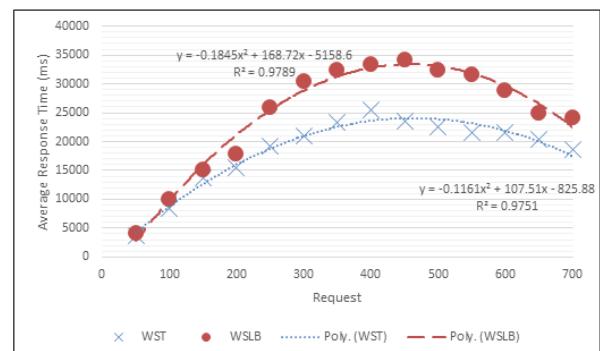
Untuk penggunaan RAM ini dapat dilihat pada Gambar 5.2, penggunaan RAM pada beban *request* 50, *Web Server Tunggal* menggunakan RAM sebesar 611 MiB dan penggunaan RAM pada *Load Balancing Web Server* masing-masing menggunakan 442 MiB dan 444 MiB, begitupun seterusnya sampai dengan beban *request* 500, *Web Server Tunggal* sudah hampir menggunakan total RAM yang ada yaitu 2048 MiB dengan penggunaan RAM sudah mencapai 1936 MiB, namun pada *Load Balancing Web Server* sudah mendekati maksimal penggunaan RAM dengan masing-masing *Web Server* menggunakan 1644 MiB dan 1774 MiB.

Hasil Perbandingan dari pengujian *Average Response Time* dan *Fail Request* dengan lebar *bandwidth* 512 Kbps, dapat dilihat pada tabel 7 dibawah ini.

Tabel 7. Perbandingan Response Time dan Fail Request (512 Kbps)

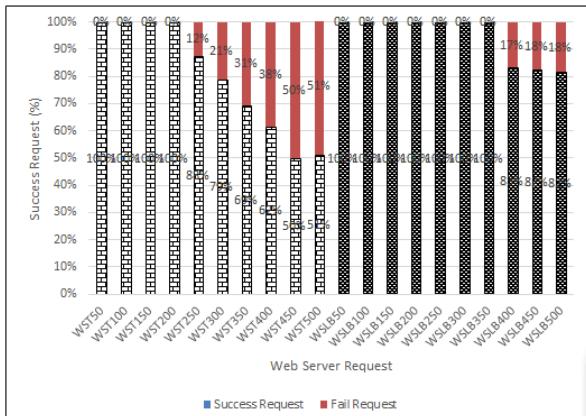
Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Terus	Ave. Resp. Time(ms)	3682	8417	13693	15420	19288	20892	22366	27423	23569	22467
Terus	Fail Request (%)	0	0	0	0	12,4	21,0	30,6	38,33	50,22	51,0
Load Balancing	Ave. Resp. Time(ms)	3988	10013	15148	17848	25941	30488	32417	33395	34148	32417
Load Balancing	Fail Request (%)	0	0	0	0	0	0	0	16,75	17,67	18,40

Selanjutnya dari tabel 7 dapat dibuatkan grafik untuk *Average Response Time* dan *Fail Request* sesuai dengan data tabel 9 diatas, dapat dilihat pada gambar dibawah ini.



Gambar 12. Perbandingan Average Response Time Web Server Tunggal dan Load Balancing (512 Kbps)

Dari gambar 3 dapat dilihat perbandingan rata-rata waktu respon (*Average Response Time*) yang diterima *Web Server Tunggal* dan *Web Server Load Balancing* pada saat beban 50 *request*, *response time* yang diterima lebih kurang sama yaitu untuk *Web Server Tunggal* 3682 *milisecond* dan *Web Server Load Balancing* 3988 *milisecond*. Pada tampilan grafik ini juga Penulis sengaja menggunakan data sampai dengan pengujian 700 *request*, ini bertujuan untuk agar dapat memperlihatkan lebih jelas bagaimana korelasi dan puncak *response time* pada grafik tersebut. Dan dengan kondisi pengujian dengan *bandwidth* 512 Kbps berbanding dengan jumlah *request* dapat dihasilkan masing-masing persamaan regresi yaitu untuk *Web Server Tunggal* $y = -0,1161x^2 + 107,51x - 825,88$ dan $R^2 = 0,9751$, serta untuk *Load Balancing Web Server* adalah $y = -0,1845x^2 + 168,72x - 5158,6$ dan $R^2 = 0,9789$.



Gambar 13. Perbandingan Fail Request Web Server Tunggal dan Load Balancing (512 Kbps)

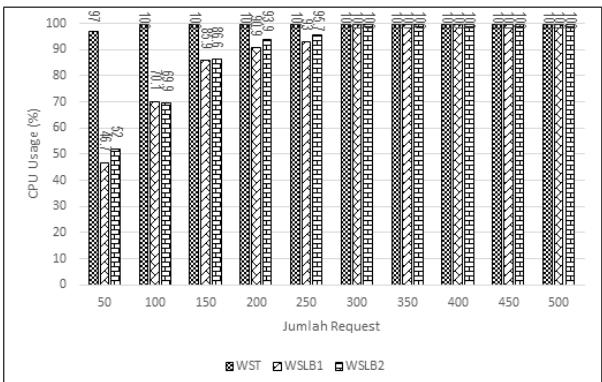
Pada Gambar 4, terlihat mulai ada perbedaan antara *Web Server Tunggal* dan *Web Server Load Balancing* pada beban *request* 250. Pada beban *request* 250, *Web Server* sudah mengalami *Fail Request* sebesar 12%, sedangkan *Web Server Load Balancing* masih mampu melayani seluruh *request* yang datang, dan terus bertambah jumlah *fail request* seiring dengan semakin banyaknya jumlah *request* yang masuk. Dan untuk *Load Balancing Web Server* pada saat beban 400 *request* baru mengalami *fail request* sebanyak 17%.

2) Hasil Perbandingan Bandwidth 2 Mbps: Hasil Perbandingan dari pengujian Beban CPU dan RAM dengan lebar bandwidth 2 Mbps, dapat dilihat pada tabel 8 dibawah ini.

Tabel 8. Perbandingan Beban CPU dan RAM (2 Mbps)

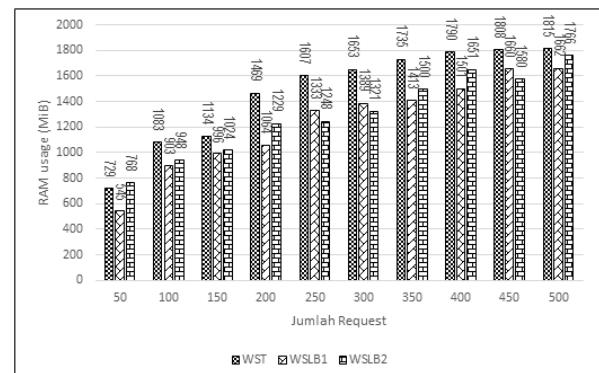
Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	CPU (%)	97,0	100	100	100	100	100	100	100	100	100
	RAM (M)	729	1083	1134	1469	1607	1653	1735	1790	1808	1815
Load Balancing 1	CPU (%)	46,7	70,1	85,9	90,9	93,0	100	100	100	100	100
	RAM (M)	545	903	996	1064	1333	1389	1413	1501	1660	1662
Load Balancing 2	CPU (%)	52,0	69,9	86,6	93,9	93,7	100	100	100	100	100
	RAM (M)	768	948	1024	1229	1248	1321	1500	1631	1580	1766

Selanjutnya dari tabel 8 dapat dibuatkan grafik dari masing-masing parameter CPU dan RAM sesuai dengan data diatas, dapat dilihat pada grafik dibawah ini.



Gambar 14. Perbandingan Penggunaan CPU Web Server Tunggal dan Load Balancing (2 Mbps)

Pada Gambar 5, untuk penggunaan CPU *Web Server Tunggal* dalam melayani 50 *request* maka didapatkan beban CPU sebesar 97%, dibanding dengan beban CPU yang diterima oleh *Web Server Load Balancing* yang didapatkan masing-masing 46,7% dan 52%. Dilihat dari data tersebut maka ada peningkatan penggunaan CPU dibanding sebelumnya menggunakan *bandwidth* 512 Kbps, peningkatan CPU meningkat kurang lebih sekitar 25% begitupun dengan beban CPU pada *Web Server Load Balancing*. Dan bila beban 100 *request*, *Web Server Tunggal* sudah mengalami puncak beban CPU yaitu 100%, berbeda dengan percobaan sebelumnya dengan *bandwidth* 512 Kbps yang belum menemui titik puncak beban CPU.



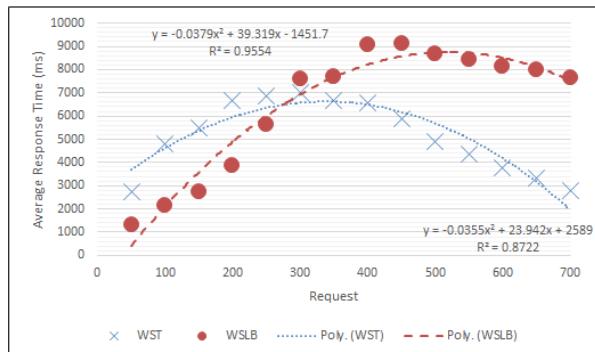
Gambar 15. Perbandingan Penggunaan RAM Web Server Tunggal dan Load Balancing (2 Mbps)

Untuk penggunaan RAM ini dapat dilihat pada Gambar 6, penggunaan RAM pada beban *request* 50 sampai dengan beban 500 *request* tidak berbeda jauh dengan percobaan sebelumnya, namun hampir keseluruhan penggunaan RAM baik untuk *Web Server Tunggal* maupun *Load Balancing Web Server* mengalami peningkatan lebih kurang 5% -10%.

Tabel 9. Perbandingan Response Time dan Fail Request (2 Mbps)

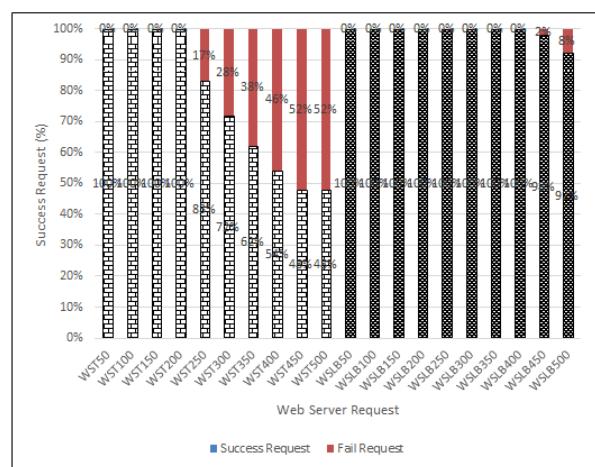
Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	Ave. Resp. Time(ms)	2761	4801	5511	6662	6882	7007	6677	6551	4877	4367
	Fail Request (%)	0	0	0	0	16,8	28	37,71	45,75	52,0	52,0
Load Balancing	Ave. Resp. Time(ms)	1301	2142	2726	3880	5653	7611	7683	9053	9106	8658
	Fail Request (%)	0	0	0	0	0	0	0	0	1,78	7,8

Selanjutnya dari tabel 9 dapat dibuatkan grafik untuk *Average Response Time* dan *Fail Request* sesuai dengan data tabel X diatas, dapat dilihat pada gambar dibawah ini.



Gambar 16. Perbandingan Average Response Time Web Server Tunggal dan Load Balancing (2 Mbps)

Dari gambar 7 dapat dilihat perbandingan rata-rata waktu respon (Average Response Time) yang diterima Web Server Tunggal dan Web Server Load Balancing. Berbeda dengan hasil dari pengujian dengan bandwidth 512 Kbps dimana response time Web Server Tunggal lebih cepat dibanding dengan Web Server Load Balancing di beban request 50 sampai 500. Di pengujian ini menghasilkan waktu respon yang berbeda pada beban request 50, 100, 150 dan 200, Web Server Load Balancing lebih unggul response time-nya dibanding dengan Web Server Tunggal. Disaat beban pengujian lebih kurang 250 sampai dengan 500 request, Web Server Tunggal lebih unggul dalam response time dibanding dengan waktu tunggu Web Server Load Balancing. Perbedaan lainnya yang dapat dilihat dibanding pada pengujian sebelumnya menggunakan bandwidth 512 Kbps, pada pengujian ini didapatkan grafik dari Web Server Tunggal dan Load Balancing Web Server yang bersinggungan di kurang lebih 250 request. Ini dapat terihat dari nilai average response time pada saat beban 500 request, Web Server Tunggal memiliki nilai 4877 milisecond dan Web Server Load Balancing 9106 milisecond. Dan dengan kondisi pengujian dengan bandwidth 2 Mbps berbanding dengan jumlah request dapat dihasilkan masing-masing persamaan regresi yaitu untuk Web Server Tunggal $y = -0,0355x^2 + 23,942x + 2589$ dan $R^2 = 0,8722$, serta untuk Load Balancing Web Server adalah $y = -0,0379x^2 + 39,319x - 1451,7$ dan $R^2 = 0,9554$.



Gambar 17. Perbandingan Fail Request Web Server Tunggal dan Load Balancing (2 Mbps)

Pada Gambar 8 terlihat perbedaan antara Web Server Tunggal dan Web Server Load Balancing, sama seperti pada pengujian sebelumnya. Untuk Web Server Tunggal mengalami Fail Request di mulai dari beban 250 sampai dengan 500 request, sedangkan pada Load Balancing Web Server sedikit berbeda, dilihat pada pengujian sebelumnya dengan bandwidth 512 Kbps, fail request terjadi pada 400 request dan pada pengujian dengan menggunakan bandwidth 2 Mbps, fail request terjadi pada 450 request.

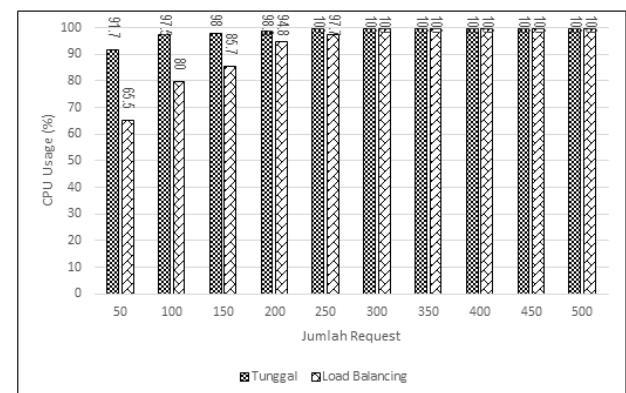
C. Pembahasan

Berdasarkan hasil dari pengujian dan perbandingan yang telah dilakukan terhadap Web Server Tunggal dan Load Balancing Web Server, selanjutnya akan dicari perbandingan antara penggunaan CPU, penggunaan RAM, serta average response time dengan fail request dari masing-masing lebar bandwidth.

Tabel 5 Penggunaan CPU dan RAM Web Server Tunggal dan Load Balancing

Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	CPU (%)	91,7	97,5	98	98,9	100	100	100	100	100	100
	RAM (MiB)	693	988	1153	1408	1642	1684	1786	1811	1830	1823
Load Balancing	CPU (%)	65,5	80	85,7	94,8	97,7	100	100	100	100	100
	RAM (MiB)	530	821	960	1111	1246	1380	1453	1577	1655	1697

Selanjutnya data pada tabel 10 di tampilkan dalam bentuk grafik pada gambar 9 untuk rata-rata CPU usage dan gambar 10 untuk rata-rata RAM usage dibawah ini.



Gambar 18. Rata-rata CPU Usage

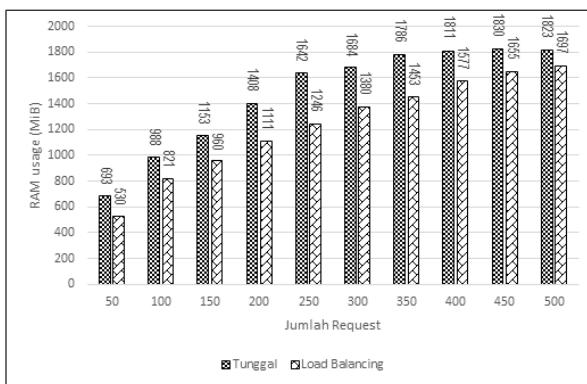
Dari gambar 9 untuk penggunaan CPU pada Web Server Tunggal, merupakan grafik dari hasil rata-rata penggunaan CPU seluruh bandwidth (512 Kbps, 2 Mbps, 5 Mbps dan 10 Mbps) dengan masing-masing request 50, 100, 150, 200, 250, 300, 350, 400, 450 dan 500. Dilihat dari gambar tersebut terlihat beban puncak penggunaan CPU 100% untuk Web Server Tunggal adalah di mulai dari 250 request, namun pada dasarnya mulai dari 100 request pun dapat dianggap bahwa Web Server Tunggal sudah menemui beban maksimal, ini dikarenakan beban yang digunakan sudah mendekati maksimal yaitu kisaran di 97%. Dan untuk Load Balancing Web Server sedikit berbeda, karena

penggunaan CPU 100% ada di 300 request, dan penggunaan CPU yang dianggap sudah beban maksimal adalah di 200 request dengan nilai kisaran 95%.

Penggunaan CPU yang sudah mendekati beban maksimal dapat dianggap bahwa *Web Server* tersebut sudah sampai pada batas ideal jumlah request yang mampu dilayani dengan baik oleh *Web Server*, sehingga apabila penggunaan CPU yang telah mencapai 100% (overload) akan ada kemungkinan bahwa tersebut tidak dapat melayani seluruh request yang datang sehingga otomatis akan terjadi *fail request* dari *client*.

Dari kedua hasil yang didapatkan dan dibandingkan antara *Web Server* Tunggal dengan *Web Server Load Balancing*, Penulis menarik kesimpulan bahwa, idealnya sebuah *Web Server* Tunggal sesuai dengan spesifikasi *hardware* yang telah disebutkan sebelumnya adalah mampu menangani beban maksimal 100 *request* dengan baik dalam beberapa kondisi *bandwidth* yang telah diuji. Dan untuk *Web Server Load Balancing* sesuai dengan *hardware* yang telah disebutkan sebelumnya adalah mampu menangani beban 250 *request* dengan baik dalam beberapa kondisi *bandwidth* yang telah diuji. Dari Kedua hal tersebut maka dapat juga disimpulkan bahwa ada peningkatan dalam menangani *request* dari *Web Server Load Balancing* dibanding dengan *Web Server* Tunggal dengan besarnya peningkatan lebih kurang 100%.

Berikutnya data RAM pada tabel 13 di tampilkan dalam bentuk grafik pada gambar 10 dibawah ini



Gambar 19. Penggunaan RAM per-bandwidth setiap request

Pada gambar 10 untuk penggunaan RAM dari *Web Server* Tunggal dan *Web Server Load Balancing* dapat semakin banyak terpakai selaras dengan jumlah *request* yang datang, jadi semakin banyak *request* yang datang maka akan menambah besar penggunaan RAM. Untuk besar kecilnya *bandwidth* yang digunakan dalam pengujian ini tidak berpengaruh secara signifikan atas penggunaan RAM baik untuk *Web Server* Tunggal maupun *Web Server Load Balancing*. Dan dari hasil beberapa pengujian dengan kondisi *bandwidth* dan *request*, maka Penulis dapat menyimpulkan bahwa apabila penggunaan CPU sudah

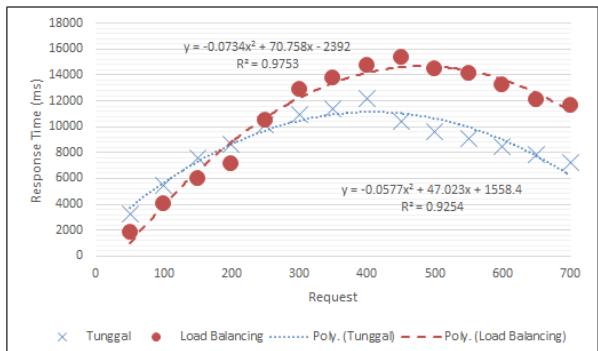
mencapai 100%, maka penggunaan RAM akan lebih banyak digunakan.

Apabila dibandingkan penggunaan RAM pada kedua Metode *Web Server* dengan penggunaan CPU pada bahasan sebelumnya, maka berbeda dengan penggunaan CPU yang mempunyai perbedaan signifikan dalam pemakaiannya terutama di 50 *request*, ini tidak terjadi dengan penggunaan RAM, walaupun dari data yang didapatkan penggunaan RAM *Web Server Load Balancing* lebih sedikit (lebih baik) dibanding dengan *Web Server* Tunggal, akan tetapi perbedaan tersebut tidak signifikan. Sehingga Penulis menyimpulkan bahwa penggunaan RAM antara *Web Server* Tunggal dan *Web Server Load Balancing* tidak berbeda jauh, walaupun sebenarnya secara fisik RAM milik *Web Server Load Balancing* mempunyai kapasitas 2 kali lipat dari RAM *Web Server* Tunggal.

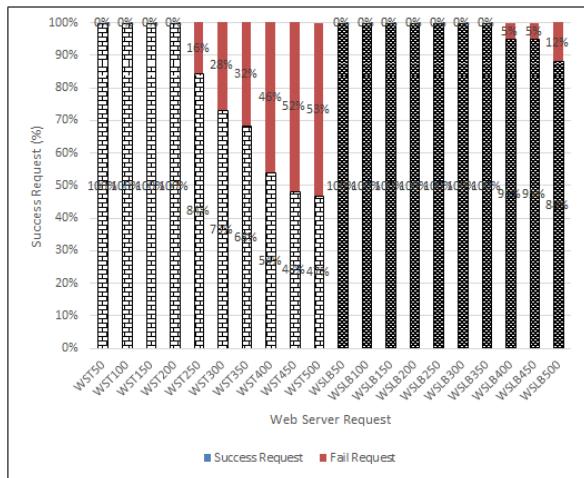
Tabel 11. Rata – rata perbandingan Response Time dengan Success Request

Web Server	Indikator	Request									
		50	100	150	200	250	300	350	400	450	500
Tunggal	Response Time (ms)	3035	5486	7586	8880	10614	9812	9158	11165	6986	6550
	Success Request (%)	100	100	100	100	84,4	73,1	63,3	54,1	48,3	47
Load Balancing	Response Time (ms)	1806	4057	5997	7075	13290	13598	13796	14714	16177	14455
	Success Request (%)	100	100	100	100	100	100	100	95	95	88,1

Selanjutnya data pada tabel 11 di tampilkan dalam bentuk grafik pada gambar 11 untuk rata-rata *response time* dan gambar 12 untuk rata-rata *success request* dibawah ini.



Gambar 20. Rata-rata Response Time



Gambar 21. Rata-rata Success Request

Pada gambar 11 dan 12 yang ditunjukkan diatas menggambarkan tentang rata-rata *response time* dan *success request* dari setiap *request* (50, 100, 150, 200, 250, 300, 350, 400, 450 dan 500), terlihat perbedaan *response time* yang antara *web server* tunggal dan *load balancing web server*, mulai dari 50 *request* sampai dengan 200 *request* terlihat bahwa *Load Balancing Web Server* lebih sedikit (lebih baik) *response time* dibanding dengan *Web Server Tunggal*. Namun pada 250 sampai dengan 500 *request* justru *Web Server Tunggal* lebih sedikit (lebih baik) *response time*-nya dibanding *Load Balancing Web Server*, akan tetapi apabila dilihat pada gambar 5.20 success request dari *Web Server Tunggal* mendapatkan *fail request* yang jauh lebih banyak bila dibanding dengan *Load Balancing Web Server*. Sebagai contoh untuk beban 250 *request*, *Web Server Tunggal* hanya dapat melayani 84,4% yang *success request* (15,6% *fail request*) sedangkan *Web Server Load Balancing* mampu melayani 100% (0% *fail request*), dan untuk *response time* *Web Server Tunggal* lebih sedikit (lebih baik) dibanding dengan *Web Server Load Balancing* yang memiliki *reponse time* lebih banyak (lebih buruk). Dilihat dari data *response time* dibanding *success request* pada beban 250 *request*, maka dapat disimpulkan bahwa *Web Server Tunggal* memiliki *response time* lebih sedikit karena mengalami sebanyak 15,6% *fail request* dari total 250 *request*, sedangkan pada *Web Server Load Balancing* tidak ada *fail request*.

Untuk pengujian dengan beban 350 *request* kedua metode *Web Server*, baik itu *Tunggal* maupun yang *Load Balancing* sudah mengalami *overload*, sehingga tidak mampu melayani semua *request* yang datang. Untuk *Web Server Tunggal* memiliki *success request* 68% (32% *fail request*) dan *response time* 9158 *milisecond*, sedangkan *Web Server Load Balancing* memiliki *success request* 95% (5% *fail request*) dan *response time* sebesar 13796 *milisecond*. Terlihat meski *response time* *Web Server Tunggal* lebih baik dibanding dengan *Load Balancing Web Server* namun jumlah *fail request* *Web Server Tunggal* jauh lebih tinggi sebesar 32% *fail request*, walaupun *Load Balancing* mengalami *fail request* namun

persentasenya jauh lebih kecil yaitu sebesar lebih kurang 5%. Dan pada pengujian beban terakhir yaitu beban 500 *request* untuk kedua metode *Web Server*, baik itu *Tunggal* maupun yang *Load Balancing*, semakin besar perbedaan *success request*-nya yaitu untuk *Web Server Tunggal* hanya 47% *success request* (53% *fail request*) sedangkan *Load Balancing Web Server* dapat melayani *request* sebesar lebih kurang 88% *success request* (12% *fail request*).

Dari hasil pengujian mulai dari 50, 100, 150, 200, 250, 300, 350, 400, 450 dan 500 *request*, setiap tahapan-tahapan dengan disertai pembatasan *bandwidth* yang digunakan, pengujian sudah menemui beban ideal dan maksimalnya masing-masing. Dari sini Penulis coba menarik kesimpulan bahwa beban CPU itu sangat berperan memberi pengaruh terhadap kesanggupan sebuah *Web Server* dalam melayani jumlah *request* yang datang, seiring dengan itu juga penggunaan RAM menjadi sangat dibutuhkan untuk membantu kinerja CPU dalam melayani *request* yang datang. Dari kedua parameter tersebut yaitu penggunaan CPU dan RAM sangat menentukan *success request* dan *response time* sebuah *Web Server*. Jadi apabila penggunaan CPU dan RAM belum maksimal, maka ada kemungkinan bahwa *Web Server* tersebut masih mampu melayani *request* yang datang, sebaliknya apabila CPU dan RAM sudah maksimal penggunaannya maka besar kemungkinan akan terjadi *fail request*.

IV. KESIMPULAN

Berdasarkan hasil pembahasan pada bab sebelumnya maka Penulis menyimpulkan beberapa hal yaitu:

1. Besar kecilnya *bandwidth* sangat berpengaruh pada *response time* *Load Balancing Web Server*, berdasarkan hasil pengujian dengan 50, 100, 150, 200, 250, 300, 350, 400, 450 dan 500 yang telah dilakukan, relatif *Load Balancing Web Server* lebih baik *response time*, kecuali pengujian dengan *bandwidth* 512 Kbps. Jadi semakin besar *bandwidth* maka akan dapat meningkatkan *response time* *Web Server*.
2. Jumlah *request* yang masuk akan mempengaruhi penggunaan CPU dan RAM, semakin banyak *request* yang ada maka semakin besar penggunaan CPU dan RAM dan apabila penggunaan sudah sangat maksimal (*overload*), maka akan menyebabkan *fail request*.
3. *Web Server Load Balancing* mampu melayani jumlah *request* yang datang lebih banyak daripada *Web Server Tunggal* dan memiliki *response time* yang lebih baik. Untuk beban 500 *request* untuk kedua metode *Web Server*, baik itu *Tunggal* maupun yang *Load Balancing*, *Web Server Tunggal* 47% *success request* (53% *fail request*) sedangkan *Load Balancing Web Server* dapat melayani *request* sebesar lebih kurang 88% *success request* (12% *fail request*). Jadi *Load Balancing Web Server* mampu melayani *request* 40 % lebih banyak dibanding *Web Server Tunggal*.

4. Berdasarkan penggunaan CPU, RAM, *Response time* dan *Fail Request*, maka Penulis mendapatkan bahwa sebuah *Web Server* Tunggal dengan spesifikasi yang telah disebutkan sebelumnya idealnya menangani *request* sebanyak 100 request, dan untuk *Load Balancing* yaitu sebanyak 200 *request*.
5. Dari masing-masing pengujian dengan berdasarkan jumlah *request* maka didapatkan persamaan regresi untuk masing-masing bandwidth yaitu:
 - a. 512k Kbps, Tunggal $y = -0,1161x^2 + 107,51x - 825,88$ dan $R^2 = 0,9751$ serta *Load Balancing* $y = -0,1845x^2 + 168,72x - 5158,6$ dan $R^2 = 0,9789$
 - b. 2 Mbps, Tunggal $y = -0,0355x^2 + 23,942x + 2589$ dan $R^2 = 0,8722$, serta *Load Balancing* $y = -0,0379x^2 + 39,319x - 1451,7$ dan $R^2 = 0,9554$.

V. SARAN

Berdasarkan penelitian yang telah dilakukan oleh Penulis, terdapat beberapa kekurangan yang menurut Penulis akan dapat dikembangkan lebih jauh di masa depan. Maka oleh karena itu Penulis memberikan saran yang harapannya dapat berguna untuk pengembangan penelitian selanjutnya, antara lain:

1. Meningkatkan spesifikasi *Hardware* pada *Web Server* seperti CPU dan RAM, sehingga dapat membandingkan jumlah *client* yang *request* dengan lebih banyak.
2. Menambah lebih banyak jumlah *Web Server* pada metode *Load Balancing* sehingga dapat dilihat lebih jauh perbedaannya.
3. Ujicoba Simulasi dapat dilakukan di internet (publik) dengan menggunakan *bandwidth* yang didapat dari penyedia internet, agar didapatkan data yang lebih *real*.
4. *Load Balancing* dapat juga berjalan di protokol selain http seperti yang Penulis teliti, contohnya FTP dan lain-lain.

Dapat menambahkan parameter/variabel pengujian yang lain, selain yang Penulis teliti, contohnya seperti *page load time*, kecepatan *download file* secara simultan dan lain-lain.

REFERENSI

- [1] Sukardi, Metodologi Penelitian Pendidikan Kompetensi dan Praktiknya, Jakarta: Bumi Aksara, 2011.
- [2] Sugiyono, Metode Penelitian kombinasi (mixed methods), Bandung: Alfabeta, 2011.