

Perbandingan Performansi Single Web Server Dan Multi Web Server Dengan Metode *Paired Sample T Test*

Muhammad Ravis^[1], Gian Muhammad^[2], Molavi Arman^[3]

Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer GI MDP^{[1],[2]}

Program Studi Manajemen Informatika Akademi Manajemen Informatika dan Komputer MDP^[3]

ravis@mhs.mdp.ac.id^[1], giant@mhs.mdp.ac.id^[2], molavi.arman@mdp.ac.id^[3]

Abstract—The more the development of the era, the more technology is also developing. Technology is very helpful for humans in carrying out daily activities. In connection with that people can use internet. It is very helpful for users in getting information and it can also be used as a communication medium. Website is a medium that is accessed through the internet. When more internet users access the web, it will cause the performance to be slow. This paper conducted research on comparing the performance of single web server and multi web servers using paired sample t-test. The performance of single and multi web servers can be seen in terms of components throughput, response time, cpu utilization by using the httpref application, then to see the difference between a single web server and multi web servers, paired sample t test was used. The result showed that the sig value obtained between testing a single web server and multi web servers is 0,000 and the value is below 0.05, so it can be concluded that between a single web server and multi web server there are significant differences from the test results.

Keywords: Performance, Server, Httpref, Web, Paired sample t-test

I. PENDAHULUAN

Internet (*Interconnected Network*) adalah kumpulan jaringan komputer di seluruh dunia yang saling berhubungan antara satu dengan yang lainnya. Internet juga didefinisikan sebagai suatu jaringan yang menghubungkan antara komputer-komputer dan jaringan komputer diseluruh dunia untuk saling berbagi data dan informasi, salah satu jenis layanan dari layanan internet yaitu *World Wide Web*. *World Wide Web* atau *Web* adalah suatu cara mengakses informasi dari sebuah internet, *web* juga bisa dikatakan sebagai media yang digunakan sebagai untuk mengakses informasi melalui internet.

Berkembangnya teknologi maka semakin banyak saja pengguna komputer untuk mengakses *website* apa saja yang ingin mereka buka, karena semakin banyaknya *request* yang meng-generate pada *website*, kemudian akan menyebabkan melambatnya sebuah performa dari *web server* tersebut dan dapat juga menyebabkan *website* tersebut menjadi *down*, maka dari itu pada penelitian kali ini yaitu ingin mengetahui performansi antara *single web server* dan *multi web server* manakah yang lebih baik digunakan dan ingin mengetahui apakah ada perbedaan signifikan antara perbandingan tersebut.

Terkait dengan penelitian yang dibuat kali ini, adapun penelitian terdahulu yang berkaitan dengan penelitian yang dibuat yang juga dijadikan salah satu referensi dalam pembuatan penelitian [1], yaitu Implementasi *Load Balancing* Menggunakan *Web Server* Metode LVS-NAT, mereka membandingkan tiga parameter dalam penelitian ini yaitu, *throughput*, *response time* dan *CPU utilization*, Dari Hasil Penelitian ini, A). Implementasi *load balancing web server* menggunakan LVS-NAT mampu meningkatkan nilai *throughput web server* dengan besaran yang cukup signifikan hingga 2 kali lipat *throughput web server* tunggal. Nilai ini diperoleh pada implementasi LVS dengan 4 *real server* menggunakan algoritma *round robin* dengan *throughput* sebesar 69,28 *replies/second*. Peningkatan *throughput web server* sejalan dengan peningkatan jumlah *real server* yang digunakan dalam LVS. *Throughput web server* dengan LVS 2 *real server* baik itu yang menggunakan algoritma *Round Robin* 53,78 *replies/second* maupun *Least Connection* 45,08 *replies/second* lebih kecil dari *throughput web server* dengan LVS 4 *real server* menggunakan algoritma *Round Robin* 69,28 *replies/second* dan *Least Connection* 47,96 *replies/second*. B). Implementasi *load balancing web server* menggunakan LVS-NAT mampu meningkatkan *response time* dan mengoptimalkan *CPU Utilization* dari *web server* namun peningkatan yang dihasilkan memiliki nilai yang tidak terlalu besar sehingga dianggap tidak terlalu signifikan. C). Implementasi LVS-NAT menggunakan algoritma *round robin* lebih handal dalam mengoptimalkan *throughput*, *CPU Utilization* dan *response time* dari *web server* jika dibandingkan dengan implementasi LVS-NAT menggunakan algoritma *least connection* [1].

Permasalahan dihadapi *single web server* adalah tidak mampu melayani permintaan *traffic* yang tinggi bila dibandingkan dengan *multi web server* yang menggunakan LVS. Pengujian ini menggunakan perhitungan *paired sample t test* sebagai landasan untuk melihat adanya perbedaan atau tidak.

Penelitian ini memberikan konfigurasi terhadap *Linux Virtual Server* yang digunakan agar mampu menghadapi kendala dengan tingginya *traffic* kunjungan terhadap sebuah *web server*.

II. KAJIAN LITERATUR

A. Least Connection

Membahas tentang *Least Connection* dimana *Least Connection* melakukan pembagian beban berdasarkan banyaknya koneksi yang sedang dilayani oleh sebuah *server*, *server* dengan koneksi yang paling sedikit akan diberikan beban berikutnya, begitu pula *server* dengan koneksi yang banyak akan dialihkan bebannya ke *server* yang lain yang bebannya lebih rendah.[2]

Persamaan *Least Connections* dibawah ini [3] :

```

Supposing that there is a server set S = {S0, S1, ..., Sn-1},
W(Si) is the weight of server Si,
C(Si) is the current connection number of server Si,

for (m = 0; m < n; m++) {
  if (W(Sm) > 0) {
    for (i = m+1; i < n; i++) {
      if (W(Si) <= 0)
        continue;
      if (C(Si) < C(Sm))
        m = i;
    }
  }
  return Sm;
}
return NULL;

```

Gambar 1. Persamaan *Least Connection*

B. NDLC

Membahas tentang *Network Development Life Cycle* (NDLC) dimana NDLC merupakan suatu metode mengembangkan atau merancang jaringan infrastruktur yang digunakan dalam mengembangkan atau merancang jaringan infrastruktur yang memungkinkan terjadinya pemantauan jaringan untuk mengetahui statistic dan kinerja jaringan.[4]

C. SPSS

Membahas tentang SPSS dimana SPSS adalah salah satu program komputer yang khusus dibuat untuk mengolah data dengan metode statistik tertentu.[5]

D. Paired Sample T Test

Membahas tentang *Paired Sample T Test* dimana Uji ini dilakukan terhadap dua sample yang berpasangan (*paired*); Sampel yang berpasangan diartikan sebagai sebuah sampel dengan subjek yang sama namun mengalami dua perlakuan atau pengukuran yang berbeda, seperti subjek A akan mendapat perlakuan I dan kemudian perlakuan II.[6]

Dasar pengambilan keputusan:

Berdasar perbandingan t hitung dengan t tabel:

- Jika Statistik Hitung (angka t output) > Statistik Tabel (tabel t), maka H0 ditolak.
- Jika Statistik Hitung (angka t output) > Statistik Tabel (tabel t), maka H0 ditolak.

t hitung dari output adalah 1,646.

Sedang statistik tabel bisa dicari pada tabel t, dengan cara:

- Tingkat signifikansi (α) adalah 10% untuk uji Dua sisi, sehingga masing-masing sisi menjadi 5%.
- df (*degree of freedom*) atau derajat kebebasan dicari dengan rumus: jumlah data - 1 atau $10-1=9$.

- uji dilakukan dua sisi karena akan diketahui apakah rata-rata sebelum sama dengan sesudah ataupun tidak, jadi bisa lebih besar atau lebih kecil, karenanya dipakai uji dua sisi, perlunya uji dua sisi bisa diketahui pula dari *output* SPSS yang menyebut adanya *two tailed test*

Berdasarkan nilai probabilitas

- Jika $Sig \geq 0.05$, maka H0 diterima
- Jika $Sig < 0.05$, maka H0 ditolak

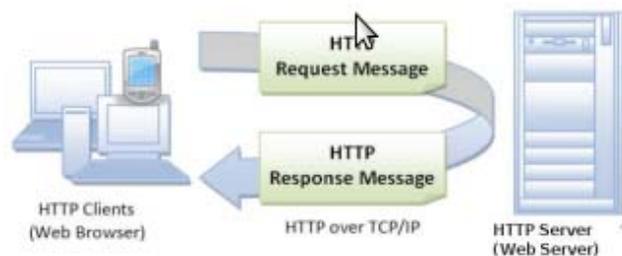
E. Web Server / HTTP

Salah satu protokol aplikasi paling populer yang digunakan di Internet adalah HTTP. HTTP adalah singkatan dari "Hypertext Transfer Protocol." HTTP adalah protokol aplikasi yang berjalan di atas protokol TCP / IP. Seluruh World Wide Web menggunakan protokol ini. Ketika *user* membuka halaman *web*, *browser* mungkin telah mengirim lebih dari 40 permintaan HTTP dan menerima respon HTTP. Header HTTP adalah bagian inti dari permintaan dan respon HTTP ini, dan membawa informasi tentang browser klien, halaman yang diminta, *server*, dan lainnya. Seperti yang diilustrasikan dalam gambar 2, klien HTTP mengirim pesan permintaan ke server HTTP. *Server*, pada gilirannya, mengembalikan pesan tanggapan. [7]

F. Linux Virtual Serveri (LVS)

Linux Virtual Server adalah solusi, untuk persyaratan. LVS adalah alat perangkat lunak yang mengarahkan koneksi jaringan ke beberapa *server* yang berbagi beban kerja mereka, yang dapat digunakan untuk membangun layanan yang sangat tersedia. Prototipe *Linux Virtual Server* sudah digunakan untuk membangun banyak situs beban berat di Internet.

LVS mengarahkan koneksi jaringan ke server berbeda sesuai dengan algoritma penjadwalan dan membuat layanan paralel *cluster* muncul sebagai layanan virtual pada satu alamat IP. Aplikasi klien berinteraksi dengan *cluster* seolah-olah itu adalah server tunggal. Klien tidak terpengaruh oleh interaksi dengan *cluster* dan tidak perlu modifikasi. Skalabilitas dicapai dengan menambahkan atau menghapus node di *cluster* secara transparan. Ketersediaan tinggi disediakan dengan mendeteksi kegagalan simpul atau daemon dan mengkonfigurasi ulang sistem dengan tepat.[8]



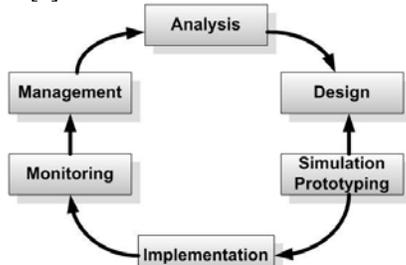
Gambar 2 Pesan Permintaan HTTP dan pesan respon

III. METODOLOGI PENELITIAN

Menyelesaikan penelitian ini digunakan sebuah metode yang digunakan untuk mendukung dalam proses pembuatan

penelitian, dimana metode yang digunakan yaitu metode NDLC (*Network Development Life Cycle*) yaitu metode pengembangan atau merancang jaringan infrastruktur yang digunakan dalam mengembangkan dan merancang jaringan infrastruktur yang memungkinkan terjadinya pemantauan jaringan untuk mengetahui statistik dan kinerja dari jaringan.

Metode NDLC ini memiliki 6 tahapan untuk penyelesaiannya yaitu 1) analisis, 2) Desain, 3) Simulasi Prototipe, 4) Implementasi, 5) *Monitoring*, dan 6) *Management*. [9]



Gambar 3. NDLC

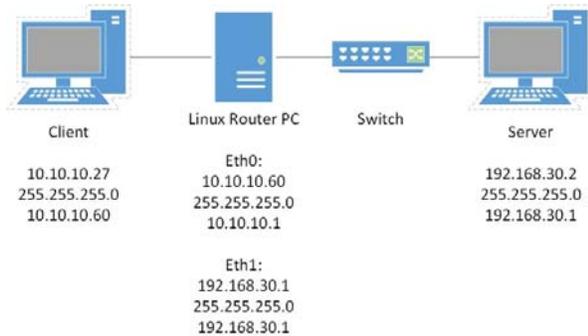
A. Analisis

Ditahap ini yaitu sebelum melakukan penelitian identifikasi dahulu apa saja yang diperlukan dan digunakan untuk merancang dan membangun dalam penelitian baik perangkat keras maupun perangkat lunak yang nantinya dapat digunakan dalam proses penelitian dan juga ditahap ini sangat membantu dan juga sangat berguna dalam memahami tema dari penelitian yang dibuat ini yaitu mengenai perbandingan antar *single web server* dan juga *multi web server*.

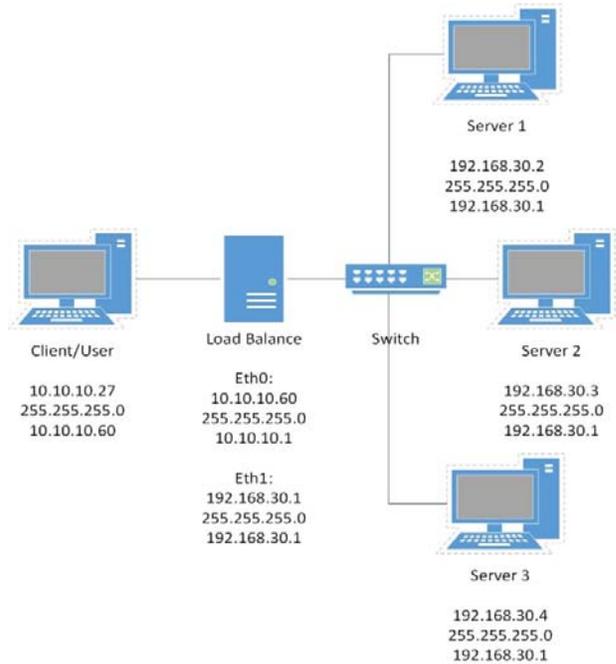
B. Desain

Desain topologi bisa berupa struktur topologi, desain akses, desain *layout* perkabelan dan sebagainya yang dimana akan memberikan gambaran jelas teradap peroyek yang akan dibangun, yang dimana hasil dari desain ini merupakan sebuah gambaran topologi, Diharapkan dengan topologi ini akan memberikan sebuah gambaran seutuhnya dari kebutuhan yang ada.

Setelah melakukan analisis tentang kebutuhan apa saja yang digunakan, kemudian ditahapan desain ini kami mendapatkan sebuah topologi yang akan digunakan dalam penelitian ini yaitu Topologi *Single web server* dan Topologi *Multi web server*.



Gambar 4. Topologi Single Web Server



Gambar 5. Topologi Multi Web Server

C. Simulasi Prototipe

Simulasi prototipe yang dimaksud disini adalah tahapan proses atau tahap penerapan yang akan dilakukan untuk penelitian yang akan dilakukan, dimana prototipe yang dibutuhkan berupa sebuah *laptop*, *switch* dan kabel LAN, sehingga tampak sebuah topologi.

D. Implementasi

Di tahapan ini akan memakan waktu lebih lama dari tahapan sebelumnya, dalam implementasinya terhadap *single web server* dan *multi web server* untuk melakukan komponen pengujian terhadap *Throughput*, *Respons Time* dan *CPU Utilization* dan kemudian ditahap selanjutnya (*Monitoring*) akan menggunakan sebuah metode perhitungan untuk melakukan perbandingan terhadap hasil yang didapat antara *single web server* dan *multi web server* dengan 3 komponen pengujian tersebut dengan menggunakan metode *Paired Samples T Test* atau Uji T Sampel berpasangan yang berguna untuk menguji perbedaan rata-rata antara dua kelompok data yang berpasangan.

Pada penelitian ini ada 2 skenario pengujian yang akan dilakukan yang pertama itu pengujian terhadap *single web server* dan juga *multi web server*, yang mana pengujian terhadap *single web server* ini tidak menggunakan *load balancing* untuk menyeimbangkan beban *traffic*, kemudian pengujian terhadap *multi web server* dengan menggunakan *load balancing* dengan implementasi algoritma penjadwalan *least connection* yang akan digunakan pada pengujianya dan menginstal *httpperf* pada pada masing masing komputer *client* untuk mendapatkan hasil dari 3 komponen pengujian tersebut, dan hasil dari pengujian tersebut didapat dari menggenerate sejumlah besar request pada *client* yang dikirim ke *server*

dengan 10 kali pengujian pada masing masing total koneksi yang akan diujiakan, dimana terdapat pada tabel 1 dibawah ini.

Tabel 1. Rancangan jumlah beban *request* pada httpperf

No	Total Koneksi	Request/Second
1	30000	3000
2	60000	6000
3	90000	9000
4	120000	12000
5	150000	15000

Pada Tabel 1, rancangan jumlah beban *request* pada httpperf adalah berupa asumsi, dikarekan untuk menunjukkan terlihat jelas perbedaan sampai dimana kemampuan *single web server* dan *multi web server* pada saat pengujian.

E. Monitoring

Setelah melakukan implementasi, tahap selanjutnya yaitu tahap *Monitoring* dimana tahap ini akan dilihat hasil dari *Paired Sample T Test* dari pengujian terhadap *single web server* dan *multi web server* yaitu komponen *Throughput*, *Response Time* dan *CPU Utilization*. Metode *Paired Samples T Test* atau Uji T Sampel berpasangan biasa digunakan untuk menguji perbedaan rata-rata antara dua kelompok data yang berpasangan, dengan hipotesisnya sebagai berikut.

Pengambilan Keputusan:

Hipotesis

- H_0 : tidak ada perbedaan signifikan antara *single web server* dan *multi web server*.
- H_1 : ada perbedaan signifikan antara *single web server* dan *multi web server*.

Kreteria Pengambilan Keputusan:

- Jika $Sig. \geq 0.05$ maka H_0 diterima
- Jika $Sig. < 0.05$ maka H_0 ditolak

F. Management

Pada tahap manajemen atau pengaturan, salah satu yang menjadi perhatian khusus adalah masalah kebijakan (*Policy*). Kebijakan perlu dibuat untuk membuat atau mengatur agar system yang telah dibangun dan berjalan dengan baik dapat berlangsung lama dan unsur *reliability* terjaga.

IV. HASIL DAN PEMBAHASAN

Dibagian ini merupakan pembahasan mengenai hasil yang didapat dari penelitian yang telah dilakukan dengan pengamatan terhadap hasil dari 3 komponen yaitu *Throughput*, *Response Time* dan *CPU Utilization* pada pengujian *single web server* dan *multi web server* yang kemudian hasilnya akan langsung dibandingkan menggunakan metode perhitungan

paired sample t test untuk mengetahui apakah ada perbedaan diantara pengujian *single web server* dan juga *multi web server* tersebut.

A. Hasil Pengujian Throughput

Hasil yang didapat, perbandingan antara *single web server* dan *multi web server*, dengan memberikan masing masing *request* yang terdiri dari 3000, 6000, 9000, 12000 dan 15000 dengan 10 kali percobaan dari setiap pengujian antara *single web server* dan *multi web server* menunjukkan untuk nilai rata rata yang dihasilkan pada pengujian *single web server* dengan jumlah *request* 3000 yaitu 960.9, 6000 yaitu 1884.64, 9000 yaitu 1558.28, 12000 yaitu 1643.82 dan 15000 yaitu 1663.71 sedangkan untuk nilai rata rata yang dihasilkan pada pengujian *multi web server* dengan jumlah *request* 3000 yaitu 962.83, 6000 yaitu 1925.6, 9000 yaitu 2888.16, 12000 yaitu 3826.46 dan 15000 yaitu 4698.43 dan dari hasil yang didapat dilihat bahwa semakin besar nilai yang dihasilkan maka semakin bagus kinerja dari *server* tersebut dan dapat dikatakan bahwa kinerja dari *multi web server* lebih baik dari pada *single web server*.

B. Hasil Pengujian Response Time

Hasil yang didapat, perbandingan antara *single web server* dan *multi web server*, dengan memberikan masing masing *request* yang terdiri dari 3000, 6000, 9000, 12000 dan 15000 dengan 10 kali percobaan dari setiap pengujian antara *single web server* dan *multi web server* menunjukkan untuk nilai rata rata yang dihasilkan pada pengujian *single web server* dengan jumlah *request* 3000 yaitu 2.09, 6000 yaitu 30.58, 9000 yaitu 71.11, 12000 yaitu 66.47 dan 15000 yaitu 60.52 sedangkan untuk nilai rata rata yang dihasilkan pada pengujian *multi web server* dengan jumlah *request* 3000 yaitu 1.06, 6000 yaitu 2.31, 9000 yaitu 2.78, 12000 yaitu 2.95 dan 15000 yaitu 6.23 dan dari hasil yang didapat dilihat semakin kecil waktu respon yang didapat maka semakin bagus juga *web server* untuk menanggapi *request* dari *client* dan semakin cepat pula *web server* tersebut dapat di akses.

C. Hasil Pengujian CPU Utilization

Dari hasil yang didapat, perbandingan antara *single web server* dan *multi web server*, dengan memberikan masing masing *request* yang terdiri dari 3000, 6000, 9000, 12000 dan 15000 dengan 10 kali percobaan dari setiap pengujian antara *single web server* dan *multi web server* menunjukkan untuk nilai rata rata yang dihasilkan pada pengujian *single web server* dengan jumlah *request* 3000 yaitu 1.591, 6000 yaitu 0.579, 9000 yaitu 0.401, 12000 yaitu 0.405 dan 15000 yaitu 0.453 sedangkan untuk nilai rata rata yang dihasilkan pada pengujian *multi web server* dengan jumlah *request* 3000 yaitu 1.762, 6000 yaitu 1.056, 9000 yaitu 0.833, 12000 yaitu 0.721 dan 15000 yaitu 0.747 dan dari hasil yang didapat dari hasil yang didapat dilihat bahwa pengujian terhadap *single web server* tanpa menggunakan *load balancing* jumlah penggunaan CPU Utilizationnya lebih banyak dibandingkan dengan pengujian terhadap *multi web server* yang menggunakan *load balancing*, jadi dapat dikatakan bahwa semakin kecil nilai persentasenya maka semakin dikit *source* yang digunakan.

Tiga komponen pengujian yang telah dilakukan dapat disimpulkan bahwa hasil dari pengujian terhadap *single web server* dan *multi web server* menunjukkan bahwa *multi web server* lebih baik digunakan pada sisi *Throughput* dan *Response Time* dan untuk *single web server* bahwa hanya untuk *CPU Utilizationnya* saja.

Setelah mendapatkan hasil dari pengujian terhadap masing-masing komponen pada pengujian *single web server* dan *multi web server*, maka selanjutnya hasil tersebut akan dibandingkan dengan menggunakan metode perhitungan *paired sample t test* dengan menggunakan aplikasi SPSS untuk

mengetahui uji beda apakah pada hasil pengujian terhadap *single web server* dan *multi web server* terdapat perbedaan atau tidak.

Untuk mendukung pengambilan keputusan yang telah dijelaskan pada metode NDLC tahap kelima (*Monitoring*) yang mana akan melihat nilai *sig* hasil dari pengujian *single web server* dan *multi web server* apakah ada atau tidak adanya perbedaan signifikan antara 3 komponen *single web server* dan *multi web server* tersebut, dan hasil tabelnya dapat dilihat pada tabel dibawah ini.

Tabel 2. Paired Samples Test Throughput 30000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-1,93000	,04830	,01528	-1,96456	-1,89544	-126,348	9	,000

Tabel 3. Paired Samples Test Throughput 60000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-40,96000	71,69344	22,67146	-92,24640	10,32640	-1,807	9	,104

Tabel 4. Paired Samples Test Throughput 90000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-1229,88000	10,71518	3,38844	-137,54518	-1222,21482	-362,964	9	,000

Tabel 5. Paired Samples Test Throughput 120000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-2182,64000	40,09229	12,67829	-2211,32030	-2153,95970	-172,156	9	,000

Tabel 6. Paired Samples Test Throughput 150000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-3034,72000	126,82628	40,10599	-3125,44606	-2943,99394	-75,667	9	,000

Dapat dilihat bahwa dari tabel 2 – 6, hasil yang didapat dari komponen pengujian *Throughput* antara *single web server* dan *multi web server* yang menunjukkan nilai sig pada koneksi 30000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 60000 yaitu 0.104 dan nilai tersebut \geq 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* ada perbedaan signifikan, pada koneksi 90000 yaitu 0.000 dan

nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 120000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, dan pada koneksi 150000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan.

Tabel 7. Paired Samples Test Response Time 30000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	1,03000	,29078	,09195	,82199	1,23801	11,201	9	,000

Tabel 8. Paired Samples Test Response Time 60000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	28,27000	15,75063	4,98079	17,00268	39,53732	5,676	9	,000

Tabel 9. Paired Samples Test Response Time 90000 Koneksi

Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
			Lower	Upper			
			68,33000	6,09791			

Tabel 10. Paired Samples Test Response Time 120000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	63,52000	6,12205	1,93596	59,14055	67,89945	32,811	9	,000

Tabel 11. Paired Samples Test Response Time 150000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	54,29000	2,43194	,76905	52,55029	56,02971	70,594	9	,000

Dapat dilihat bahwa dari tabel 7 – 11, hasil yang didapat dari komponen pengujian *Response Time* antara *single web server* dan *multi web server* yang menunjukkan nilai sig pada koneksi 30000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 60000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 90000

yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 120000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, dan pada koneksi 150000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan.

Tabel 12. Paired Sample Test CPU Utilization 30000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-,17100	,12096	,03825	-,25753	-,08447	-4,470	9	,002

Tabel 13. Paired Samples Test CPU Utilization 60000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-,47700	,08603	,02720	-,53854	-,41546	-17,534	9	,000

Tabel 14. Paired Samples Test CPU Utilization 90000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-,43200	,03736	,01181	-,45872	-,40528	-36,569	9	,000

Tabel 15. Paired Samples Test CPU Utilization 120000 Koneksi

	Paired Differences				t	df	Sig. (2-tailed)	
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower				Upper
Pair 1 Single - Multi	-,31600	,08030	,02539	-,37345	-,25855	-12,444	9	,000

Tabel 16. Paired Samples Test CPU Utilization 150000 Koneksi

Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
			Lower	Upper			
			-,29400	,06883			

Dapat dilihat bahwa dari tabel 12 – 16, hasil yang didapat dari komponen pengujian *CPU Utilization* antara *single web server* dan *multi web server* yang menunjukkan nilai pada

koneksi 30000 yaitu 0.002 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 60000

yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 90000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, pada koneksi 120000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan, dan pada koneksi 150000 yaitu 0.000 dan nilai tersebut < 0.05 dan dapat dinyatakan bahwa antara *single web server* dan *multi web server* tidak ada perbedaan signifikan

V. KESIMPULAN

Berdasarkan hasil penelitian ini dapat disimpulkan sebagaimana berikut ini:

- Untuk membandingkan performansi antara *single web server* dan *multi web server* dapat dilihat dari segi komponen yang diteliti yaitu *Throughput*, *Response Time* dan *CPU Utilization*, dari tiga komponen tersebut dapat dibandingkan untuk mengetahui performansi antara *single web server* dan *multi web server* yang manakah yang lebih baik. Berdasarkan penelitian yang telah dilakukan bahwa hasil yang didapat dengan pengujian dengan menggunakan *paired sample t test* menunjukkan dari setiap perbandingan performansi antara *single web server* dan *multi web server* dengan masing masing pengujian yaitu 3000, 6000, 9000, 12000 dan 15000 pada setiap pengujian *Throughput*, *Response Time*, dan *CPU Utilization* yang mana apabila nilai signifikannya lebih kecil dari 0,05 maka artinya H_0 ditolak, dan secara otomatis H_1 yang diterima, maka dapat dikatakan hasil penelitian tersebut ada perbedaan atau perbandingan performansi yang signifikan antara hasil *single web server* dan *multi web server*, tetapi hanya pada pengujian *Throughput* yang ke 6000 yang mana hasil yang di dapat yaitu 0,104, nilai tersebut dikatakan tidak terdapat perbedaan signifikan karena nilai signifikannya yang didapat diatas 0,05.
- Perbandingan antara *single web server* dan *multi web server* yang paling baik digunakan dari segi komponen *Throughput*, *Response Time* dan *CPU Utilization* adalah *multi web server* karena dari nilai rata – rata dari setiap pengujian *throughput*, *response time* dan *cpu utilization* menunjukkan bahwa *multi web server* lebih baik dari *single web server*.
- Penelitian ini memiliki peluang untuk dikembangkan lagi atau dikombinasikan dalam ruang lingkup virtualisasi komputer.

DAFTAR PUSTAKA

- [1] J. A. Rabu, J. Purwadi, and W. S. Raharjo, "Implementasi Load Balancing Web Server Menggunakan Metode LVS-NAT," *Jurnal Informatika*, vol. 8, no. 2, Nov. 2013.

- [2] Wahana, K, "Administrasi Jaringan dengan Linux Ubuntu 11", Andi, 2011.
- [3] H. Nasser and T. Witono, "Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler," *Jurnal Informatika*, vol. 12, no. 1, 2016.
- [4] Linux Virtual Server. "Least-Connection Scheduling." Internet: http://kb.linuxvirtualserver.org/wiki/Least-Connection_Scheduling, 2005, [July, 24, 2018]
- [5] Prabowo, R.T. and Kurniawan, M.T, "Analisis dan Desain Keamanan Jaringan Komputer dengan Metode Network development Life Cycle" *Jurnal Rekayasa Sistem dan Industri*, vol 2, no.1, Jan. 2015
- [6] Singgih ,S, " Menguasai SPSS 22 *From Basic To Expert Skills* ", Elex Media Komputindo, 2015.
- [7] Suroto, Suroto. "A Review of Defense Against Slow HTTP Attack." *JOIV : International Journal on Informatics Visualization*1, no. 4 (April 2017): 127. <https://doi.org/10.30630/joiv.1.4.51>.
- [8] Zhang W. Linux virtual server for scalable network services. In *Ottawa Linux Symposium 2000 Jul 19 (Vol. 2000)*.
- [9] Singgih ,S, " *SPSS 22 From Essential to Expert Skills* ", Elex Media Komputindo, 2014.