

Analisa Kinerja Web Server *E-learning Menggunakan Apache Benchmark dan Httperf*

Molavi Arman

STMIK GI MDP Palembang

Jalan Rajawali No.14

Pos-el: molavi.arman@mdp.ac.id

Abstrak

Penelitian bertujuan untuk menguji 1) Jika beban traffic tinggi maka akan sangat mempengaruhi kinerja web server tersebut ketika diakses. 2) Kemampuan Web Server dipengaruhi konfigurasi perangkat lunak pada IT Infrastruktur. Untuk melihat ada tidaknya perbedaan antara sebelum dan sesudah konfigurasi pada web server dilakukan dengan uji beda paired sample test. Pada tes uji tingkat performance penulis menggunakan tools berupa httperf dan ab (apache benchmark) sebagai uji beban serta teknik analisis yang digunakan adalah Independent Sample Test (riset pengujian) terhadap dua sample. Hasil penelitian menunjukkan bahwa dalam pengujian terlihat perbedaan antara sebelum dan sesudah konfigurasi pada web server, perbedaan yang sangat terlihat pada bagian response time, mengacu pada seberapa cepat suatu sistem/aplikasi dapat merespon suatu aksi, mengalami peningkatan yang sebabkan hasil konfigurasi. Kemudian pada komponen Throughput, menunjukkan banyaknya tugas yang bisa dilakukan dalam satu satuan waktu, mengalami peningkatan setelah dikonfigurasi. Selanjutnya pada komponen Resource Utilization, menunjukkan tingkat penggunaan sumber daya perangkat keras tertentu, mengalami peningkatan setelah dikonfigurasi. Disarankan 1) untuk perangkat keras perlu tingkatkan kemampuannya seperti memory dan kapasitas penyimpanan, 2) Untuk teknologi jaringan perlu ditingkatkan dengan kecepatan gigabit., dan 3) Perlunya sumber daya manusia yang melakukan maintenance berkala.

Kata Kunci: Kinerja Web Server *E-learning*, paired sample test / Independent Sample Test, response time, throughput, resource utilization.

Abstract

The research aimed to examine 1) if traffic load was high, it would influence the performance of web server when accessed, 2) web server performance was influenced by the configuration of software in IT infrastructure. To see whether or not there was difference between pre- and post-configuration of the web server, paired sample test was used. The writer used httperf for performance test, and ab (apache benchmark) for load test. The technique analysis used was Independent sample test between two samples. The result showed that there was significant difference between pre and post configuration of the web server. There was a significant difference at response time, referring to how fast a system/application could respond an action, due to the configuration. Then, the result showed that there was an increase at throughput component, referring to the tasks that could be completed in a time. Next, there was also an increase at the resource utilization, referring to the use of certain resources. It is suggested 1) for hardware it is necessary to increase the performance of such as memory and storage capacity, 2) for network technology it is necessary to increase the speed of gigabyte, 3) it also necessary to do maintenance periodically by the human resources.

Key Word: Performance Web Server *E-learning*, paired sample test / Independent Sample Test, response time, throughput, resource utilization.

1. PENDAHULUAN

Sebagai sebuah media pembelajaran, *e-learning* memiliki kendali penuh atas akses informasi yang terkandung didalamnya dari sisi pengguna / user. Pengguna menentukan sendiri apakah pengguna mau mengakses sebuah halaman *web* dan kemudian melakukan pencarian atas informasi yang ada di dalamnya atau tidak. Dalam menghasilkan *e-learning* yang menarik dan diminati membutuhkan tiga prasyarat yaitu sederhana, persona, dan cepat.[1] Faktor desain yang menjadi salah satu hal yang menyebabkan

seorang pengguna mau terus mengakses halaman *web*. Setelah pengguna tertarik dengan tampilan yang ada, maka selanjutnya pengguna akan banyak menghabiskan waktunya untuk mencari informasi yang ada di dalam *web*. Bila pengguna merasakan bahwa ia terlalu lama mengakses suatu halaman *web* maka kecondongan untuk enggan mengakses akan ada demikian pula saat pengguna sedang mengakses halaman tertentu.

Beberapa penerapan *e-learning* yang sudah ada lebih menekankan pada aspek konten/isi tanpa memperhatikan aspek teknis. Hal ini kadang menimbulkan kejadian bahwa infrastruktur *e-learning* yaitu *web server* sering dianggap menjadi suatu hal

yang sepele. Sebagaimana kita ketahui bahwa *web server* sebagai infrastruktur aplikasi basis web menyediakan layanan untuk setiap permintaan halaman *web* untuk tiap akses per satuan waktu. Penerapan konfigurasi *web server* yang hanya berdasarkan konfigurasi dasar tanpa mempertimbangkan segi yang lain memang menjadi suatu hal yang mudah, namun sejalan dengan waktu dimana tingkat akses yang semakin tinggi dan bertambahnya konten yang ada untuk aplikasi *web* (*e-learning*) membutuhkan kajian atau tinjauan ulang atas konfigurasi yang telah dilakukan.

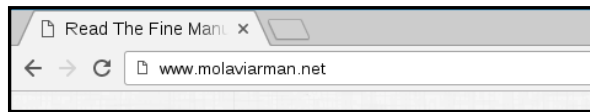
Berdasarkan pemikiran itulah maka penulis mencoba untuk mengkaji hal-hal yang berkaitan dengan aspek teknis atas sebuah *web server* dengan konten *e-learning*, dimana selain isi yang menarik untuk dibaca aspek teknis yaitu kecepatan dan waktu respon yang dihasilkan menjadi ideal dalam kondisi tertentu.

1.1 Tinjauan Pustaka

Web server yang digunakan adalah *Apache*, tugas utama dari *Web server* adalah menerjemahkan permintaan ke dalam respon yang cocok untuk keadaan pada saat itu, ketika klien membuka komunikasi dengan *Apache*, *Apache* mengirimkan permintaan untuk sumber daya. *Apache* menyediakan sumber daya yang baik atau memberikan respon alternatif untuk menjelaskan mengapa permintaan tidak dapat terpenuhi. Dalam banyak kasus, sumber daya adalah *Hypertext Markup Language (HTML)* halaman *web* yang berada pada *disk* lokal, tetapi ini hanya pilihan sederhana. Hal ini dapat berupa *file* gambar, hasil dari sebuah *script* yang menghasilkan *output HTML*, *applet Java* yang diunduh dan dijalankan oleh klien, dan seterusnya, *Apache* menggunakan *HTTP* untuk berbicara dengan klien. Permintaan / tanggapan protokol ini, yang berarti bahwa ia mendefinisikan bagaimana membuat permintaan klien dan bagaimana *server* menanggapi mereka. Setiap komunikasi *HTTP* dimulai dengan permintaan dan berakhir dengan jawaban. *Executable Apache* mengambil nama dari protokol, dan pada sistem *Unix* umumnya disebut *httpd*, kependekan *daemon HTTP* [2].

Cara kerja koneksi *web server* dan klien, *web* atau *World Wide Web (WWW)* merupakan salah satu layanan Internet yang paling populer, bisa dikatakan *Web* adalah “wajahnya Internet”, berbagai hal dapat “ditampilkan” di halaman *Web*, mulai dari teks, gambar, video, musik, dan sebagainya. Protokol bernama *HTTP (Hyper Text Transfer Protocol)* bertanggung jawab menangani proses komunikasi antara *client Web* (*browser* atau *Web browser*) dan *server Web*. Ketika user mengetikkan alamat *server web* atau situs pada kotak *Address* atau *URL (Uniform Resource*

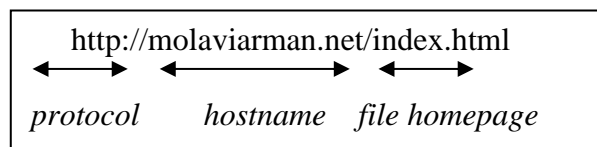
Locator) dan menekan [*ENTER*] maka *Web Browser* akan mencoba membentuk koneksi ke sebuah *server Web*. *Browser* meminta (*request*) sebuah halaman *Web* dan kemudian menampilkan hasilnya[3].



Gambar 1. *Browser*

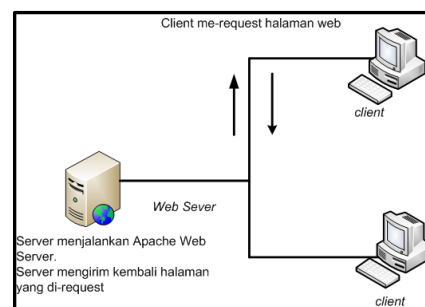
Bagaimana koneksi dilakukan dan apa yang terjadi selama proses dapat dijelaskan dengan sederhana sebagai berikut :

1. *Web browser* memecah *URL* menjadi 3 bagian, yaitu :
 - a. Bagaimana protokol (misal: *http, ftp, dan sebagainya*)
 - b. *Hostname Server Web* (misal: *slackware.com*)
 - c. Nama *file home page* (misal: *index.html*)



Gambar 2. Bagian *URL*

2. *Web browser* berkomunikasi dengan *resolver* (mencari *server DNS*), meminta informasi *IP address*, setelah *IP address* diketahui selanjutnya *IP address* ini digunakan untuk koneksi dengan *server Web*.
3. *Web browser* kemudian membentuk koneksi dengan *server* menggunakan *IP address* dan *port 80* (standar *port* untuk *http*).
4. *Web browser* mengirimkan *request GET* ke *server Web*, menanyakan *file home page* (dalam hal ini *index.html*).
5. *Server Web* mengirimkan halaman *Web* yang diminta dalam bentuk *tag-tag HTML* kepada *Web browser*.
6. *Web browser* membaca *tag-tag HTML*, kemudian menampilkan dalam keadaan sudah diformat. *Server Web* memutuskan koneksi dengan *client*.
7. *Server web* memutuskan koneksi dengan *client*.



Gambar 3. Analogi acara kerja *Web Server*

1.2 Uji Performansi *Web Server*

Beberapa pendapat ahli tentang uji performansi adalah sebagai berikut:

- 1) Keseluruhan performansi dari *web* dipengaruhi oleh klien, *web server*, dan kapasitas jaringan yang menghubungkan klien ke *web server* [4].
- 2) Uji performansi terhadap sistem *web server* yang ada menentukan seberapa baik performanya, sehingga dapat dilakukan suatu perubahan-perubahan untuk meningkatkan performansinya [5]
- 3) Ada tiga cara untuk sebuah website menangani trafik yang tinggi, pertama adalah *mirroring*, *distributed caching* dan melakukan performansi *web server* [6]
- 4) Proses performansi *web server* dapat dibagi menjadi dua kategori yaitu : yang menguji *web server* dalam penggunaan secara nyata dan yang menguji *web server* menggunakan *load generator* [7].

[8] Uji performansi memiliki bentuk-bentuk tersendiri sebagai berikut:

- 1) *Performance Test*
Uji performansi digunakan untuk menguji setiap bagian dari suatu *web server* untuk menemukan teknik terbaik untuk mencapai optimasi ketika trafik *web* meningkat
- 2) *Load Test*
Load test dilakukan dengan pengujian *website* menggunakan estimasi trafik dari sebuah *website* yang mampu dilayani. Caranya adalah mendefinisikan waktu maksimum sebuah halaman *web* dimuat. Pada akhir pengujian dilakukan perbandingan seberapa maksimum waktu yang dibutuhkan untuk membuka halaman *web* pada sebuah *web server*.
- 3) *Stress Test*
Stress test adalah berupa simulasi serangan "*brute force*" yang menjalankan muatan atau permintaan secara berlebihan menuju *web server*. Tujuan *stress test* adalah untuk estimasi muatan maksimum sebuah *web server* sanggup menanganinya.

Dalam pengujian performansi *web server*, tujuan utamanya adalah mengetahui tingkat kejenuhan dari *web server*. Komponen yang dibutuhkan dalam pengujian *web server* adalah sebagai berikut:

- 1) Sebuah *server* yang menjalankan perangkat lunak *web server*.
- 2) Satu atau lebih klien yang menjalankan perangkat lunak pembangkit paket

- 3) Jaringan yang menghubungkan klien dengan *server*.

1.4 Kriteria Penerimaan Kinerja

[9] Umumnya ada tiga hal yang menjadi sumber perhatian dalam suatu uji kinerja:

1. *Response time* mengacu pada seberapa cepat suatu sistem/aplikasi dapat merespon suatu aksi. Misalnya, halaman *Web* harus bisa tampil kurang dari lima detik.
2. *Throughput* menunjukkan banyaknya tugas yang bisa dilakukan dalam satu satuan waktu. Misalnya, *Web server* harus bisa melayani minimal 10 pengunjung dalam satu detik.
3. *Resource Utilization* menunjukkan tingkat penggunaan sumber daya perangkat keras tertentu. Misalnya, pada saat jam sibuk *Web server* tidak boleh memakan *CPU time* lebih dari 20%.

Perencanaan dan perancangan uji kinerja menentukan skenario yang diinginkan, menentukan variasi yang digunakan, menentukan data untuk pengujian.

2. PEMBAHASAN

2.1 Alat Analisis

Alat analisis yang digunakan adalah berupa perangkat lunak, penulis menggunakan *httperf* dan *ab* (*apache benchmark*) sebagai perangkat lunak uji yang *open source* [10].

2.2 Metodologi

Metodologi yang digunakan adalah deskriptif kualitatif untuk mengidentifikasi masalah, harus dilakukan analisis terhadap kinerja *web server* yang dinilai berdasarkan, *Response Time*, *Throughput* dan *Resource Utilization* [10].

Adapun Definisi Operasional untuk kinerja *web server* dapat dijelaskan pada tabel berikut:

Tabel 1 Definisi Operasional Kinerja *Web Server*

<i>Response Time</i>	mengacu pada seberapa cepat suatu sistem/aplikasi dapat merespon suatu aksi. Misalnya, halaman <i>Web</i> harus bisa tampil kurang dari lima detik.
<i>Throughput</i>	menunjukkan banyaknya tugas yang bisa dilakukan dalam satu satuan waktu. Misalnya, <i>Web server</i> harus bisa melayani minimal 10 pengunjung dalam satu detik
<i>Resource utilization</i>	menunjukkan tingkat penggunaan sumber daya perangkat keras tertentu. Misalnya, pada saat jam sibuk <i>Web server</i> tidak boleh memakan <i>CPU time</i> lebih dari 20%.

2.3 Teknik Analisis

Dengan menggunakan uji *Independent Sample Test* (riset pengujian) terhadap dua sample yang berpasangan diartikan sebagai sebuah sampel dengan subjek yang sama, namun mengalami dua perlakuan atau pengukuran yang berbeda. Alat uji untuk mengukur kinerja (riset eksplorasi) menggunakan *htperf* dan *Apache Benchmark (AB)*, guna melihat kemampuan *web server*. Untuk variable konfigurasi menggunakan teknik analisis deskriptif kualitatif.

[9] Metode yang digunakan dalam pengujian ini menggunakan tahapan-tahapan seperti berikut yaitu:

1. Tahap pertama, Studi Pustaka. Mencari dan memahami literatur mengenai kinerja web server, terutama mengenai pendekatan-pendekatan dan teknik-teknik dalam menilai kinerja web server.
2. Tahap kedua, Identifikasi Lingkungan Pengujian. Mencari informasi mengenai apa saja yang digunakan dalam pengujian pada penelitian ini. Ini meliputi perangkat keras, perangkat lunak, dan tools khusus lainnya.
3. Tahap ketiga, Identifikasi kriteria penerimaan kinerja. Menentukan kriteria kinerja web server.
4. Tahap keempat, Perencanaan dan Perancangan Pengujian. Membuat rencana dan rancangan dari skenario pengujian. Ini juga meliputi data yang digunakan untuk pengujian.
5. Tahap kelima, Konfigurasi Lingkungan Pengujian. Menyiapkan lingkungan yang digunakan untuk pengujian. Beberapa diantaranya adalah topologi jaringan, konfigurasi sistem operasi, dan konfigurasi tools untuk pengujian.
6. Tahap keenam, Implementasi Rancangan Pengujian. Melakukan pembuatan scripts untuk melakukan pengujian sesuai dengan apa yang sudah direncanakan dan dirancang.
7. Tahap ketujuh, Pengujian, berisi aktivitas pengujian itu sendiri.
8. Tahap kedelapan, Analisis, Laporan, dan Uji Ulang. Mengumpulkan data-data hasil pengujian dan melakukan analisis. Uji ulang dilakukan jika ada hasil yang dirasa kurang sesuai dengan yang diharapkan.
9. Tahap kesembilan, Kesimpulan. Menarik kesimpulan dari hasil analisis yang dilakukan pada tahap kesembilan.

2.4 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini yaitu kinerja *web server elearning* yang dapat diuraikan sebagai berikut :

- a. Kinerja *web server elearning* adalah kemampuan *web* bekerja dalam menangani proses komunikasi antar *client web browser* dan *server web* [3].

Tabel 2 Butir-butir indikator variabel kinerja *web server elearning*

Variabel	Butir-butir Indikator
Kinerja <i>Web server Elearning</i>	<ol style="list-style-type: none"> 1. <i>Response Time</i> 2. <i>Throughput</i> 3. <i>Resource Utilization</i>

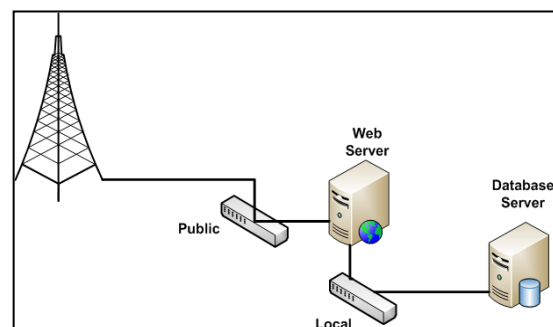
2.5 Uji Hipotesis

[11] Alat analisis yang digunakan untuk menguji hipotesis mengenai ada tidaknya perbedaan sesudah dan sebelum konfigurasi dalam meningkatkan kinerja *web server elearning* sebagai berikut.

- a. Jika probabilitas $> 0,05$, maka H_0 diterima. Artinya tidak ada perbedaan sesudah dan sebelum konfigurasi dalam meningkatkan kinerja *web server elearning*.
- b. Jika probabilitas $< 0,05$, maka H_0 ditolak. Artinya ada perbedaan sesudah dan sebelum konfigurasi dalam meningkatkan kinerja *web server elearning*.

2.6 Gambaran Umum Objek Penelitian

IT Infrastruktur berupa komputer , *switch* , kabel jaringan , digambarkan dalam topologi seperti dibawah ini



Gambar 4 Topologi Jaringan

Dari topologi diatas terdapat 2 komputer sebagai *web server* dan *database server* dan 2 *public switch* dan *local* , digunakan untuk memisahkan jaringan luar dan dalam.

Sistem Operasi yang digunakan adalah *Slackware Linux 14.1* dengan versi *apache 2.4.20* dan *mariadb 10.0.25* serta menggunakan LMS (*Learning Management System*) berupa *moodle 3.1*

2.7 Hasil Dari Proses Pengujian / Aktifitas Penelitian

Aktifitas yang pertama kali dilakukan adalah menyediakan IT Infrastruktur berupa 2 komputer , 2 switch , kabel jaringan, tang cramping , network cable tester dan RG 45.

Kemudian melakukan instalasi sistem operasi Slackware Linux dengan konfigurasi default yang dilanjutkan uji coba dengan tools Apache Benchmark dan Httpperf.

Meningkatnya kinerja web server adalah dikarenakan dari topologi didisain dengan memisahkan server web dan database untuk menghindari proses dalam satu mesin.

Meningkatkan kinerja web server juga didukung dengan menjalankan file tuning-primer.sh , dimana dari file tersebut ada alokasi pembagian memory dan thread yang dapat dilihat pada tabel dibawah ini :

Tabel 3 Pengalokasian memory

```

MEMORY USAGE
Max Memory Ever Allocated : 20 M
Configured Max Per-thread
Buffers : 405 M
Configured Max Global Buffers :
17 M
Configured Max Memory Limit :
423 M
Physical Memory : 995 M
Max memory limit seem to be
within acceptable norms
    
```

Terlihat perbedaan sebelum dan setelah konfigurasi dengan keluaran sebagai berikut

Tabel 4 Ringkasan Hasil AB Sebelum Konfigurasi

```

ab -n 100 -c 5 -t 60 http://elearningnetworking.net/
> n100-c5
    
```

AB-LAN	Complete requests	Requests per second/s	Transfer rate kb/s
hari-1	491	8.15	231.71
hari-2	498	8.30	235.82
hari-3	471	7.85	222.63
hari-4	478	7.97	225.93
hari-5	413	6.85	195.74

Tabel 5 Ringkasan Hasil Httpperf Sebelum Konfigurasi

```

httpperf --server=elearningnetworking.net --rate=10 --
    
```

```

num-conns=200 --timeout 60 > hasil-200
httpperf-LAN
    
```

Hari	connection	Request	replies	test-duration/sec
hari-1	200	200	200	24.583
hari-2	200	200	200	24.583
hari-3	200	200	200	24.689
hari-4	200	200	200	24.712
hari-5	200	200	200	24.678

Tabel 6 Ringkasan Hasil AB Sesudah Konfigurasi

```

ab -n 100 -c 5 -t 60 http://elearningnetworking.net/
> n100-c5
    
```

AB-LAN	Complete requests	Requests per second/s	Transfer rate kb/s
hari-1	1339	22.28	632.87
hari-2	1271	21.18	600.64
hari-3	1384	23.06	654.66
hari-4	1171	19.45	552.40
hari-5	1130	18.81	533.91

Tabel 7 Ringkasan Hasil Httpperf Sesudah Konfigurasi

```

httpperf --server=elearningnetworking.net --rate=10 --num-
conns=200 --timeout 60 > hasil-200
    
```

httpperf-LAN	connection	Request	replies	test-duration/sec
hari-1	200	200	200	20.084
hari-2	200	200	200	20.054
hari-3	200	200	200	20.052
hari-4	200	200	200	21.163
hari-5	200	200	200	20.087

Berdasarkan ke dua hasil AB diatas terlihat perbedaan dimana hasil sebelum konfigurasi complete request 491 sedangkan setelah konfigurasi complete request 1339. Hal ini berarti kinerja web server yang baik adalah setelah dikonfigurasi.

2.7.1 Hasil dari Response Time, Throughput , Resource Utilization dan Performance

Hasil dari Respon Time

Berdasarkan hasil sebelum dan sesudah konfigurasi didapat data sebagai berikut :

Tabel 8 Hasil *Respon Time*

Hari	conn ectio n	Req uest	replies	test- duration/sec (sebelum)	test- durati on/sec (sesud ah)
1	200	200	200	24.583	20.084
2	200	200	200	24.583	20.054
3	200	200	200	24.689	20.052
4	200	200	200	24.712	21.163
5	200	200	200	24.678	20.087

Berdasarkan tabel diatas terlihat perbedaan sebelum dan sesudah konfigurasi dimana terlihat sebelum konfigurasi kecepatan rata-rata 24.649/sec, sedangkan hasil sesudah konfigurasi kecepatan rata-rata 20.288/sec. Hal ini berarti kinerja web server yang baik adalah setelah dikonfigurasi.

Hasil *Throughput*

Berdasarkan hasil uji sebelum dan sesudah konfigurasi di diperoleh data sebagai berikut :

Tabel 9 Hasil *Throughput*

ab -n 100 -c 5 -t 60 http://elearningnetworking.net/ > n100-c5						
Sebelum			Sesudah			
A	Req	Trans	Com	Reques	Transf	
B	Co	per	plete	ts per	er rate	
-	mpl	rate	reque	second/	er rate	
L	este	sec	sts	s	kb/s	
A	reque	nd/s	rate	sts	second/	er rate
N	ests	nd/s	kb/s	sts	s	kb/s
			231.7			
1	491	8.15	1	1339	22.28	632.87
			235.8			
2	498	8.30	2	1271	21.18	600.64
			222.6			
3	471	7.85	3	1384	23.06	654.66
			225.9			
4	478	7.97	3	1171	19.45	552.40
			195.7			
5	413	6.85	4	1130	18.81	533.91

Berdasarkan tabel diatas terlihat perbedaan sebelum dan sesudah konfigurasi dimana terlihat sebelum konfigurasi *complete request* rata-rata sebesar 470, request/sec rata-rata sebesar 7.82 dan *transfer rate* rata-rata sebesar 222.37 kbs/sec. sedangkan hasil sesudah konfigurasi *complete request* rata-rata sebesar 1259, request/sec rata-rata sebesar 20,96 dan *transfer rate* rata-rata sebesar 594.90 kbs/sec. Hal ini berarti kinerja web server yang baik adalah setelah dikonfigurasi.

Hasil *Performance*

Untuk menjawab hipotesis tentang ada tidaknya perbedaan antara beban traffic yang tinggi terhadap web server maka digunakan uji *Independent Sample Test* dengan hasil sebagai berikut:

Tabel 10 Hasil Uji *Independent Sample Test*

		testduration	
		Equal variances assumed	Equal variances not assumed
Levene's F		5.688	
Test for Equality of Variance	Sig.	.044	
t-test for Equality of Means	T	-32.31539	-32.31539
	Df	8	4.045
	Sig. (2-tailed)	.000	.000
	Mean Difference	-11.88320	-11.88320
	Std. Error Difference	.36773	.36773
	95% Confidence Interval of the Difference	Lower -12.73118	-12.89971
		Upper -11.03522	-10.86669

Berdasarkan tabel diatas diperoleh nilai sig.F sebesar 0.044 lebih kecil dari 0.05 hal ini berarti bahwa ke dua varian (*number connection* 100 dan *number connection* 200) adalah berbeda sehingga pengujian menggunakan *Equal Variances Not Assumed*. Berdasarkan nilai Signifikan sebesar 0.000 yang berarti bahwa ada perbedaan antar *test durations* untuk *number connection* 100 dan 200 hal ini membuktikan bahwa jika beban *traffic* tinggi maka akan sangat mempengaruhi kinerja web server tersebut ketika diakses.

Pengujian diatas untuk nilai *number connection* 100 menggunakan :
`httpperf --server=elearningnetworking.net --rate=10 --num-conns=100 --timeout 60 > hasil-100`. Sedangkan untuk nilai *number connection* 200 menggunakan :
`httpperf --server=elearningnetworking.net --rate=10 --num-conns=200 --timeout 60 > hasil-200`.

Hasil Resource Utilization

Berdasarkan hasil uji sebelum dan sesudah konfigurasi di diperoleh data sebagai berikut :

Tabel 11 Hasil Resouce Utilization

Sebelum	Sudah
top - 20:00:24 up 2 min, 4 users, load average: 2.89, 1.49, 0.58	top - 20:00:24 up 2 min, 4 users, load average: 2.89, 1.49, 0.58
Tasks: 189 total, 1 running, 188 sleeping, 0 stopped, 0 zombie	Tasks: 189 total, 1 running, 188 sleeping, 0 stopped, 0 zombie
Cpu(s): 7%us, 3.1%sy, 0.0%ni, 74.0%id, 18.0%wa, 0.0%hi, 0.0%si, 0.0%st	Cpu(s): 18%us, 3.1%sy, 0.0%ni, 74.0%id, 18.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 3006736k total, 940716k used, 2066020k free, 99304k buffers	Mem: 3006736k total, 940716k used, 2066020k free, 99304k buffers
Swap: 2096476k total, 0k used, 2096476k free, 477880k cached	Swap: 2096476k total, 0k used, 2096476k free, 477880k cached

Berdasarkan tabel diatas terlihat perbedaan sebelum dan sesudah konfigurasi dimana terlihat sebelum konfigurasi proses cpu mengalami kenaikan menjadi 7%, sedangkan sesudah konfigurasi proses cpu mengalami lonjakan menjadi 18%. Hal ini berarti kinerja web server yang baik adalah setelah dikonfigurasi.

3. KESIMPULAN

Berdasarkan dari hasil pengujian yang dilakukan dalam penelitian ini, dapat disimpulkan sebagai berikut :

1. Terdapat perbedaan sebelum dan setelah konfigurasi
2. Pada uji *Independent Sample Test*, dijelaskan berdasarkan nilai signifikan sebesar 0.000 yang berarti bahwa ada perbedaan antar *test durations* untuk *number connection* 100 dan 200 hal ini membuktikan bahwa jika beban *traffic* tinggi maka akan sangat mempengaruhi kinerja *web server* tersebut ketika diakses..
3. Pada Kriteria Kinerja yaitu *Response time*, *Throughput*, dan *Resource Utilization*, mengalami peningkatan setelah konfigurasi.

SARAN

Berdasarkan hasil penelitian selama ini yang penulis lakukan, terdapat beberapa hal yang harus ditambahkan guna pengembangan lebih lanjut, sebagai berikut :

1. Untuk perangkat keras perlu tingkatan kemampuannya seperti *memory* dan kapasitas penyimpanan.
2. Untuk teknologi jaringan perlu ditingkatkan dengan kecepatan *gigabit*.
3. Perlu nya sumber daya manusia yang melakukan *maintenance* berkala.

DAFTAR PUSTAKA

- [1] Purbo, Onno W. dan Antonius AH. 2002. Teknologi e-Learning Berbasis PHP dan MySQL: Merencanakan dan Mengimplementasikan Sistem e-Learning. Jakarta: Gramedia.
- [2] Desy Lukitasari, Ahmad Fali Oklilas., 2010, Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web server Cluster Menggunakan Linux Virtual Server, Jurnal Generic, Vol.5, No.2.
- [3] Iwan Sofana (2007 , Juli) , Mudah Membangun Server dengan Fedora Core, Informatika, Yogyakarta.
- [4] Martin A, Carey W, Lance T. 2003. Performance Benchmarking of Dynamic Web Technologies, Department of Computer Science University of Calgary, ACM SIGMETRICS Performance Evaluation, Vol.31, No.3.
- [5] Mohammad Obaidat. (2010). Fundamental of Performance Evaluation of Computer and Telecommunication System. New Jersey, John Wiley.
- [6] Yiming Hu. 1999. Measurement, Analysis and Performance Improvement of The Apache Web Server. 18th IEEE International Performance, Computing, and Communications Conference, page 261-267 February 10-12, Phoenix/Scottsdale, Arizona.
- [7] Paul B, Mark C. 1999. A Performance Evaluation of Hyper Text Transfer Protocols. Proceedings of the 1999 ACM SIGMETRICS Vol. 27, No.1. Page 188-197.
- [8] Meier, J. D., et al. 2010. Performance Testing Guidance for Web Applications, Wiley-Interscience New York, NY, USA.

- [9] Rio Rasian, Petrus Mursanto, 2009, Perbandingan Kinerja Pendekatan Virtualisasi, Jurnal Sistem Informasi, Vol. 5, No. 2.
- [10] Jefry Alvonsius Rabur, Joko Purwadi, Willy S. Raharjo. 2012, Informatika, Vol.8 No.2.
- [11] Arikunto, Suharsimi (2002). Prosedur Penelitian, Suatu Pendekatan Praktek. Jakarta: PT.Rineka Cipta.
- [12] file-primer.sh , <https://launchpad.net/mysql-tuning-primer>. (Diakses tanggal 9 Februari 2016.)