

ANALISIS PERBANDINGAN KINERJA WEB SERVER

NGINX DENGAN LITESPEED

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik Pada Fakultas Teknik
Universitas Islam Riau



OLEH:

RONALDI PUTRA

143510211

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS ISLAM RIAU

PEKANBARU

2019

LEMBAR PENGESAHAN PEMBIMBING SKRIPSI

Nama : Ronaldi Putra
NPM : 143510211
Jurusan : Teknik
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul Skripsi : Analisa Perbandingan Kinerja Web Server Nginx Dengan Litespeed

Format sistematika dan pembahasan materi pada masing-masing bab dan sub bab dalam skripsi ini telah dipelajari dan dinilai relatif telah memenuhi ketentuan-ketentuan dan kriteria - kriteria dalam metode penulisan ilmiah. Oleh karena itu, skripsi ini dinilai layak dapat disetujui untuk disidangkan dalam ujian komprehensif.

Pekanbaru, 13 Desember 2019

Disetujui Oleh

Dosen Pembimbing I

Dosen Pembimbing II

YUDHIARTA, ST., M.Kom

APRI SISWANTO, S.Kom., M.Kom

Disahkan Oleh :

Dekan Fakultas Teknik

Ketua Prodi Teknik Informatika



KUS ZAINI, MT., MS., TR

081 733 03 02 098

amretorosi

AUSE LABELAPANSA, ST., M.Cs., M.Kom

**LEMBAR PENGESAHAN
TIM PENGUJI UJIAN SKRIPSI**

Nama : Ronaldi Putra
NPM : 143510211
Jurusan : Teknik
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul Skripsi : Analisis Perbandingan Kinerja Web Server Nginx Dengan Litespeed

Skripsi ini secara keseluruhan dinilai telah memenuhi ketentuan-ketentuan dan kaidah-kaidah dalam penulisan penelitian ilmiah serta telah diuji dan dapat dipertahankan dihadapan Tim Penguji. Oleh karena itu, Tim Penguji Ujian Skripsi Fakultas Teknik Universitas Islam Riau menyatakan bahwa mahasiswa yang bersangkutan dinyatakan **Telah Lulus Mengikuti Ujian Komprehensif Pada Tanggal 13 Desember 2019** dan disetujui serta diterima untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Strata Satu Bidang Ilmu Teknik Informatika.

Pekanbaru, 12 Desember 2019

Tim Penguji

1. Dr. Evizal, M.Eng. Sebagai Tim Penguji I
2. Abdul Syukur, S.Kom., M.Kom. Sebagai Tim Penguji II



Dosen Pembimbing I

Dosen Pembimbing II

YUDHI ARTA, ST., M.Kom

APRI SISWANTO, S.Kom., M.Kom

Disahkan Oleh :

Dekan Fakultas Teknik

Ketua Prodi Teknik Informatika
Am Selapoti



Dr. H. ABU KUDUS ZAINI, MT, MS., TR
NPK : 88 03 02 098

AUSE LABELLAPANSA, ST., M.Cs., M.Kom

LEMBAR PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan dibawah ini :

Nama : Ronaldi Putra

Tempat/Tgl Lahir : Dumai, 23 April 1996

Alamat : Jalan Cemara Perum Puri Cemara

Adalah mahasiswa Universitas Islam Riau yang terdaftar pada:

Fakultas : Teknik

Program Studi : Teknik Informatika

Jenjang Pendidikan : Strata-1 (S1)

Dengan ini menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis adalah benar dan asli hasil dari penelitian yang telah saya lakukan dengan judul "**Analisis Perbandingan Kinerja Web Server Nginx Dengan Litespeed**".

Apabila dikemudian hari ada yang merasa dirugikan atau menuntut karena penelitian ini menggunakan sebagian hasil tulisan atau karya orang lain tanpa mencantumkan nama penulis yang bersangkutan, atau terbukti karya ilmiah ini bukan karya saya sendiri atau plagiat hasil karya orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan perundangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya untuk dapat digunakan sebagaimana semestinya.

Pekanbaru, 16 Desember 2019

Yang membuat pernyataan,



Ronaldi Putra
143510211

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Dengan segala kerendahan hati Penulis haturkan rasa syukur dalam kehadiran Allah SWT, yang telah memberikan rahmat dan karunia-Nya yang berupa kemampuan, kesehatan dan juga kesempatan kepada Penulis untuk menyelesaikan laporan tugas akhir “ANALISIS PERBANDINGAN KINERJA WEB SERVER NGINX DENGAN LITESPEED ” ini.

Penulisan skripsi ini tidak lepas dari bantuan dan dukungan serta bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan sebesar-besarnya kepada :

1. Kedua orang tua Penulis yang telah memberikan motivasi untuk menyelesaikan proposal tugas akhir ini.
2. Saudara kandung Penulis, Eka Yanfitri, Desi Fitria, Roni Putra, dan Irda ramadhani. yang telah memberikan motivasi untuk menyelesaikan proposal tugas akhir ini
3. Dosen dan karyawan fakultas teknik Universitas Islam Riau
4. Bapak Hendra Gunawan, ST, M.Eng selaku penasehat akademik Penulis.
5. Teman-teman Penulis di TI UIR kelas D, khususnya Agung, Bobby, Kurniawan, Luthfy, Ery, Brama, Rizal, Maya boy, Ayu butet, Ima mak ee, Yani asep, dan teman teman Penulis lainnya yang tidak dapat penulis sebutkan satu persatu.

6. Dosen pembimbing bapak Yudhi Arta, ST.,M.Kom dan bapak Apri Siswanto, S.Kom.,M.Kom.
7. Jajaran asisten dosen mata kuliah sistem operasi dan konsentrasi jaringan yaitu Abdul aziz, Andi cahyono, Hermanto, dan Hanif.
8. Dan berbagai pihak yang tidak dapat Penulis sebutkan satu persatu
Penulis sangat menyadari bahwa masih terdapat banyak kekurangan didalam penulisan laporan ini. Untuk itu Penulis mengharapkan kritik dan saran yang membangun guna kebaikan dan kesempurnaan laporan tugas akhir ini. Penulis berharap skripsi ini bisa bermanfaat bagi pembaca nantinya.

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Pekanbaru, Desember 2019

Penulis

Ronaldi Putra
143510211

DAFTAR ISI

	Hal
KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	ix
BAB I PENDAHULUAN	
1.1.Latar belakang Masalah	1
1.2.Identifikasi Masalah	4
1.3.Rumusan Masalah	4
1.4.Batasan Masalah	5
1.5.Tujuan Dan Manfaat Penelitian.....	5
BAB II LANDASAN TEORI	
2.1.Studi Kepustakaan.....	7
2.2.Dasar teori	9
2.2.1. Web Server.....	9
2.2.2. Httpperf	9
2.2.3. Siege.....	10
2.2.4. Nikto.....	10
2.2.5. Wireshark	11
2.2.6. Nginx	11
2.2.7. Litespeed	12

2.2.8. Pengujian Kinerja	13
--------------------------------	----

BAB III METODOLOGI PENELITIAN

3.1 Definisi Masalah dan Analisis	15
3.2 Alat dan Bahan Penelitian Yang Digunakan.....	16
3.2.1. Perangkat Keras (<i>Hardware</i>)	16
3.2.2. Perangkat Lunak (<i>Software</i>)	16
3.3 Prosedur Penelitian.....	17
3.4 Pengembangan dan Perancangan Sistem	18
3.5 Skenario Pengujian.....	19
3.5.1 Dokumen Pengujian	24
3.6 Parameter Kinerja.....	24

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian <i>Web Server</i> Nginx dan Litespeed.....	27
4.1.1 Pengujian Terhadap <i>Web Statis</i>	27
4.1.2 Pengujian Terhadap File Gambar.....	34
4.1.3 Pengujian Terhadap File Php	41
4.1.4 Pengujian Terhadap Halaman <i>Web</i> Bukalapak	48
4.1.5 Pengujian Terhadap Halaman <i>Web</i> Wordpress.....	55
4.2 Pengujian <i>Log Web Server</i> Nginx dan Litespeed.....	62
4.3 Pengujian <i>Security Web Server</i> Nginx dan Litespeed.....	63

BAB V KESIMPILAN DAN SARAN

5.1 Kesimpulan.....67

5.2 Saran.....68

DAFTAR PUSTAKA 69



Dokumen ini adalah Arsip Miik :
Perpustakaan Universitas Islam Riau

DAFTAR GAMBAR

	Hal
Gambar 3.1. Skema Prosedur Penelitian.....	17
Gambar 3.2. Topologi Jaringan	18
Gambar 3.3. Perancangan Jaringan	19
Gambar 3.4 Skenario Pengujian.....	20
Gambar 3.5 Desain Pengujian.....	21
Gambar 3.6 Desain Pengujian Keamanan.....	23
Gambar 4.1 Tampilan Halaman Testing	28
Gambar 4.2 Hasil <i>Throughput</i> Halaman Testing	28
Gambar 4.3 Hasil <i>connection</i> halaman testing.....	29
Gambar 4.4 hasil <i>reply time</i> halaman testing.....	30
Gambar 4.5 hasil <i>request</i> halaman testing.....	30
Gambar 4.6 Hasil <i>error</i> halaman testing Nginx.....	31
Gambar 4.7 Hasil <i>error</i> halaman testing Litespeed	31
Gambar 4.8 Tampilan gambar bisa.jpg.....	34
Gambar 4.9 Hasil <i>throughput</i> bisa.jpg.....	35
Gambar 4.10 Hasil <i>connection</i> bisa.jpg	36
Gambar 4.11 Hasil <i>reply time</i> bisa.jpg.....	36
Gambar 4.12 Hasil <i>request time</i> bisa.jpg	37
Gambar 4.13 Hasil <i>error</i> bisa.jpg pada Nginx	38
Gambar 4.14 Hasil <i>error</i> bisa.jpg pada Litespeed	38
Gambar 4.15 Tampilan halaman info.php.....	42

Gambar 4.16 Hasil <i>throughput</i> info.php	42
Gambar 4.17 Hasil <i>connection</i> info.php	43
Gambar 4.18 Hasil <i>reply time</i> info.php	44
Gambar 4.19 Hasil <i>request time</i> info.php	44
Gambar 4.20 Hasil <i>error</i> info.php pada Nginx	45
Gambar 4.21 Hasil <i>error</i> info.php pada Litespeed	45
Gambar 4.22 Tampilan halaman web Bukalapak	48
Gambar 4.23 Hasil <i>throughput</i> bukalapak.html	49
Gambar 4.24 Hasil <i>connection time</i> bukalapak.html	50
Gambar 4.25 Hasil <i>reply time</i> bukalapak.html	51
Gambar 4.26 Hasil <i>request time</i> bukalapak.html	51
Gambar 4.27 Hasil <i>error</i> bukalapak.html Nginx	52
Gambar 4.28 Hasil <i>error</i> bukalapak.html Litespeed	52
Gambar 4.29 Tampilan halaman web Wordpress	55
Gambar 4.30 Hasil <i>throughput</i> wordpress	56
Gambar 4.31 Hasil <i>connection time</i> wordpress	57
Gambar 4.32 Hasil <i>reply time</i> wordpress	57
Gambar 4.33 Hasil <i>request time</i> wordpress	58
Gambar 4.34 Hasil <i>error</i> wordpress pada Nginx	59
Gambar 4.35 Hasil <i>error</i> wordpress pada Litespeed	59
Gambar 4.36 Hasil <i>log</i> web server Litespeed	62
Gambar 4.37 Hasil <i>log</i> web server Nginx	63
Gambar 4.38 Hasil <i>scanning</i> web server Nginx	64

Gambar 4.39 Hasil *scanning* web server Litespeed.....64

Gambar 4.40 Hasil *capture packet* Nginx.....65

Gambar 4.41 Hasil *capture packet* Litespeed66



DAFTAR TABEL

	Hal
Tabel 1.1 Perbandingan Pangsa Pasar Web Server.....	3
Tabel 4.1 Hasil Pengujian Halaman Teseting.....	32
Tabel 4.2 Hasil Pengujian <i>Throughput</i> Halaman Testing.....	33
Tabel 4.3 Hasil pengujian bisa.jpg.....	39
Tabel 4.4 Hasil pengujian <i>throughput</i> bisa.jpg.....	40
Tabel 4.5 Hasil pengujian file PHP.....	46
Tabel 4.6 Hasil pengujian <i>throughput</i> file PHP.....	47
Tabel 4.7 Hasil pengujian Bukalapak.....	53
Tabel 4.8 Hasil <i>throughput</i> bukalapak.....	54
Tabel 4.9 Hasil pengujian Wordpress.....	60
Tabel 4.10 Hasil <i>throughput</i> wordpress.....	61

BAB I

PENDAHULUAN

1.1. Latar Belakang

World wide web diciptakan pada tahun 1989 oleh Tim Berners – lee seorang ilmuwan di CERN (Organisasi Eropa untuk Riset Nuklir), sejak awal penemuannya *world wide web* hanya digunakan untuk memenuhi permintaan berbagi informasi sesama ilmuwan di CERN. Ternyata penemuan ini memberikan sebuah konsep baru untuk pertukaran informasi dan dapat diterapkan diseluruh dunia. Berdasarkan data dari *netcraft* dari tahun 2011 ke tahun 2013 lebih dari 350 juta situs baru dibuat (Wikipedia).

Pertumbuhan ini dikarenakan banyaknya permintaan akan informasi yang cepat dan dapat diterapkan diberbagai bidang. Saat ini, hampir seluruh perusahaan, lembaga pendidikan, dan institusi pemerintah memiliki sebuah *website* sebagai sarana layanan terhadap masyarakat luas. Pemerosesan data dalam sebuah *website* ditangani oleh bahasa program. Ada dua jenis bahasa program yang digunakan dalam membentuk sebuah halaman web, yaitu *server side scripting* dan *client side scripting*. *Server side scripting* adalah bahasa program yang tidak terlihat oleh pengguna, yang dijalankan oleh *server* web. Contoh dari *server side scripting* ini adalah PHP dan ASP. Sedangkan *client side scripting* adalah jenis bahasa program yang kode sumbernya dapat dilihat oleh pengguna. Contohnya adalah HTML, CSS, dan *JavaScript*. Dengan adanya bahasa program yang menangani sebuah halaman web, konten dari halaman web itu sendiri semakin bervariasi. Pemanfaatannya pun tidak hanya sekedar

menampilkan profil usaha atau halaman statis saja, namun dapat berupa sistem, forum, media komunikasi, dan media informasi. *Website* ini dijalankan oleh satu atau lebih *server*, dimana *server* membutuhkan suatu sistem informasi yang kuat dan handal dalam berbagai hal. Program untuk membangun *web server* sekarang ini sangat banyak di antaranya *apache*, *nginx*, dan *litespeed*. Masing – masing program tersebut mempunyai karakter masing – masing. Misalkan *nginx* yang terkenal dengan kecepatan menyajikan data, stabil, kekayaan akan fitur, mudah dikonfigurasi dan lain-lain sedangkan *litespeed* yang lebih cepat mengolah PHP dan kemanannya lebih baik dari *web server* lainnya, pembatasan validasi HTTP *request*, mampu melebihi *apache* hingga 6 kali lebih cepat.

Perkembangan *social-networking* seperti facebook, twitter dan instagram sangat pesat. Berdasarkan data dari engadget.com saat ini setidaknya ada 143.199 kicuan per detik yang dikirim oleh penggunanya diseluruh dunia. Tidak ketinggalan juga dengan situs forum di Indonesia, seperti Kaskus dan Indowebster memiliki banyak pengunjung setiap harinya. Maka yang dibutuhkan saat ini adalah sebuah *web server* yang mampu melayani banyak permintaan tapi dengan *resource* yang sedikit. Saat ini *litespeed* dan *nginx* sedang meningkat popularitasnya, hal ini disebabkan oleh teknologi yang di terapkan di kedua *web server* tersebut lebih baik dibandingkan dengan yang lain, berdasarkan data dari *netcraft* pangsa pasar dari jutaan situs tersibuk *nginx* pada bulan mei 2018 sebesar 242,826 dan pada bulan juni 2018 sebesar 246,812 atau naik sebesar 0.40%. *Litespeed* sendiri mengalami peningkatan juga, pada bulan mei 2018 sebesar 14,777 dan pada bulan juni 2018 sebesar 15,019. Sedangkan *apache* sendiri

mengalami penurunan, pada bulan mei 2018 sebesar 360,234 dan pada bulan juni 2018 sebesar 355,382(*Netcraft*).

Tabel 1.1 Perbandingan pangsa pasar *web server* (sumber : *netcraft*)

Developer	May 2018	Percent	June 2018	Percent	Change
Apache	360,234	36.02%	355,382	35.54%	-0.49
nginx	242,826	24.28%	246,812	24.68%	0.40
Microsoft	92,776	9.28%	93,092	9.31%	0.03
LiteSpeed	14,777	1.48%	15,019	1.50%	0.02

Dari data yang telah *netcraft* tersebut bisa dilihat bahwa yang *nginx* dan *litespeed* mengalami peningkatan, sedangkan *apache* mengalami penurunan, khusus untuk *litespeed* walau pun masih baru akan tetapi perlahan tapi pasti terus meningkat penggunaannya sehingga diprediksi akan menggantikan *apache*. Pada saat ini di butuhkan *web server* yang mampu menangani banyak masalah yang umum terjadi seperti beban permintaan, kecepatan transfer data, koneksi, dan lain-lain. Jika tidak di tanggapi dengan serius maka akan berdampak pada layanan sebuah *server* , pemilik *website* tentu tidak ingin terkena dampak tersebut sehingga para penyedia layanan *server* harus tau *web server* mana yang harus digunakan sesuai dengan kebutuhan suatu *website*.

Dengan adanya perbandingan kinerja *Web Server* diatas maka perlu adanya kajian lebih lanjut dengan dilakukannya pengujian terhadap kinerja dari masing-masing *Web Server* Litespeed dan Nginx.

1.2. Identifikasi Masalah

Berdasarkan permasalahan yang telah diuraikan diatas, maka rumusan masalah dalam pembuatan sistem ini adalah sebagai berikut:

1. Pengelola layanan *server* butuh referensi untuk memilih layanan server yang sesuai dengan kebutuhan.
2. Penyedia layanan ingin *web server* yang tahan dengan beban permintaan yang besar.
3. Membandingkan sistem keamanan antara *Web Server* Nginx dan Litespeed.
4. Membandingkan kinerja antara *Web Server* Nginx dan Litespeed.
5. Memberikan manfaat dan solusi dari perbandingan *Web Server* Nginx dan Litespeed.

1.3. Rumusan Masalah

Penelitian ini akan membatasi cakupan permasalahan terkait pada monitoring jaringan lokal. Batasan masalah yang dimaksud adalah:

1. Bagaimana tahapan merancang *web server* yang baik dan benar?.
2. Bagaimana melakukan *peforma testing* pada *Web Server* Nginx dan Litespeed?
3. Bagaimana menguji *Web Server* Nginx dan Litespeed terhadap beban permintaan yang banyak?
4. Bagaimana menguji sistem keamanan *Web Server* Nginx dan Litespeed terhadap ancaman peretasan atau pun pembobolan?

Selain dari pembahasan batasan masalah diatas tidak akan dibahas pada penelitian ini.

1.4. Batasan Masalah

Agar penelitian ini lebih terarah dan tidak menimbulkan perluasan pada pembahasannya nanti, maka diberi batasan ruang lingkup pembahasan yang dibahas. Batasan masalah yang dimaksud adalah:

1. Aplikasi yang digunakan adalah HTTPERF dan SIEGE.
2. Parameter yang diukur dalam pengujian ini adalah *Throughput*, *Connection*, *Request*, *Reply*, *Error*, *Log*, dan *Web Security*.
3. Pengujian *log* dilakukan dengan melihat *file access,log* pada kedua *web server*.
4. *Web Server* yang akan digunakan adalah *Litespeed* dan *Nginx*.
5. Pengujian *web security* menggunakan aplikasi Nikto dan Wireshark.

Selain dari pembahasan batasan masalah diatas tidak akan dibahas pada penelitian ini.

1.5. Tujuan dan Manfaat Penelitian

Secara umum tujuan dan manfaat dilakukannya penelitian ini adalah sebagai berikut:

1. Tujuan penelitian

Berikut ini adalah tujuan dilakukannya penelitian ini:

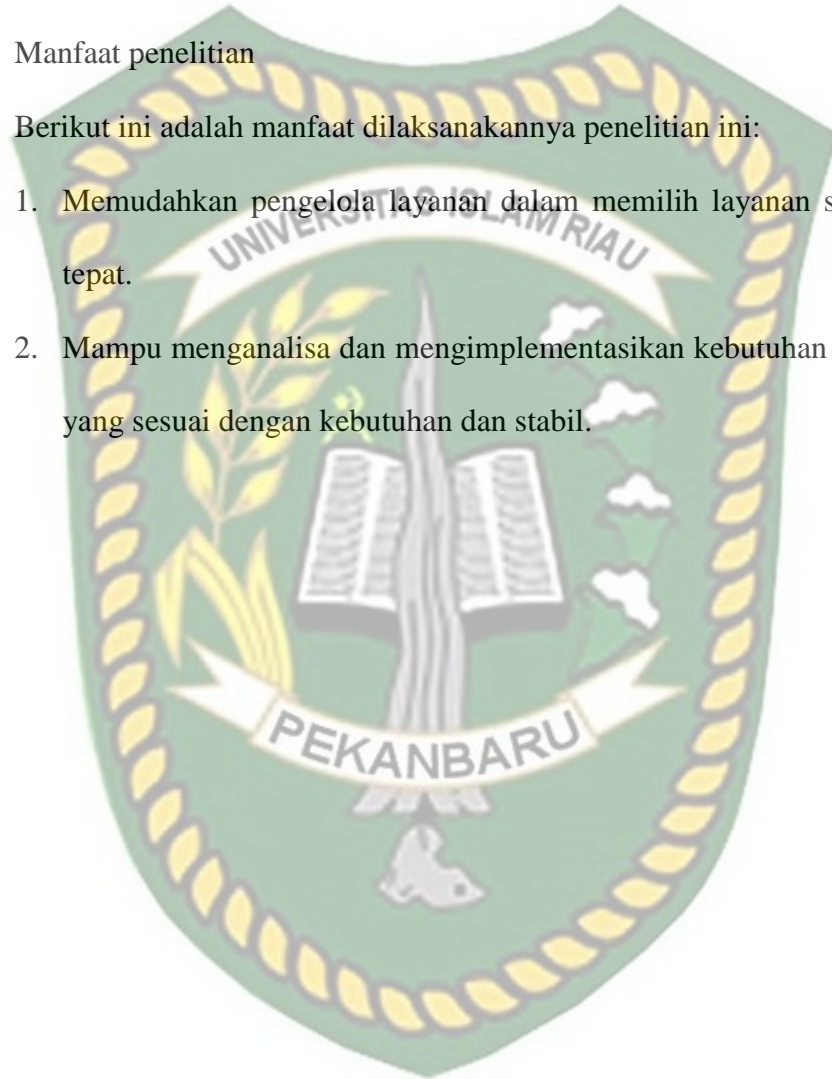
1. Membandingkan kinerja *Web Server* Nginx dan Litespeed.

2. Memberikan informasi kepada pembaca tentang *Web Server* yang sesuai dengan kebutuhan.

2. Manfaat penelitian

Berikut ini adalah manfaat dilaksanakannya penelitian ini:

1. Memudahkan pengelola layanan dalam memilih layanan server yang tepat.
2. Mampu menganalisa dan mengimplementasikan kebutuhan *web server* yang sesuai dengan kebutuhan dan stabil.



BAB II

LANDASAN TEORI

2.1. Studi Kepustakaan

Untuk menyusun proposal penelitian ini penulis juga melakukan studi kepustakaan yang merujuk kepada penelitian-penelitian sebelumnya yang berkaitan dengan penelitian yang penulis buat. Studi kepustakaan ini dilakukan sebagai bahan perbandingan dan referensi bagi penulis.

Rahmad Dawod, Said Fairuz Qiana, dan Sayed Muchalil (2014) melakukan penelitian tentang kelayakan Raspberry Pi sebagai *web server* yang membandingkan kinerja antara Nginx, Apache, dan Lighttpd pada platform Raspberry Pi. Penelitian tersebut membandingkan beberapa parameter seperti: *reply time* dan *maximum request* menggunakan software Raspberry Pi. Penelitian tersebut juga mengambil kesimpulan bahwa dari ketiga *Web Server* yang dibandingkan bahwa Nginx lah yang dapat menangani 100 permintaan secara simultan dan memiliki kurun waktu yang lebih cepat

Intan Feriza dkk (2017) juga melakukan penelitian tentang perbandingan *web server* Apache dan Nginx. Pada penelitian tersebut membandingkan kinerja antara dua *web server* secara virtual yang menguji *throughput*, *reply*, dan *errorr*. Pada penelitian tersebut menarik kesimpulan bahwa perbandingan *troughput* dan *reply* Nginx lah yang lebih baik dari Apache, pada pengujian *errorr* Nginx dan Apache memiliki kinerja yang sama baiknya.

Intan Yuli Andhica, Dadan Irwan (2017) melakukan penelitian tentang peforma kinerja *web server* berbasis Ubuntu Linux dan Turnkey Linux. Pada

penelitian tersebut menguji kinerja *web server* berdasarkan kemampuannya dari *request rate*, *reply time* dan *throughput* dengan menggunakan aplikasi Httpperf untuk melihat *reply time* dan *throughput* masing-masing *web server*.

Fariq Adnan (2016) melakukan penelitian tentang analisis perbandingan performa *web server apache* dan *nginx* menggunakan Httpperf pada vps dengan sistem operasi centos. Pada penelitian tersebut Fariq Adnan (2016) menggunakan dua parameter sebagai acuan dari pengujian yang sudah tersedia pada aplikais *benchmark httpperf* yaitu *reply time* dan *throughput*. Pengujian ini dilakukan dengan menggunakan centos sebagai system operasinya.

Novaldo Aditya Putra, Indrastanti Ratna Widiyari, M.T (2016) juga melakukan penelitian tentang perbandingan kinerja *web server* Nginx dengan Tomcat pada *platform* Raspberry Pi. Pada penelitian ini menggunakan aplikasi JMeter untuk membangkitkan sejumlah *request simultan* ke aplikasi *web server* untuk mensimulasikan sejumlah *user* yang secara bersamaan melakukan *request* kepada aplikasi *web server* melalui *web browser* mereka. Untuk menguji kelayakan suatu aplikasi *web server* maka Raspberry Pi, secara bergantian diinstalasi dengan setiap jenis aplikasi *web server* yang diuji, yaitu Nginx dan Tomcat. Setiap instalasi aplikasi *web server* memakai konfigurasi standar bawaanya. Raspberry Pi kemudian dimasukkan ke dalam topologi dan JMeter dijalankan pada *client* untuk mengirimkan sejumlah *request* ke aplikasi web server. Aplikasi *web server* dikatakan layak untuk dijalankan pada Raspberry Pi jika bisa menangani *minimum request* tanpa kesalahan (*error*) dalam *maximum reply time* tertentu.

2.2. Dasar Teori

Berikut ini adalah beberapa dasar teori yang berkenaan dalam penelitian tugas akhir ini:

2.2.1. Web Server

Web server adalah sebuah bentuk dari *server* yang khusus digunakan untuk menyimpan halaman *website* atau *home page*. Sebuah komputer dapat dikatakan *web server* apabila komputer tersebut memiliki suatu program *server* yang disebut PWS atau *Personal Web Service*. PWS ini kemudian nantinya difungsikan agar halaman web yang ada di dalam sebuah komputer *server* dapat dipanggil oleh komputer klien (Nugroho, 2004).

Fungsi utama *web sever* adalah sebagai alat untuk memproses berbagai berkas yang diminta oleh klien, lalu memberikan respon dalam bentuk halaman web. Halaman *website* tersebut terdiri dari teks, gambar, dokumen, video, dan lain-lain.

2.2.2. HTTPERF

Httpperf adalah program untuk mengukur kinerja atau performansi dari *web server* yang dibuat oleh David Mosberger dari HP Labs. *Httpperf* menyediakan fitur yang fleksibel dalam pembuatan beban kerja sesuai dengan variabel yang diberikan padanya (Mosberger, 1998:1).

Httpperf adalah sebuah tool untuk mendapatkan performansi dari sebuah *web server*. *Httpperf* sangat fleksibel untuk menghasilkan berbagai macam HTTP request untuk mendapatkan performansi dari sebuah *web server*. Fokus dari

httperf adalah tidak mengimplementasikan sebuah *benchmark* saja, tetapi juga memberikan *tool* yang *robust* dan memiliki performansi tinggi untuk memfasilitasi baik di level mikro atau makro dalam sebuah *benchmark*. *Tool* *httperf* juga telah support *http/1.1* dan protokol SSL.

2.2.3. SIEGE

Siege merupakan alat penguji beban *http* dan alat untuk *benchmarking*. Aplikasi ini dirancang untuk memungkinkan pengembang *web* mengukur *web server* dalam tekanan. Siege mendukung dasar otentikasi, *cookies*, protocol HTTP dan HTTPS. Penggunaanya dapat menguji *web server* dengan sejumlah simulasi *web browser* yang dapat dikonfigurasi (Fulmer,2012).

Fungsi utama siege digunakan untuk memberikan tekanan kepada *web server* dengan cara membangun simulasi sejumlah pengguna internet yang secara simultan mengakses sebuah atau banyak halaman dalam rentang waktu tertentu. Selanjutnya program ini melaporkan jumlah klik yang direkam, besar data yang ditransfer, berapa lama *web server* memberikan respon, konkurensi, dan status balasan *web server*. Semua fitur-fitur pengujian dapat diatur dengan perintah melalui *command line*.

2.2.4. NIKTO

Nikto merupakan suatu alat gratis (*opensource*) yang digunakan untuk pemindaian suatu *web* aplikasi dengan menggunakan CLI(*command line*) karena nikto dikembangkan menggunakan bahasa pemrograman perl. Meskipun tidak ada

user interface pada nikto tetapi didalamnya terdapat 6700 potensi kerentanan file/program untuk pengujian. Dengan fitur *auto scanning* yang dilakukan nikto untuk mencari kerentanan secara otomatis (Jajang, J., 2018).

2.2.5. WIRESHARK

Wireshark adalah *tool* yang ditujukan untuk penganalisaan paket data jaringan. Wireshark disebut juga *network packet analyzer* yang berfungsi menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut serinci mungkin (Kurniawan, 2012).

2.2.6. NGINX

Nginx adalah *web server* yang ringan digunakan dan memiliki beberapa modul untuk digunakan bekerja sebagai *reserve proxy* dan *mail server* tanpa menginstal dependensi lainnya. Nginx dirancang khusus untuk menggunakan sedikit *resource* (sumber daya) namun memberikan konten pada kecepatan yang lebih tinggi (Kula, P.J., 2014).

Nginx menjadi pilihan berkat kinerjanya yang tinggi, stabilitas, kekayaan akan fitur, mudah dikonfigurasi dan terutama hemat sumber daya. Berbeda dengan *server* tradisional, Nginx tidak sepenuhnya bergantung pada '*threads*' untuk menangani permintaan(request). Sebagai pengganti Nginx menggunakan teknik arsitektur yang digerakkan berdasarkan event yang lebih skalabel (*scalable asynchronous event-driven architecture*). Arsitektur ini ternyata pada saat

dibebani hanya memerlukan sedikit memori dan menurut pengembangnya dalam jumlah yang bisa diprediksi.

2.2.7. Litespeed

Litespeed merupakan teknologi *web server* yang dapat mempercepat akses server 50 kali lebih cepat dibandingkan dengan server biasanya. Teknologi Litespeed ini menawarkan performa tinggi, cepat, ringan, stabil, dan handal dalam menangani pekerjaan yang berskala besar.

Litespeed mempunyai beberapa fitur seperti kompatibel dengan *Apache*, lebih cepat, *server* lebih stabil dalam 15 menit, satu-satunya kontrol solusi panel-terintegrasi, dukungan HTTP/2, dan anti-DDoS *advances*. Dengan beberapa fiturnya *Litespeed* sudah jauh lebih unggul dari pada *Apache*. Berikut adalah beberapa detail kelebihan dari *litespeed*:

1. Performa PHP mampu meningkat 50%.
2. Mampu melebihi performa *Apache* hingga 6 kali lebih cepat.
3. Pembatasan validasi HTTP *request*.
4. Anti DDOS.
5. Pencegahan *System Overloading*.
6. *Recover* dari kegagalan secara langsung dan otomatis.
7. Kompatibel dengan *Cpanel*, *Plesk* dan *direct admin*.
8. Dukungan kompatibilitas dengan *mod_security request filtering*.
9. Kompatibel dengan *Apache .htaccess*.

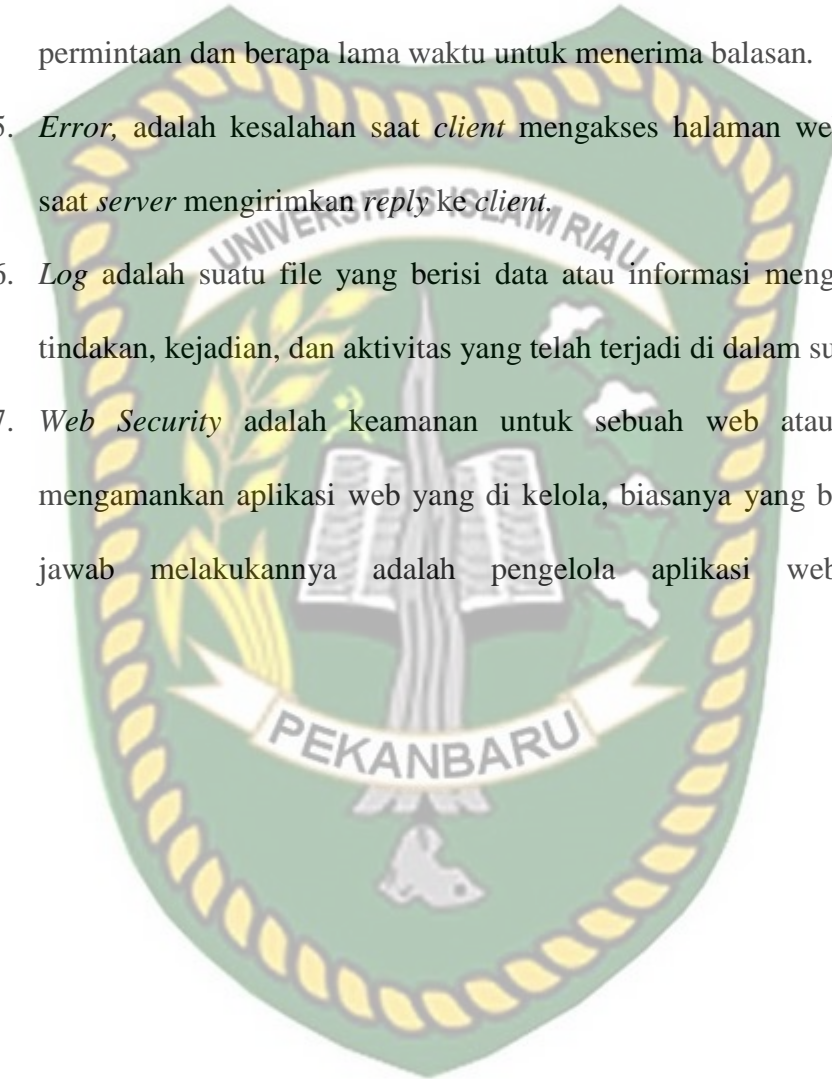
2.2.8. Pengujian Kinerja

Web performance test adalah serangkaian proses pengujian untuk mengukur kinerja perangkat lunak aplikasi *web server* (Guntoro, 2015).

Dalam rekayasa perangkat lunak *performance testing* adalah suatu pengujian yang dilakukan untuk menentukan bagaimana sistem melakukan respon dan stabilitas di bawah beban kerja tertentu (degradasi kinerja), sehingga dengan melakukan pengujian akan dihasilkan data yang dapat dipakai untuk menyelidiki, mengukur, memvalidasi atau memverifikasi dari atribut kualitas sistem tersebut. Dengan *web performance test* dapat dengan mudah membangun sebuah kerangka kerja tes untuk dilakukan secara berulang yang dapat membantu dalam menganalisis kinerja aplikasi *website* dan mengidentifikasi hambatan potensial. Ada beberapa parameter yang dapat diukur dalam sebuah pengujian kinerja *web server* yaitu:

1. *Throughput*, adalah kecepatan transfer sebenarnya antara *server* ke *client* dan *client* ke *server*, misalnya seseorang mendownload dari sebuah *server*, dan kecepatan internetnya 100 kbps, kecepatan *server* 1 mbps maka kecepatan *throughput* sebenarnya adalah 100 kbps. Jika *bandwidth* yang dihasilkan lebih baik maka *server* tersebut akan mampu menampung banyak data.
2. *Connection*, adalah hubungan antara *server* ke *client* dan *client* ke *server*, jika semakin kecil *connection* yang ditunjukkan maka semakin bagus kinerja dari suatu *server*.

3. *Request*, adalah permintaan *client* berupa halaman web atau data kepada *server*.
4. *Reply*, adalah lama waktu yang dibutuhkan *server* untuk merespon permintaan dan berapa lama waktu untuk menerima balasan.
5. *Error*, adalah kesalahan saat *client* mengakses halaman web atau pun saat *server* mengirimkan *reply* ke *client*.
6. *Log* adalah suatu file yang berisi data atau informasi mengenai daftar tindakan, kejadian, dan aktivitas yang telah terjadi di dalam suatu sistem.
7. *Web Security* adalah keamanan untuk sebuah web atau tata cara mengamankan aplikasi web yang di kelola, biasanya yang bertanggung jawab melakukannya adalah pengelola aplikasi web tersebut



BAB III

METODOLOGI PENELITIAN

Metodologi penelitian adalah langkah-langkah atau tahapan perencanaan dengan bantuan beberapa metode, teknik, alat (*tools*) dan dokumentasi dengan tujuan untuk membantu peneliti dalam meminimalkan resiko kegagalan dan menekankan pada proses atau sasaran penelitian (Zainal A. Hasibuan, 2007).

3.1 Definisi Masalah dan Analisis

Pada penelitian ini terdapat beberapa masalah mengenai layanan *web server* yaitu:

1. Penyedia layanan server membutuhkan referensi dalam memilih layanan server yang sesuai dengan kebutuhan.
2. Penyedia layanan ingin *web server* yang tahan dengan beban permintaan yang besar.
3. Keingintahuan terhadap performa dari *web server Nginx* dan *web server Litespeed*.

Dari ketiga permasalahan itu kemudian dilakukan pengujian terhadap kedua *web server*, parameter yang akan di uji adalah *Throughput*, *Connection*, *Request*, *Reply*, *Error*, *Log*, *Web Security*, dari beberapa parameter tersebut akan dihitung oleh aplikasi *httperf* yang hasilnya akan dijadikan kesimpulan dari kinerja kedua *web server* tersebut.

3.2 Alat dan Bahan Penelitian Yang Digunakan

Adapun spesifikasi perangkat keras (*hardware*) yang digunakan untuk melakukan pengujian dan spesifikasi perangkat lunak (*software*) yang dibutuhkan untuk sistem yang akan dibangun adalah sebagai berikut:

3.2.1 Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras (*hardware*) yang dibutuhkan sebagai *server* atau *pc* yang akan digunakan dalam penelitian ini adalah sebagai berikut:

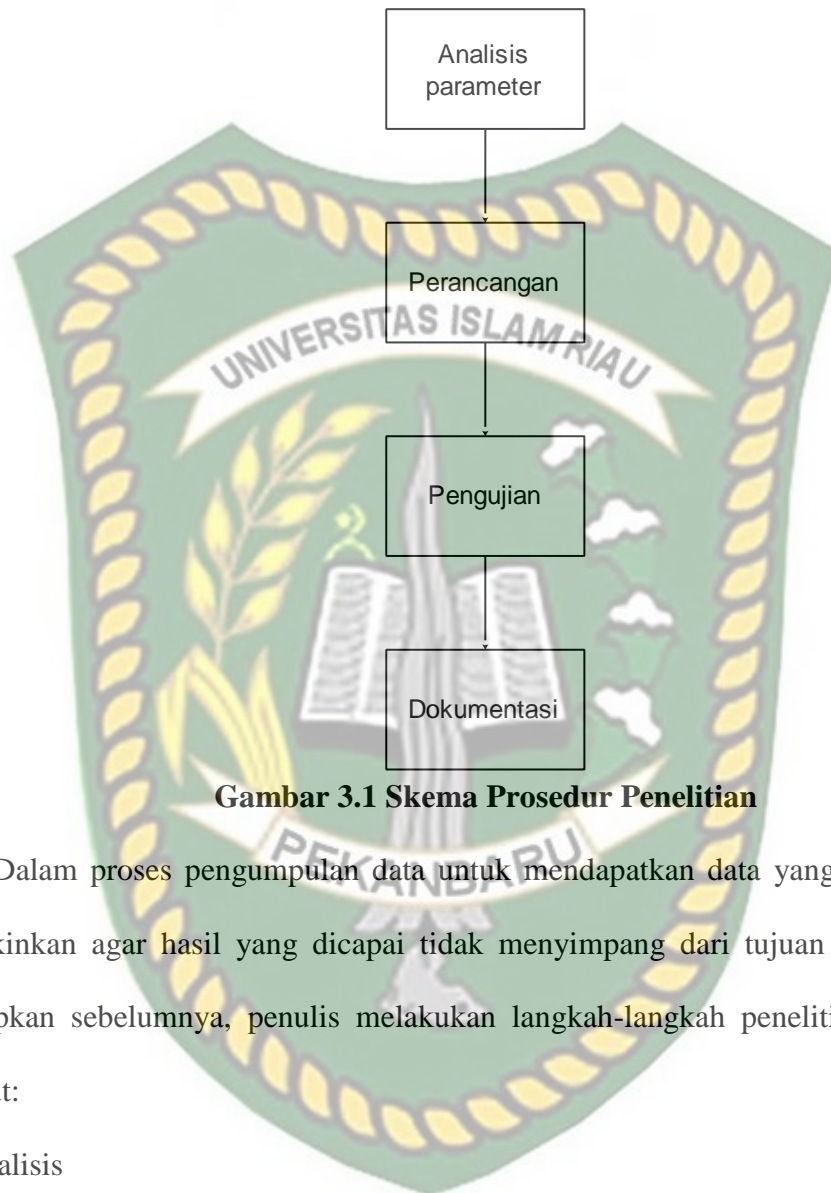
1. *PC/Server*
 - a. *Processor* setara Intel Atom
 - b. RAM 1 GB
 - c. *Hardisk* minimal 320 GB
 - d. *Type System* 32-bit *Operating System*
2. Perangkat Keras
 - a. Kabel *UTP* serial *cross* 1 buah untuk menghubungkan antara *pc client* dan *server*.

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) yang digunakan dalam pembuatan pengujian kinerja ini sebagai berikut:

1. Ubuntu server dan desktop 16.04.
2. *Httpperf* 0.9.0
3. *Siege* 3.0.8
4. *Web Server* NGINX 1.14.1
5. *Web Server* LITESPEED 5.2.8

3.3 Prosedur Penelitian



Gambar 3.1 Skema Prosedur Penelitian

Dalam proses pengumpulan data untuk mendapatkan data yang benar dan meyakinkan agar hasil yang dicapai tidak menyimpang dari tujuan yang telah diterapkan sebelumnya, penulis melakukan langkah-langkah penelitian sebagai berikut:

1. Analisis

Di dalam tahapan ini akan di rancang sebuah proses pengiriman data atau halaman *website* dari *server* menuju *client*, sehingga dapat diketahui parameter-parameter antara lain *throughput*, *request*, *reply*, *connection*, *error*, *log*, dan *web security* sehingga kesimpulan akan di ambil dari hasil perbandingan yang sudah diperoleh.

2. Perancangan

Tahap ini akan merancang analisa spesifikasi kebutuhan yang telah di dapat ke dalam bentuk arsitektural perangkat lunak, untuk diimplementasikan kepada aplikasi yang akan dibuat.

3. Pengujian

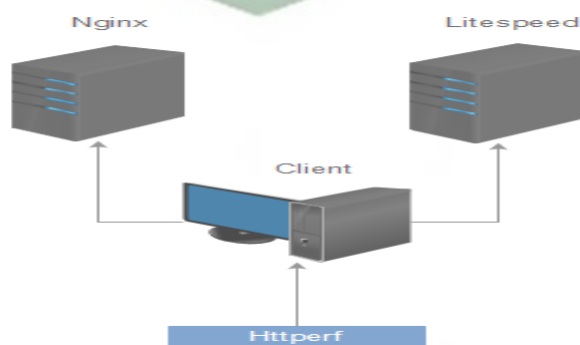
Dalam tahap pengujian kinerja web server NGINX dan LITESPEED dilakukan dengan menggunakan aplikasi untuk mendapatkan hasil dari pengujian yang sedang berjalan.

4. Dokumentasi

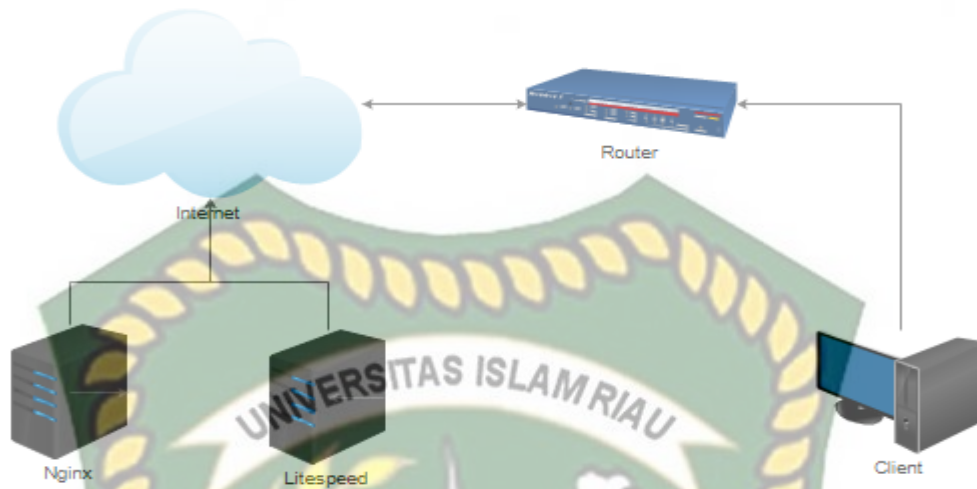
Pada proses dokumentasi, penulis juga melakukan studi pustaka, membaca dan mempelajari dokumen-dokumen, buku-buku acuan, serta sumber lainnya yang berkaitan dengan penelitian untuk dijadikan referensi.

3.4 Pengembangan dan Perancangan Sistem

Pada proses pengujian kinerja ini, akan menggunakan jaringan LAN sederhana untuk mensimulasikan komunikasi antara *server* dengan *client* dimana *server* dan *client* terhubung pada satu jaringan yang sama. Pada gambar 3.2 adalah topologi yang akan digunakan pada saat pengujian.



Gambar 3.2 Topologi Jaringan



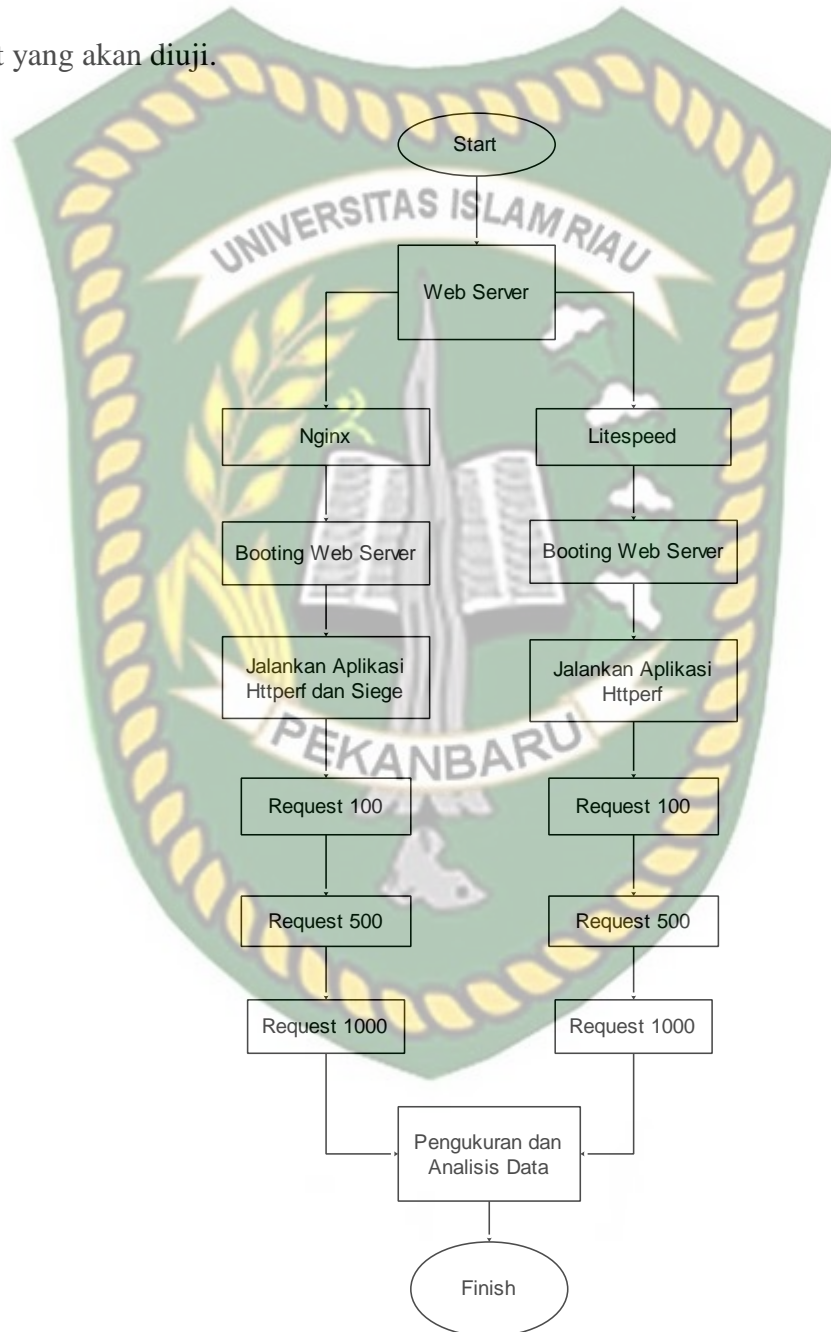
Gambar 3.3 Perancangan Jaringan

Pada gambar 3.3 diatas adalah contoh rancangan jaringan *web server* yang biasa digunakan untuk interaksi antara *client* dan *server* melalui internet. Jika computer *client* ingin mengakses sebuah *website* yang berada pada *server* nginx atau litespeed , maka jalur yang akan dilalui oleh *client* adalah melalui router dan kemudian *client* akan masuk ke internet, di internet lah client akan bisa mengakses *website* menggunakan proses yang disebut *request* yang kemudian akan mendapatkan respon dari *web server* berupa proses yang disebut *reply*. Hasil dari respon yang dikirimkan oleh *web server* adalah berupa tampilan *website* yang di minta oleh *client*.

3.5 Skenario Pengujian

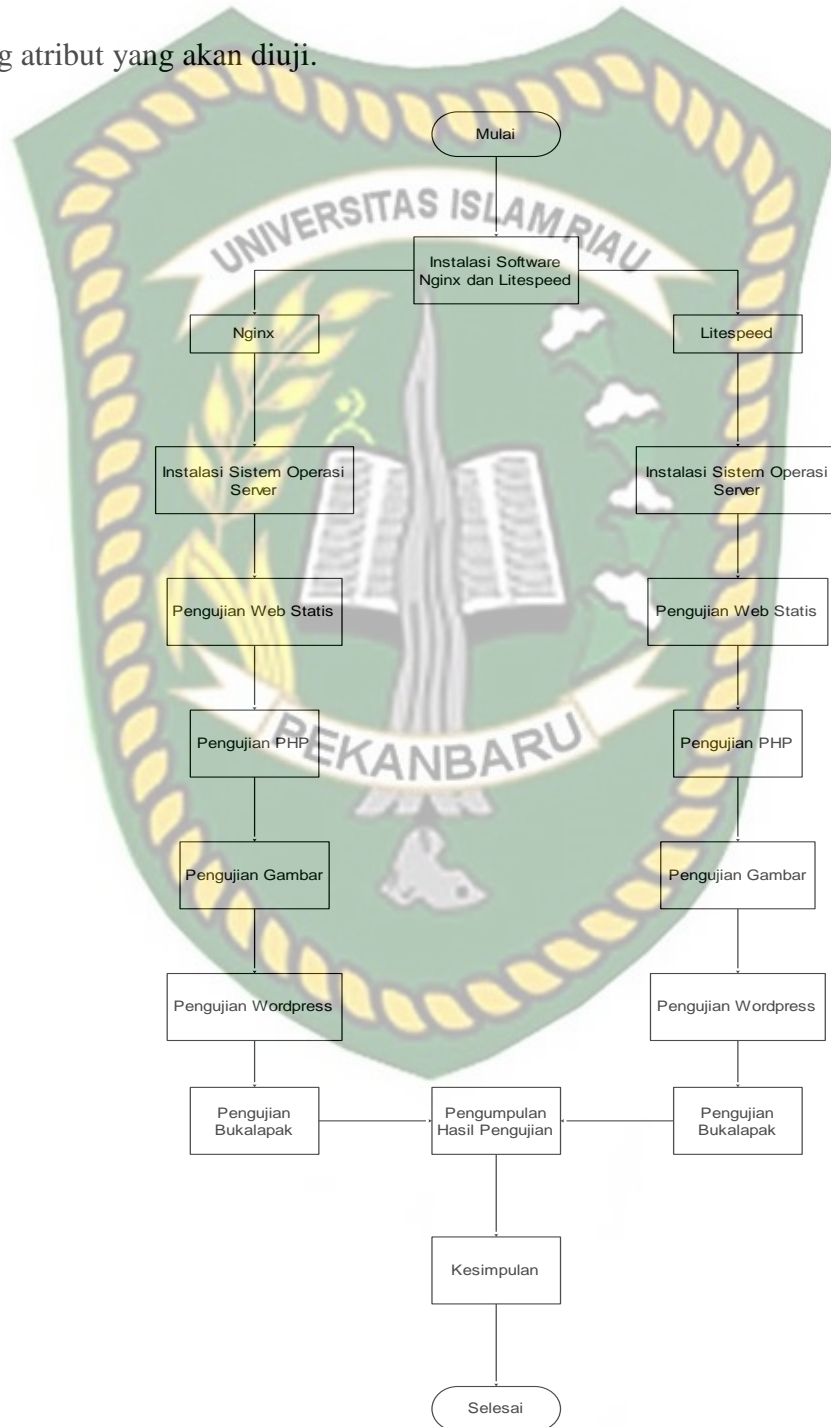
Skenario pengujian dilakukan hanya untuk pengukuran terhadap kinerja *web server* nginx dan litespeed. Pengukuran dilakukan dengan mengambil nilai rata-rata dari masing-masing pengujian yang dilakukan pada halaman web statis, gambar, PHP dan wordpress beserta dengan beban yang diberikan. Target

pengujian beban terhadap kedua *web server* adalah 1000 user yang akan ditambah secara bertahap di mulai dari beban *number connection* 100, 500, dan 1000 dengan *rate/second* setiap beban yaitu 10, 50, dan 100 pada masing-masing atribut yang akan diuji.



Gambar 3.4 Skenario Pengujian

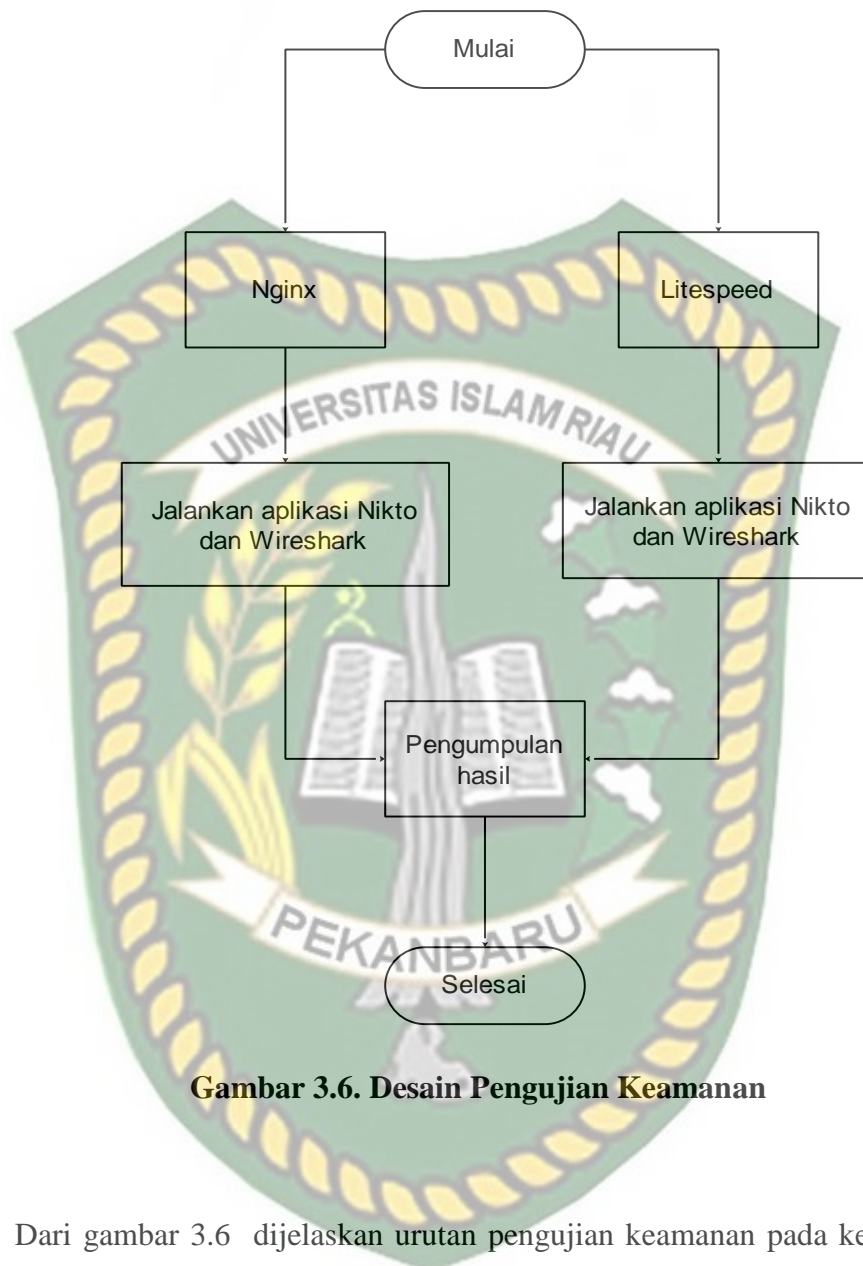
Target pengujian beban terhadap kedua *web server* adalah 1000 user yang akan ditambah secara bertahap di mulai dari beban *number connection* 100, 500, dan 1000 dengan *rate/second* setiap beban yaitu 10, 50, dan 100 pada masing-masing atribut yang akan diuji.



Gambar 3.5 Desain Pengujian

Dari gambar 3.5 diatas telah dijelaskan alur desain pengujian kinerja, atribut yang akan mengukur kinerja dari 2 *web server* adalah *web statis*, gambar, php, dan *wordpress*. Setelah memiliki sistem operasi, tahapan selanjutnya adalah menginstall *web server* dan perangkat lunak pendukungnya. Kemudian tahapan selanjutnya adalah pengujian menggunakan web statis, pengujian menggunakan PHP, pengujian menggunakan gambar, dan pengujian menggunakan wordpress. Setelah semua hasil pengujian didapat dan dikumpulkan maka langkah selanjutnya adalah mengambil kesimpulan dari hasil pengujian-pengujian kedua *web server* tersebut.





Gambar 3.6. Desain Pengujian Keamanan

Dari gambar 3.6 dijelaskan urutan pengujian keamanan pada kedua *web server*. Dengan menggunakan 2 aplikasi maka hasil yang ditampilkan akan berbeda dimana Nikto akan mencoba melakukan penetrasi ke dalam *web server* sehingga mendapatkan informasi penting yang ada di dalam *web server* yang kemudian akan ditampilkan. Pada pengujian Wireshark akan melakukan monitoring paket data antara *server* dan *client*, wireshark akan melihat apakah data yang di *upload* ke server akan dienkripsi atau tidak.

3.5.1 Dokumen Pengujian

1. Web statis

Pengujian pada web statis dilakukan untuk mengetahui kinerja kedua *web server* terhadap halaman yang berisikan tulisan atau teks. Pada halaman web statis ini berisikan tulisan “Hello World” yang juga telah di modifikasi dengan bantuan css dan json.

2. File PHP

Pengujian file PHP dilakukan untuk mengetahui kinerja kedua *web server* dalam menangani halaman web yang dibuat menggunakan PHP.

3. File gambar

Pengujian terhadap file berbentuk gambar dilakukan untuk melihat kinerja kedua *web server* terhadap permintaan file gambar.

4. Wordpress

Pengujian ini dilakukan untuk melihat kinerja kedua *web server* terhadap halaman web yang kompleks.

3.6 Parameter Kinerja

1. *Throughput*

Throughput adalah kecepatan (rate) transfer data efektif yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses dan diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman paket}} \times 8 \text{ byte}$$

2. *Connection* adalah hubungan antara *server* ke *client* dan *client* ke *server*, jika semakin kecil *connection* yang ditunjukkan maka semakin bagus kinerja dari suatu *server*.
3. *Reply* adalah waktu yang dibutuhkan untuk melayani permintaan-permintaan yang di terima, mengacu pada seberapa cepat suatu system/aplikasi dapat merespon suatu aksi, jika *reply time* semakin kecil maka waktu tunggu untuk mengirim balasan ke *web browser* akan semakin pendek dan ini penting untuk kepuasan pemakaian karena akan mengurangi waktu tunggu para pemakai dalam mendapatkan balasan dari aplikasi *web server*.
4. *Request* adalah permintaan *client* terhadap suatu halaman web atau data kepada *server*, jika semakin besar *request* maka suatu perangkat dapat melayani lebih banyak permintaan dalam kurun waktu tertentu.
5. *Error* adalah kesalahan saat *client* mengakses halaman web atau pun saat *server* mengirimkan *reply* ke *client*, jika semakin besar *error* pada sebuah *server* maka semakin buruk pula kinerja suatu *web server*.

$$\text{Error connection client} = \text{demand request} - \text{replied request}$$

6. *Log* adalah suatu file yang berisi data atau informasi mengenai daftar tindakan, kejadian, dan aktivitas yang telah terjadi di dalam suatu

sistem. Informasi tersebut akan berguna apabila sistem mengalami kegagalan sehingga akan dapat dicari penyebabnya berdasarkan log file yang terdapat pada sistem tersebut.

7. *Web Security* adalah keamanan untuk sebuah web atau tata cara mengamankan aplikasi web yang di kelola, biasanya yang bertanggung jawab melakukannya adalah pengelola aplikasi web tersebut. Keamanan pada sebuah *web server* sangat penting karena untuk melindungi data atau halaman *web* dari serangan virus, pencurian data atau bahkan manipulasi. Pada pengujian keamanan ini akan dilakukan peretasan(*hack*) menggunakan aplikasi Nikto dimana peretasan ini bertujuan menguji keamanan data dan beberapa informasi penting yang berkaitan dengan kedua *web server*. Pada pengujian keamanan selanjutnya akan menggunakan aplikasi Wireshark yang akan melihat transaksi data antara *server* dan *client*, pada pengujian menggunakan wireshark ini akan dilihat apakah data yang di upload ke *server* di enkripsi oleh *web server* atau tidak, apabila di enkripsi maka keamanan pada *web server* tersebut bagus dan sebaliknya apabila tidak di enkripsi maka keamanan *web server* tersebut tidak bagus.

BAB IV

HASIL DAN PEMBAHASAN

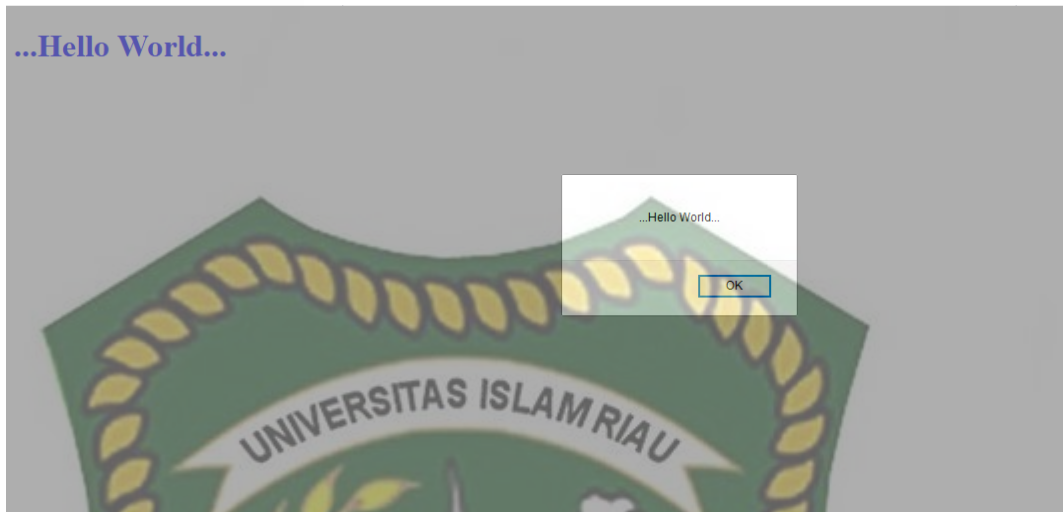
Pengujian kinerja kedua web server dilakukan secara bergantian dengan mendahulukan pengujian terhadap *web server* Nginx, lalu kemudian pengujian dilakukan terhadap *web server* Litespeed.

4.1 Hasil pengujian *web server* Nginx dan Litespeed

Pada saat pengujian akan menguji 5 parameter menggunakan aplikasi Httperf dan Wildshark, dimana aplikasi Httperf akan digunakan untuk melihat *connection*, *error*, *reply*, *request*, dan *log*. Sedangkan aplikasi Wildshark digunakan untuk melihat nilai *throughput*. Sementara untuk pengujian sistem keamanan akan menggunakan aplikasi Nikto.

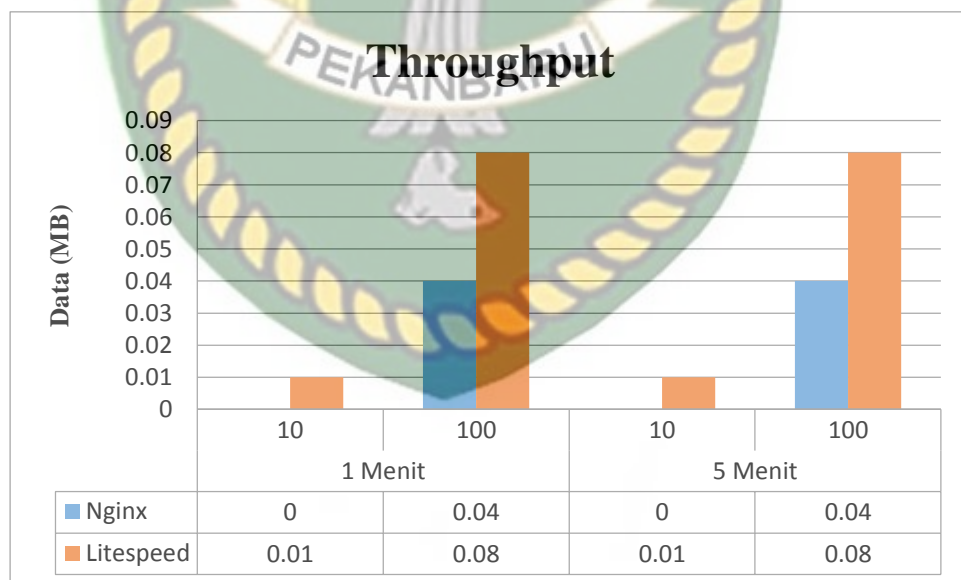
4.1.1 Pengujian terhadap *web statis*

Pengujian yang dilakukan terhadap *web statis* yang berisi teks bertuliskan “Hello Word” yang telah di modifikasi dengan menggunakan css dan json, ukuran dari *web statis* ini adalah 298 *bytes*. *Web statis* ini terdapat dalam folder testing yang di dalamnya terdapat file *index.html*,*css* dan *json*. Berikut adalah tampilan halaman testing beserta dengan pengujiannya.



Gambar 4.1. Tampilan halaman testing

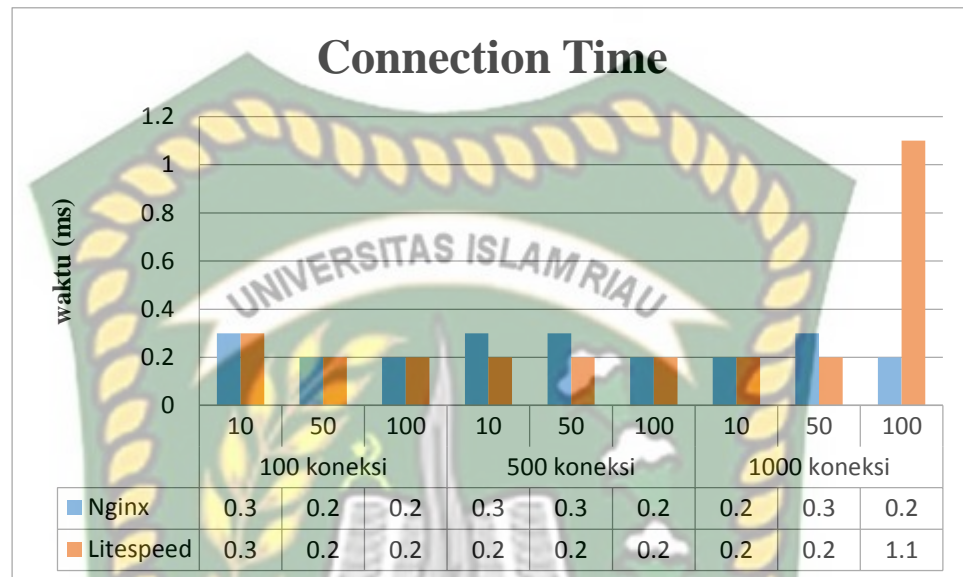
Berikut adalah hasil pengujian *throughput* menggunakan aplikasi Siege terhadap dokumen testing dengan memberikan beban sebanyak 1 dan 100 dan lama waktu pegujiannya adalah 1 dan 5 menit.



Gambar 4.2. Hasil *throughput* halaman testing

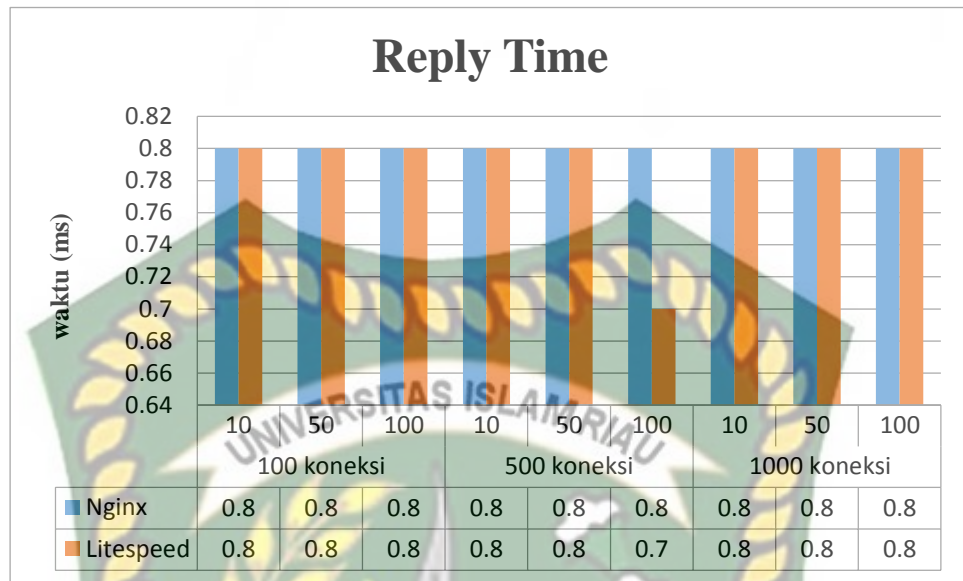
Dari hasil perbandingan yang ditampilkan gambar 4.2 Litespeed memiliki nilai *throughput* yang lebih besar saat di berikan beban 10 dan 100 client dengan

durasi waktu 1 menit dan 5 menit. Dimana *bandwith* yang dihasilkan jauh lebih baik sehingga mampu menampung banyak data.



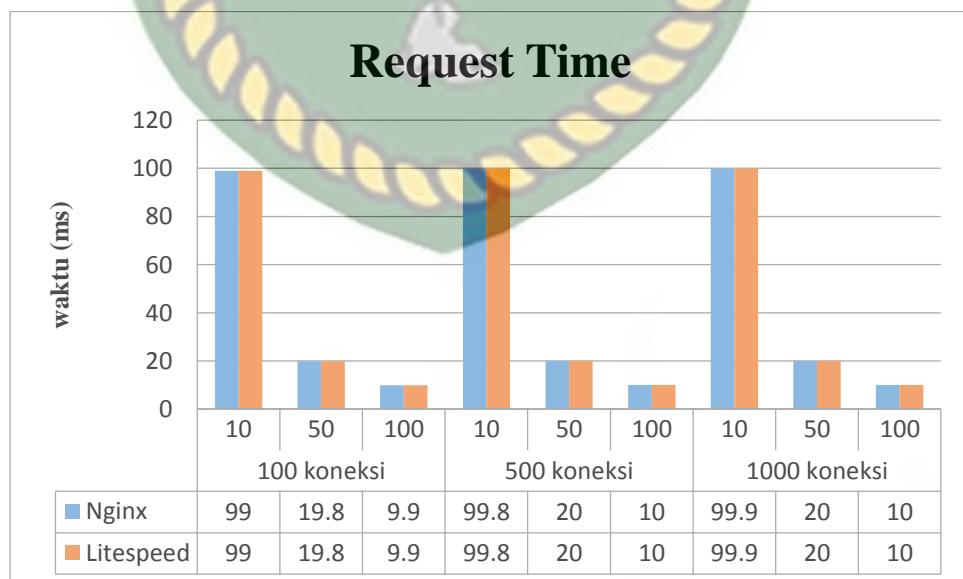
Gambar 4.3. Hasil *connection* halaman testing

Dari hasil perbandingan yang di tampilkan gambar 4.3 pada saat diberikan beban 100 koneksi dan 500 koneksi Litespeed sedikit lebih unggul namun tidak signifikan, akan tetapi pada saat diberikan beban 1000 koneksi dengan 100 request maka akan terlihat bahwa Nginx lebih unggul dari Litespeed. Perbedaan keduanya tidak terlalu signifikan, namun secara keseluruhan Litespeed lebih unggul dimana semakin kecil *connection* yang ditunjukkan maka akan semakin bagus kinerja dari web server tersebut.



Gambar 4.4. Hasil *reply time* halaman testing

Dari hasil perbandingan gambar 4.4 kedua web server tidak menunjukkan perbedaan yang signifikan, hanya pada saat diberikan beban sebanyak 500 koneksi dengan 100 request Litespeed sedikit lebih unggul dibandingkan Nginx karena nilai *reply* Litespeed lebih kecil dari Nginx. Namun secara keseluruhan keduanya memiliki *reply* yang sama-sam baik.



Gambar 4.5. Hasil *request* halaman testing

Dari hasil perbandingan yang ditunjukkan gambar 4.5 terlihat bahwa kedua web server memiliki nilai *request* yang sama, hal ini menunjukkan bahwa dalam hal *request time* kedua web server memiliki kinerja yang sama-sama baik.

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1
Total: connections 100 requests 100 replies 100 test-duration 9.901 s
Connection rate: 10.1 conn/s (99.0 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.8 avg 1.1 max 1.3 median 1.5 stddev 0.1
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000
Request rate: 10.1 req/s (99.0 ms/req)
Request size [B]: 74.0
Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 0.8 transfer 0.0
Reply size [B]: header 214.0 content 194.0 footer 0.0 (total 408.0)
Reply status: 1xx=0 2xx=0 3xx=100 4xx=0 5xx=0
CPU time [s]: user 0.97 system 8.81 (user 9.8% system 89.0% total 98.9%)
Net I/O: 4.8 KB/s (0.0*10^6 bps)
Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@ronaldi-Lenovo-ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.6. Hasil *error* halaman testing Nginx

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1
Total: connections 100 requests 100 replies 100 test-duration 9.901 s
Connection rate: 10.1 conn/s (99.0 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.8 avg 1.1 max 1.3 median 1.5 stddev 0.1
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000
Request rate: 10.1 req/s (99.0 ms/req)
Request size [B]: 75.0
Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 0.8 transfer 0.0
Reply size [B]: header 204.0 content 1172.0 footer 0.0 (total 1376.0)
Reply status: 1xx=0 2xx=0 3xx=100 4xx=0 5xx=0
CPU time [s]: user 1.06 system 8.76 (user 10.7% system 88.5% total 99.2%)
Net I/O: 14.3 KB/s (0.1*10^6 bps)
Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@ronaldi-Lenovo-ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.7. Hasil *error* halaman testing Litespeed

Dari hasil perbandingan yang ditunjukkan gambar 4.6 dan 4.7 tidak terjadi *error* dalam proses pengujian pada kedua web server, hal ini menunjukkan bahwa Nginx dan Litespeed memiliki kinerja yang sama-sama baik.

Table 4.1 Hasil pengujian halaman testing

Beban		Nginx				Litespeed			
nums/conn	rate/seq	C (ms)	Req (ms)	Rep (ms)	Erorr	C (ms)	Req (ms)	Rep (ms)	Erorr
100	10	0.3	99	0.8	0	0.3	99	0.8	0
	50	0.2	19.8	0.8	0	0.2	19.8	0.8	0
	100	0.2	9.9	0.8	0	0.2	9.9	0.8	0
500	10	0.3	99.8	0.8	0	0.2	99.8	0.8	0
	50	0.3	20	0.8	0	0.2	20	0.8	0
	100	0.2	10	0.8	0	0.2	10	0.7	0
1000	10	0.2	99.9	0.8	0	0.2	99.9	0.8	0
	50	0.3	20	0.8	0	0.2	20	0.8	0
	100	0.2	10	0.8	0	1.1	10	0.8	0

Hasil yang ditampilkan oleh table 4.1 adalah rangkuman dari pengujian halaman testing, di dalam tabel 4.1 *connection*, *request*, *reply*, dan *error* adalah parameter yang di ukur menggunakan aplikasi httpperf, nums/conn adalah banyak beban koneksi yang akan diberikan oleh httpperf pada saat pengujian dan rate/seq adalah banyak *request*/detik yang akan di kirimkan pada pengujian. Berikut adalah penjelasan dari hasil pengujian.

1. *Connection*, semakin kecil nilai pada pengujian atau mendekati angka 0 maka lebih baik pula kinerja web server, pada tabel 4.1 nilai *connection* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request*/detik 10, 50, dan 100 maka yang lebih unggul adalah Litespeed, karena dari beberapa kali pengujian terlihat bahwa nilai *connection* Litespeed lebih kecil dibandingkan Nginx, hal ini menunjukkan bahwa *web server* Litespeed lebih baik dalam pengujian *connection* pada halaman testing ini.

2. *Request*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik pula kinerja *web server* dalam menampung permintaan. Pada tabel 4.1 nilai *request* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request*/detik 10, 50, dan 100 maka di dapatkan hasil bahwa Nginx dan Litespeed sama baik, karena nilai yang dihasilkan *httperf* pada saat pengujian tidak berbeda baik untuk Nginx maupun Litespeed.
3. *Reply*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik dan cepat pula kinerja *web server* dalam melayani permintaan. Pada tabel 4.1 nilai *reply* Nginx dan Litespeed tidak ada perbedaan hanya saat diberikan beban 500 koneksi dan 100 *request*/detik maka nilai yang dihasilkan adalah Litespeed lebih baik dibandingkan Nginx, namun secara keseluruhan kinerja keduanya sama baik.
4. *Error*, pada tabel 4.1 menampilkan bahwa kedua *web server* tidak mengalami kesalahan pada saat proses pengujian, hal ini menunjukkan bahwa kinerja kedua *web server* sama baik. Apabila terjadi error pada salah satu *web server* maka kinerja *web server* tersebut kurang baik.

Tabel 4.2 hasil pengujian *throughput* halaman testing

Beban		Nginx	Litespeed
waktu/menit	client	T (mb)	T (mb)
1 Menit	10	0	0.01
	100	0.04	0.08
5 Menit	10	0	0.01
	100	0.04	0.08

Hasil yang ditampilkan oleh table 4.2 adalah hasil dari pengujian halaman testing, di dalam tabel 4.2 *throughput* adalah parameter yang di ukur menggunakan aplikasi siege. Pada pengujian menggunakan aplikasi siege berbeda dengan httpperf karena hasil *request* akan diketahui setelah pengujian, beban yang akan diberikan adalah 10 dan 100 *client* dengan durasi masing-masing pengujian selama 1 menit dan 5 menit. Setelah dilakukan pengujian maka hasilnya adalah *throughput* Litespeed lebih besar dibandingkan Nginx karena semakin besar nilai *throughput* maka akan semakin banyak pula data yang bisa ditampung oleh *web server*. Pada pengujian ini Litespeed lebih unggul diandingkan Nginx.

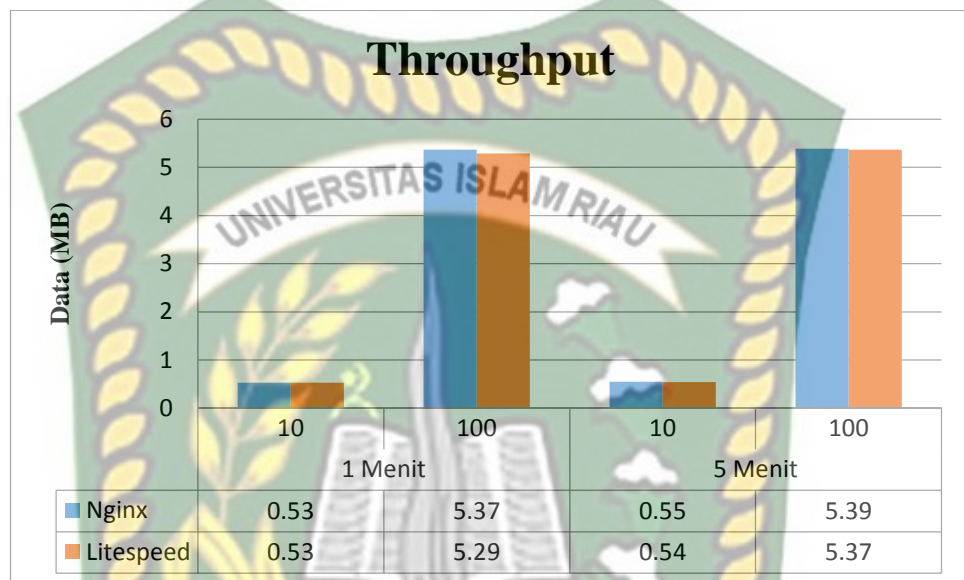
4.1.2 Pengujian terhadap file gambar

Pengujian ini dilakukan terhadap file gambar yang berekstensi .jpg dan berukuran 28493 *bytes*. Dalam pengujian ini nama file gambar tersebut adalah bisa.jpg. Berikut adalah tampilan dari file gambar yang akan digunakan untuk pengujian dan hasil pengujiannya.



Gambar 4.8. Tampilan file gambar

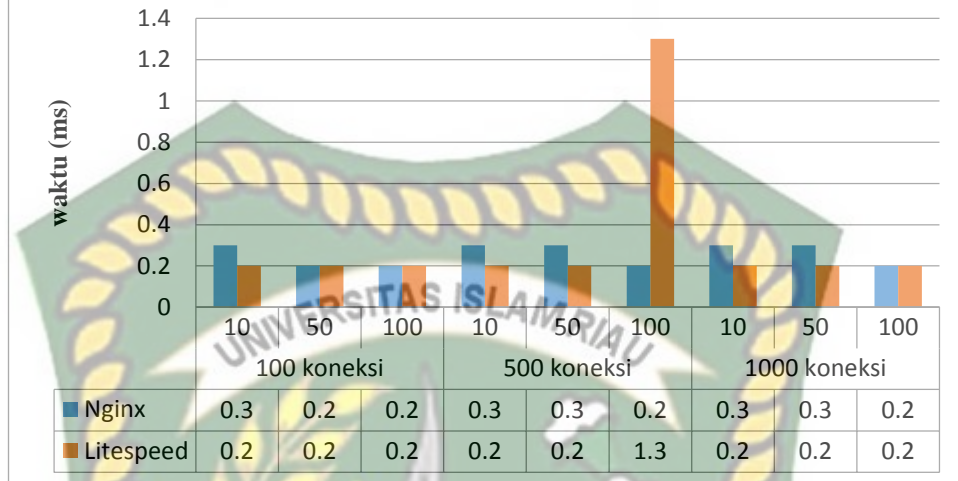
Berikut adalah hasil pengujian *throughput* menggunakan aplikasi Siege terhadap file gambar bisa.jpg dengan memberikan beban sebanyak 1 dan 100 dan lama waktu pegujiannya adalah 1 dan 5 menit.



Gambar 4.9. Hasil *throughput* file gambar

Dari hasil perbandingan yang ditampilkan gambar 4.9 Nginx memiliki nilai *throughput* yang lebih besar saat di berikan beban 10 dan 100 client dengan durasi waktu 1 menit dan 5 menit. Dimana *bandwith* yang dihasilkan jauh lebih baik sehingga mampu menampung banyak data. Namun perbedaan yang ditunjukkan tidak terlalu signifikan.

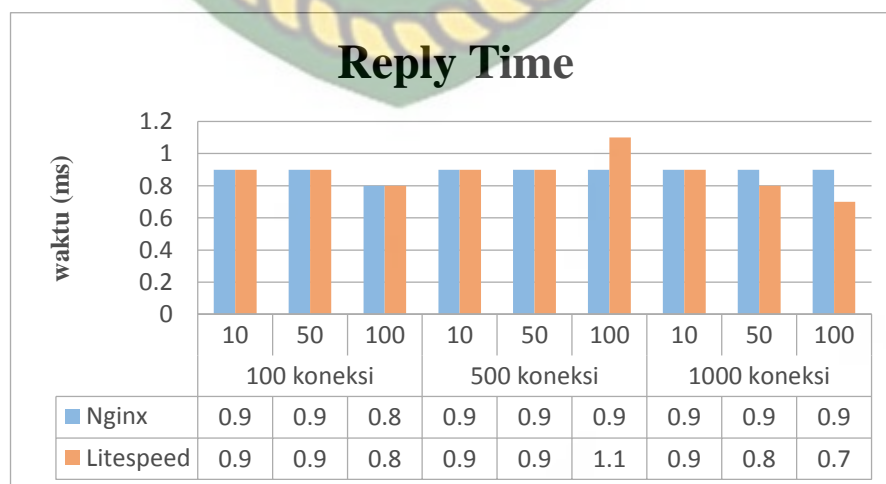
Connection Time



Gambar 4.10. Hasil *connection* file gambar

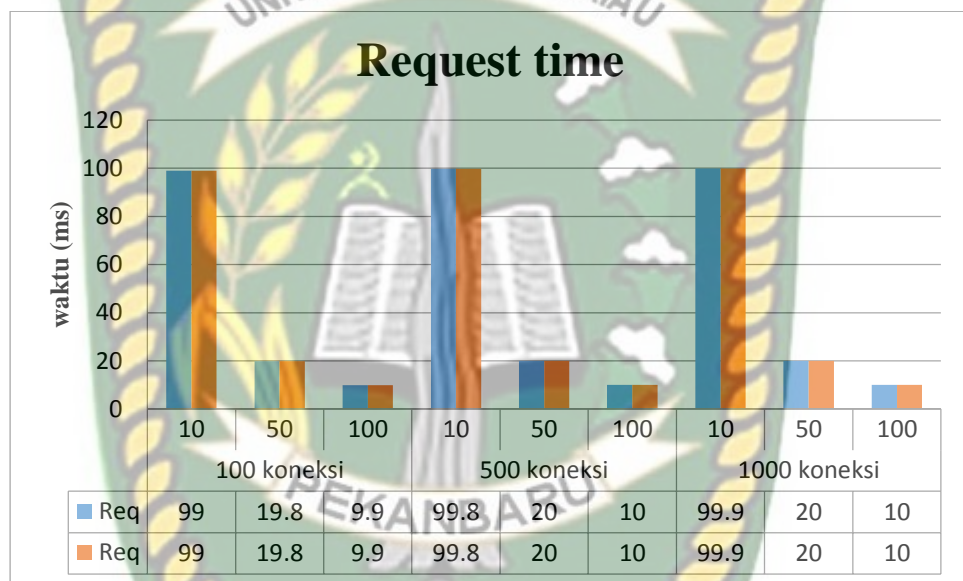
Dari hasil perbandingan yang di tampilkan gambar 4.10 pada saat diberikan beban 100, 500, hingga 1000 koneksi Litespeed sedikit lebih unggul namun tidak signifikan, akan tetapi pada saat diberikan beban 500 koneksi dengan 100 request maka akan terlihat bahwa Nginx lebih unggul dari Litespeed. Perbedaan keduanya tidak terlalu signifikan, namun secara keseluruhan Litespeed lebih unggul dimana semakin kecil *connection* yang ditunjukkan maka akan semakin bagus kinerja dari web server tersebut.

Reply Time



Gambar 4.11. Hasil *reply time* file gambar

Dari hasil perbandingan gambar 4.11 kedua web server tidak menunjukkan perbedaan yang signifikan, hanya pada saat diberikan beban sebanyak 500 koneksi dengan 100 request Nginx sedikit lebih unggul dibandingkan Nginx karena nilai *reply* Nginx lebih kecil dari Litespeed. Namun pada saat beban 1000 koneksi dengan 50 dan 100 *request* maka yang lebih unggul adalah Litespeed. Secara keseluruhan keduanya memiliki nilai *reply* yang sama-sama baik.



Gambar 4.12. Hasil *request time* file gambar

Dari hasil perbandingan yang ditunjukkan gambar 4.12 terlihat bahwa kedua web server memiliki nilai *request* yang sama, hal ini menunjukkan bahwa dalam hal *request time* kedua web server memiliki kinerja yang sama-sama baik.

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1

Total: connections 100 requests 100 replies 100 test-duration 9.904 s

Connection rate: 10.1 conn/s (99.0 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 3.3 avg 3.6 max 5.6 median 3.5 stddev 0.2
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000

Request rate: 10.1 req/s (99.0 ms/req)
Request size [B]: 75.0

Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 0.9 transfer 2.4
Reply size [B]: header 325.0 content 28493.0 footer 0.0 (total 28818.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.12 system 8.72 (user 11.4% system 88.0% total 99.4%)
Net I/O: 284.9 KB/s (2.3*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi#

```

Gambar 4.13. Hasil *error* file gambar pada Nginx

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1

Total: connections 100 requests 100 replies 100 test-duration 9.904 s

Connection rate: 10.1 conn/s (99.0 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 3.3 avg 3.6 max 5.6 median 3.5 stddev 0.2
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000

Request rate: 10.1 req/s (99.0 ms/req)
Request size [B]: 75.0

Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 0.9 transfer 2.4
Reply size [B]: header 325.0 content 28493.0 footer 0.0 (total 28818.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.12 system 8.72 (user 11.4% system 88.0% total 99.4%)
Net I/O: 284.9 KB/s (2.3*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi#

```

Gambar 4.14. Hasil *error* file gambar pada Litespeed

Dari hasil perbandingan yang ditunjukkan gambar 4.13 dan 4.14 tidak terjadi *error* dalam proses pengujian pada kedua web server, hal ini menunjukkan bahwa Nginx dan Litespeed memiliki kinerja yang sama-sama baik.

Table 4.3 Hasil pengujian bisa.jpg

Beban		Nginx				Litespeed			
nums/conn	rate/seq	C (ms)	Req (ms)	Rep (ms)	Erorr	C (ms)	Req (ms)	Rep (ms)	Erorr
100	10	0.3	99	0.9	0	0.2	99	0.9	0
	50	0.2	19.8	0.9	0	0.2	19.8	0.9	0
	100	0.2	9.9	0.8	0	0.2	9.9	0.8	0
500	10	0.3	99.8	0.9	0	0.2	99.8	0.9	0
	50	0.3	20	0.9	0	0.2	20	0.9	0
	100	0.2	10	0.9	0	1.3	10	1.1	0
1000	10	0.3	99.9	0.9	0	0.2	99.9	0.9	0
	50	0.3	20	0.9	0	0.2	20	0.8	0
	100	0.2	10	0.9	0	0.2	10	0.7	0

Hasil yang ditampilkan oleh table 4.2 adalah rangkuman dari pengujian bisa.jpg, di dalam tabel 4.2 *connection*, *request*, *reply*, dan *error* adalah parameter yang di ukur menggunakan aplikasi httpperf, nums/conn adalah banyak beban koneksi yang akan diberikan oleh httpperf pada saat pengujian dan rate/seq adalah banyak *request*/detik yang akan di kirimkan pada pengujian. Berikut adalah penjelasan dari hasil pengujian.

1. *Connection*, semakin kecil nilai pada pengujian atau mendekati angka 0 maka lebih baik pula kinerja web server, pada tabel 4.2 nilai *connection* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request*/detik 10, 50, dan 100 maka yang lebih unggul adalah Litespeed, karena dari beberapa kali pengujian terlihat bahwa nilai *connection* Litespeed lebih kecil dibandingkan Nginx, namun perbedaan yang ditampilkan tidak terlalu signifikan hanya pada saat beban koneksi 500 dengan *request* 100 maka terjadi perbedaan yang cukup signifikan, secara keseluruhan menunjukkan bahwa *web server*

Litespeed lebih baik dalam pengujian *connection* pada file gambar bisa.jpg ini.

2. *Request*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik pula kinerja *web server* dalam menampung permintaan. Pada tabel 4.2 nilai *request* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request/detik* 10, 50, dan 100 maka di dapatkan hasil bahwa Nginx dan Litespeed sama baik, karena nilai yang dihasilkan *httpperf* pada saat pengujian tidak berbeda baik untuk Nginx maupun Litespeed.
3. *Reply*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik dan cepat pula kinerja *web server* dalam melayani permintaan. Pada tabel 4.1 nilai *reply* Nginx dan Litespeed tidak ada perbedaan yang signifikan, secara keseluruhan kinerja keduanya sama baik.
4. *Error*, pada tabel 4.2 menampilkan bahwa kedua *web server* tidak mengalami kesalahan pada saat proses pengujian, hal ini menunjukkan bahwa kinerja kedua *web server* sama baik. Apabila terjadi error pada salah satu *web server* maka kinerja *web server* tersebut kurang baik.

Tabel 4.4 hasil pengujian *throughput* bisa.jpg

Beban		Nginx	Litespeed
waktu/menit	koneksi	T (mb)	T (mb)
1 Menit	10	0.53	0.53
	100	5.37	5.29
5 Menit	10	0.55	0.54
	100	5.39	5.37

Hasil yang ditampilkan oleh table 4.4 adalah hasil dari pengujian bisa.jpg, di dalam tabel 4.4 *throughput* adalah parameter yang di ukur menggunakan aplikasi siege. Pada pengujian menggunakan aplikasi siege berbeda dengan httpperf karena hasil *request* akan diketahui setelah pengujian, beban yang akan diberikan adalah 10 dan 100 *client* dengan durasi masing-masing pengujian selama 1 menit dan 5 menit. Setelah dilakukan pengujian maka hasilnya adalah *throughput* Nginx lebih besar dibandingkan Litespeed karena semakin besar nilai *throughput* maka akan semakin banyak pula data yang bisa ditampung oleh *web server*. Pada pengujian ini Nginx lebih unggul diandingkan Litespeed.

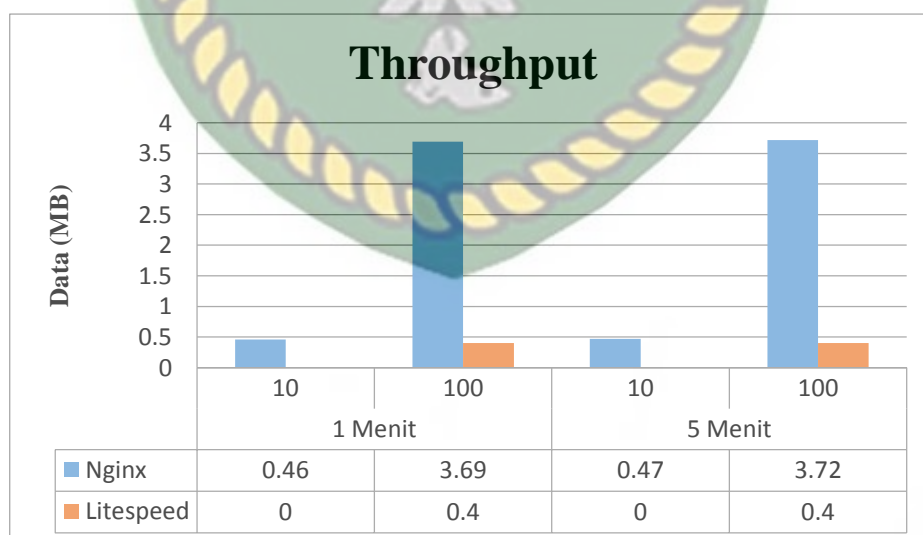
4.1.3 Pengujian terhadap file PHP

Pengujian yang dilakukan terhadap file php yang berisi tentang informasi php yang digunakan oleh kedua web server yang biasa disebut phpinfo, file ini hanya berukuran 22 *bytes*. Nama dari file php yang akan digunakan pada pengujian ini dalah info.php. Berikut adalah tampilan file php beserta dengan pengujiannya.

PHP Version 7.0.33-0ubuntu0.16.04.7	
System	Linux ubuntu 4.4.0-166-generic #195-Ubuntu SMP Tue Oct 1 09:36:25 UTC 2019 i686
Server API	FPMFastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/fpm
Loaded Configuration File	/etc/php/7.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/fpm/conf.d
Additional .ini files parsed	/etc/php/7.0/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.0/fpm/conf.d/10-openssl.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/15-xml.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-ctype.ini, /etc/php/7.0/fpm/conf.d/20-curl.ini, /etc/php/7.0/fpm/conf.d/20-dom.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-ffi.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gdlib.ini, /etc/php/7.0/fpm/conf.d/20-gettext.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-ldap.ini, /etc/php/7.0/fpm/conf.d/20-ldap_sasl.ini, /etc/php/7.0/fpm/conf.d/20-mbstring.ini, /etc/php/7.0/fpm/conf.d/20-mcrypt.ini, /etc/php/7.0/fpm/conf.d/20-mysql.ini, /etc/php/7.0/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.0/fpm/conf.d/20-phar.ini, /etc/php/7.0/fpm/conf.d/20-posix.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-simplexml.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-systems.ini, /etc/php/7.0/fpm/conf.d/20-tidy.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini, /etc/php/7.0/fpm/conf.d/20-wddx.ini, /etc/php/7.0/fpm/conf.d/20-xsl.ini, /etc/php/7.0/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.0/fpm/conf.d/20-xml.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled

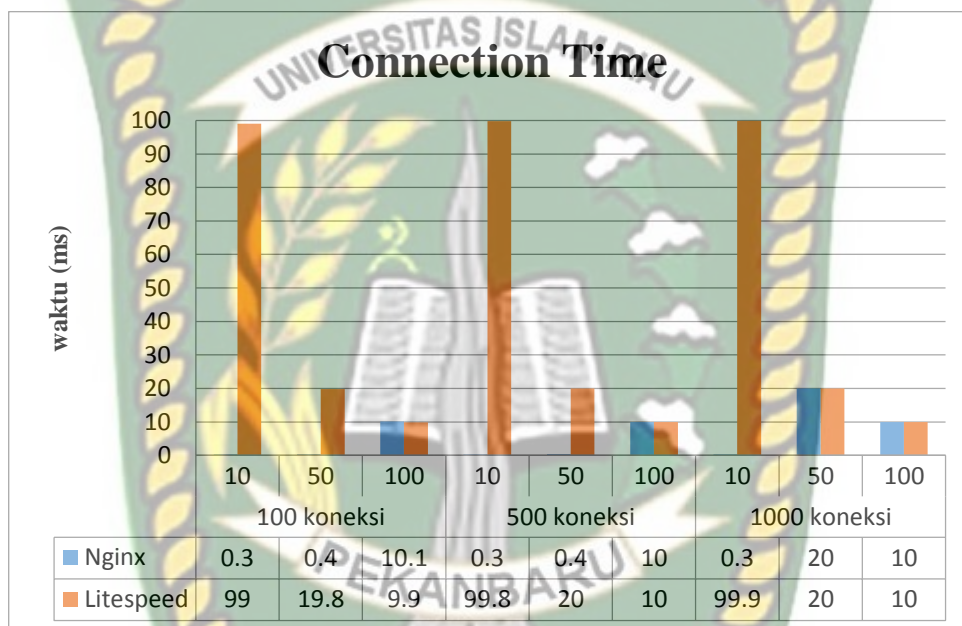
Gambar 4.15. Tampilan halaman info.php

Berikut adalah hasil pengujian *throughput* menggunakan aplikasi Siege terhadap file gambar bisa.jpg dengan memberikan beban sebanyak 1 dan 100 dan lama waktu pegujiannya adalah 1 dan 5 menit.



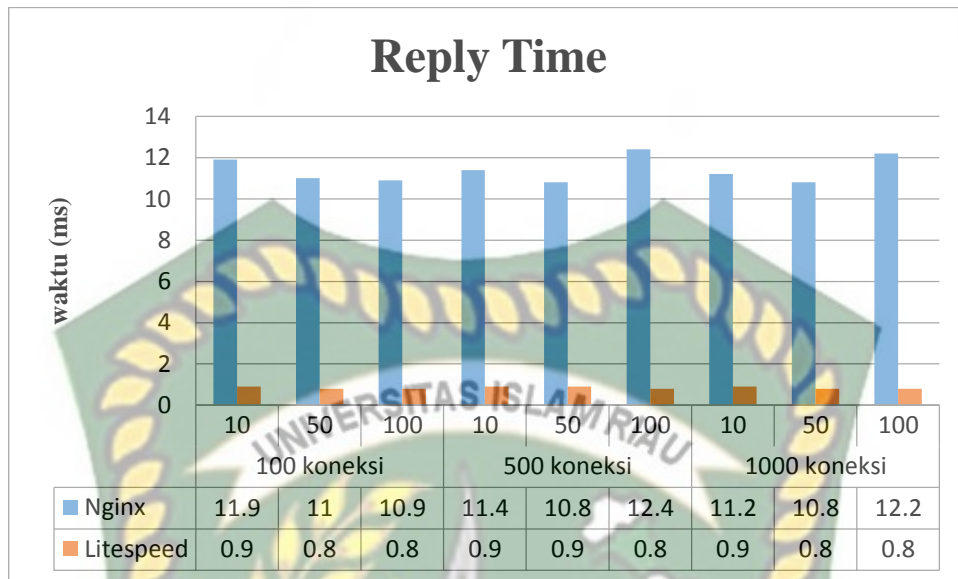
Gambar 4.16. Hasil *throughput* info.php

Dari hasil perbandingan yang ditampilkan gambar 4.16 Nginx memiliki nilai *throughput* yang lebih besar saat di berikan beban 10 dan 100 client dengan durasi waktu 1 menit dan 5 menit. Dimana *bandwith* yang dihasilkan jauh lebih baik sehingga mampu menampung banyak data. Hal ini menunjukkan bahwa dalam Nginx jauh lebih unggul dibandingkan dengan Litespeed



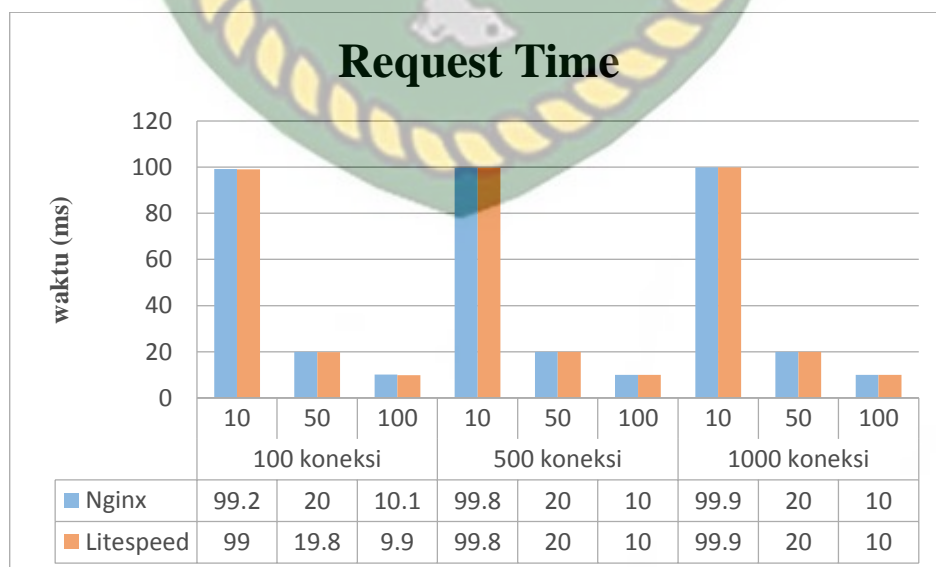
Gambar 4.17. Hasil *connection info.php*

Dari hasil perbandingan yang di tampilkan gambar 4.17 pada saat diberikan beban 100, 500, hingga 1000 koneksi Nginx lebih unggul daripada Litespeed, namun ada saat Nginx dan Litespeed seimbang dalam hal pengujian *connection* ini, contohnya saja pada saat 100 request maka akan terlihat bahwa kedua web server seimbang baik diberikan beban 100 koneksi, 500 koneksi, maupun 1000 koneksi. Secara keseluruhan Nginx lebih unggul dimana semakin kecil *connection* yang ditunjukkan maka akan semakin bagus kinerja dari web server tersebut.



Gambar 4.18. Hasil *reply time* info.php

Dari hasil perbandingan yang ditampilkan oleh gambar 4.18 terlihat bahwa nilai *reply* Litespeed lebih kecil daripada Nginx. Hal ini menunjukkan bahwa Litespeed lebih unggul karena semakin kecil waktu *reply* maka akan semakin bagus dan cepat pula kinerja dari web server.



Gambar 4.19. Hasil *request time* info.php

Dari hasil perbandingan yang ditunjukkan gambar 4.19 terlihat bahwa kedua web server memiliki nilai *request* yang sama dan perbedaan antar Nginx dan Litespeed tidak terlalu signifikan, hal ini menunjukkan bahwa dalam hal *request time* kedua web server memiliki kinerja yang sama-sama baik.

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1
Total: connections 100 requests 100 replies 100 test-duration 9.921 s
Connection rate: 10.1 conn/s (99.2 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 17.9 avg 20.5 max 69.6 median 20.5 stddev 5.6
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000
Request rate: 10.1 req/s (99.2 ms/req)
Request size [B]: 75.0
Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 11.9 transfer 8.2
Reply size [B]: header 179.0 content 88947.0 footer 2.0 (total 89128.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0
CPU time [s]: user 1.08 system 8.78 (user 10.8% system 88.5% total 99.3%)
Net I/O: 878.0 KB/s (7.2*10^6 bps)
Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 fltab-full 0 other 0
root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi#

```

Gambar 4.20. Hasil *error* info.php pada Nginx

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1
Total: connections 100 requests 100 replies 100 test-duration 9.901 s
Connection rate: 10.1 conn/s (99.0 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.9 avg 1.1 max 1.3 median 1.5 stddev 0.1
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000
Request rate: 10.1 req/s (99.0 ms/req)
Request size [B]: 79.0
Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 0.9 transfer 0.0
Reply size [B]: header 248.0 content 195.0 footer 0.0 (total 443.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=100 5xx=0
CPU time [s]: user 0.89 system 8.90 (user 9.0% system 89.9% total 98.9%)
Net I/O: 5.1 KB/s (0.0*10^6 bps)
Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 fltab-full 0 other 0
root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi#

```

Gambar 4.21. Hasil *error* info.php pada Litespeed

Dari hasil perbandingan yang ditunjukkan gambar 4.20 dan 4.21 tidak terjadi *error* dalam proses pengujian pada kedua web server, hal ini menunjukkan bahwa Nginx dan Litespeed memiliki kinerja yang sama-sama baik.

Table 4.5 Hasil pengujian file PHP

Beban		Nginx				Litespeed			
nums/conn	rate/seq	C (ms)	Req (ms)	Rep (ms)	Erorr	C (ms)	Req (ms)	Rep (ms)	Erorr
100	10	0.3	99.2	11.9	0	99	99	0.9	0
100	50	0.4	20	11	0	19.8	19.8	0.8	0
100	100	10.1	10.1	10.9	0	9.9	9.9	0.8	0
500	10	0.3	99.8	11.4	0	99.8	99.8	0.9	0
500	50	0.4	20	10.8	0	20	20	0.9	0
500	100	10	10	12.4	0	10	10	0.8	0
1000	10	0.3	99.9	11.2	0	99.9	99.9	0.9	0
1000	50	20	20	10.8	0	20	20	0.8	0
1000	100	10	10	12.2	0	10	10	0.8	0

Hasil yang ditampilkan oleh table 4.5 adalah rangkuman dari pengujian file PHP, di dalam tabel 4.5 *connection*, *request*, *reply*, dan *error* adalah parameter yang di ukur menggunakan aplikasi *httperf*, *nums/conn* adalah banyak beban koneksi yang akan diberikan oleh *httperf* pada saat pengujian dan *rate/seq* adalah banyak *request/detik* yang akan di kirimkan pada pengujian. Berikut adalah penjelasan dari hasil pengujian.

1. *Connection*, semakin kecil nilai pada pengujian atau mendekati angka 0 maka lebih baik pula kinerja web server, pada tabel 4.5 nilai *connection* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request/detik* 10, 50, dan 100 maka yang lebih unggul adalah Litespeed, Nginx karena dari beberapa kali pengujian terlihat bahwa nilai *connection* Nginx lebih kecil dibandingkan Litespeed, namun pada saat beban 100 koneksi dengan 100 *request* maka

Litespeed sedikit lebih unggul dibandingkan Nginx, secara keseluruhan Nginx lebih unggul daripada Litespeed.

2. *Request*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik pula kinerja *web server* dalam menampung permintaan. Pada tabel 4.5 nilai *request* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request/detik* 10, 50, dan 100 maka di dapatkan hasil bahwa Nginx dan Litespeed sama baik, perbedaan yang ditampilkan tidak terlalu signifikan.
3. *Reply*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik dan cepat pula kinerja *web server* dalam melayani permintaan. Pada tabel 4.5 nilai *reply* Litespeed jauh lebih unggul dibandingkan Nginx. Sehingga Litespeed lebih cepat dalam melayani permintaan.
4. *Error*, pada tabel 4.5 menampilkan bahwa kedua *web server* tidak mengalami kesalahan pada saat proses pengujian, hal ini menunjukkan bahwa kinerja kedua *web server* sama baik. Apabila terjadi error pada salah satu *web server* maka kinerja *web server* tersebut kurang baik.

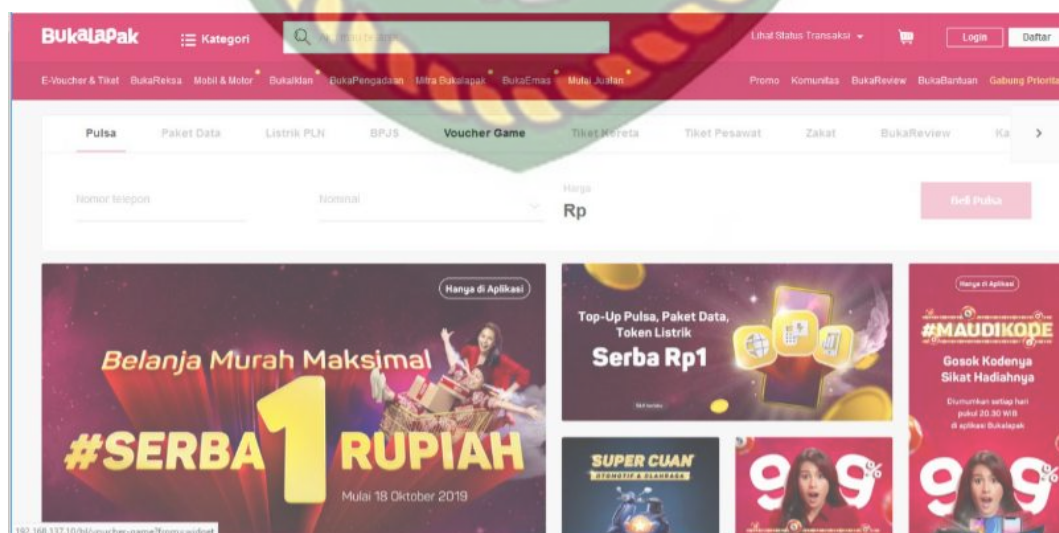
Tabel 4.6 Hasil pengujian *throughput* file php

Beban		Nginx	Litespeed
Waktu/menit	client	T (mb)	T (mb)
1 Menit	10	0.46	0
	100	3.69	0.4
5 Menit	10	0.47	0
	100	3.72	0.4

Hasil yang ditampilkan oleh tabel 4.6 adalah hasil dari pengujian file PHP, di dalam tabel 4.6 *throughput* adalah parameter yang di ukur menggunakan aplikasi siege. Pada pengujian menggunakan aplikasi siege berbeda dengan httpperf karena hasil *request* akan diketahui setelah pengujian, beban yang akan diberikan adalah 10 dan 100 *client* dengan durasi masing-masing pengujian selama 1 menit dan 5 menit. Setelah dilakukan pengujian maka hasilnya adalah *throughput* Nginx lebih besar dibandingkan Litespeed karena semakin besar nilai *throughput* maka akan semakin banyak pula data yang bisa ditampung oleh *web server*. Pada pengujian ini Nginx lebih unggul diandingkan Litespeed.

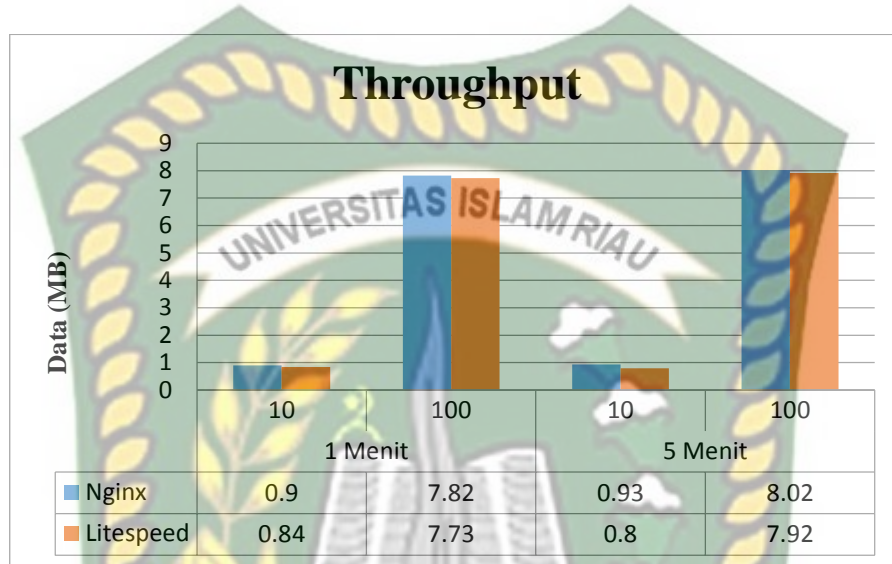
4.1.4 Pengujian terhadap halaman web Bukalapak

Pengujian ini dilakukan terhadap halaman utama dari web online shop Bukalapak berukuran 232190 *bytes*. Dalam pengujian ini nama file web bukhalapak adalah bukhalapak.html. Berikut adalah tampilan dari halaman web bukhalapak yang akan digunakan untuk pengujian dan hasil pengujiannya.



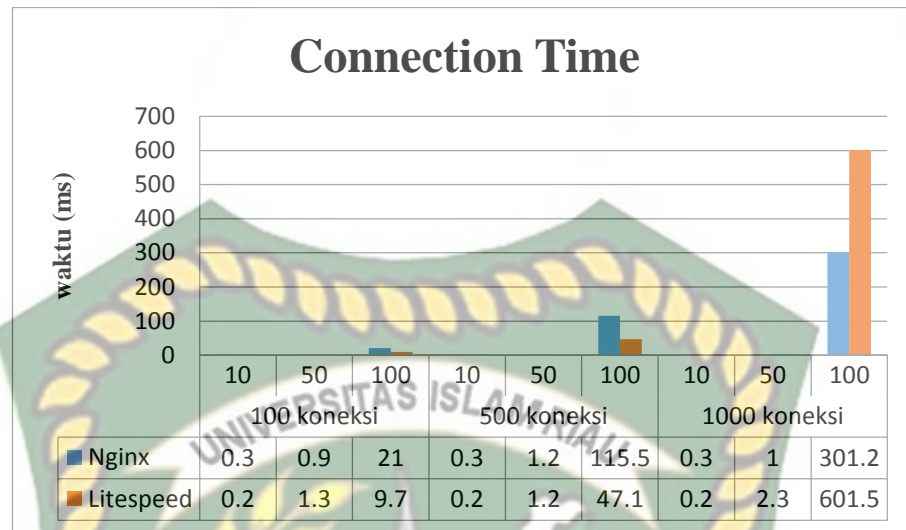
Gambar 4.22. Tampilan halaman web Bukalapak

Berikut adalah hasil pengujian *throughput* menggunakan aplikasi Siege terhadap halaman web bukalapak dengan memberikan beban sebanyak 1 dan 100 dan lama waktu pegujiannya adalah 1 dan 5 menit.



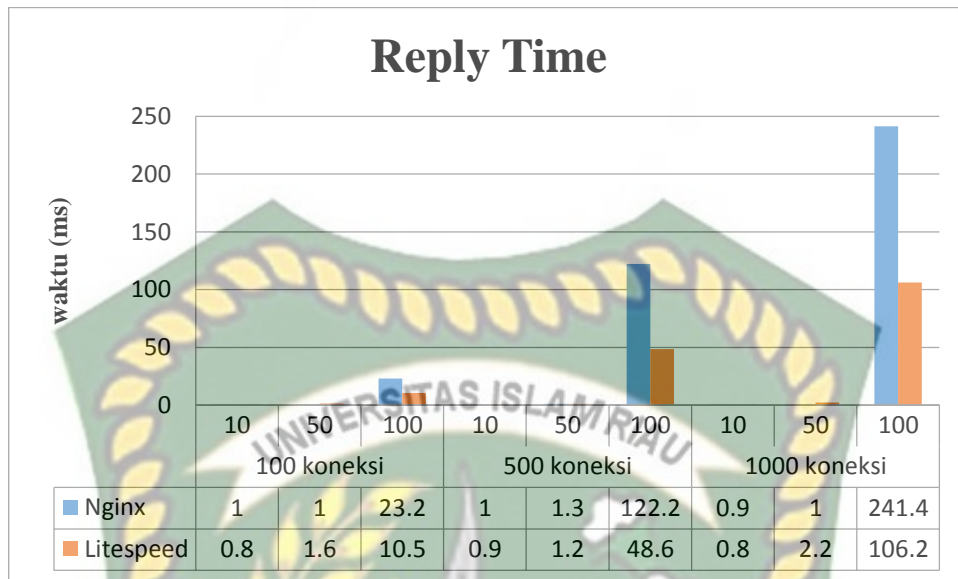
Gambar 4.23. Hasil *throughput* bukalapak.html

Dari hasil perbandingan yang ditampilkan gambar 4.23 Nginx memiliki nilai *throughput* yang lebih besar saat di berikan beban 10 dan 100 client dengan durasi waktu 1 menit dan 5 menit. Dimana *bandwith* yang dihasilkan jauh lebih baik sehingga mampu menampung banyak data. Namun perbedaan yang ditunjukkan diantara kedua *web server* tidak terlalu signifikan.



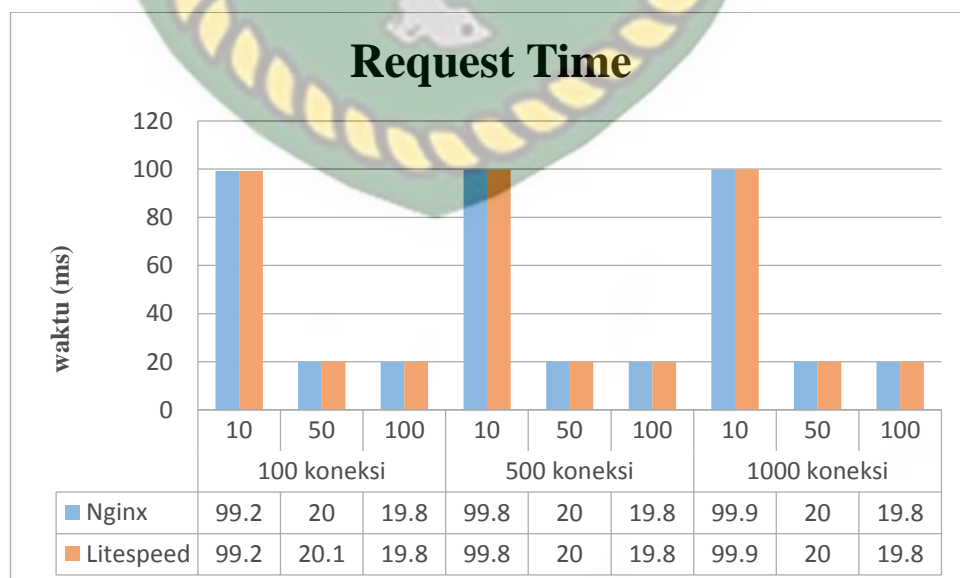
Gambar 4.24. Hasil *connection time* bukalahpak.html

Dari hasil perbandingan yang ditampilkan gambar 4.24 Nginx dan Litespeed mempunyai nilai *connection* yang tidak jauh berbeda, perbedaan yang sangat signifikan terjadi pada saat beban 100, dan 500 koneksi dengan 100 request maka yang jauh lebih unggul adalah Litespeed. Sedangkan pada saat beban 1000 koneksi dan 100 request maka yang yang unggul justru Nginx, jika diambil hasil dari pengujian tersebut maka Litespeed lebih unggul dibandingkan Nginx.



Gambar 4.25 Hasil *reply time* bukalapak.html

Dari hasil perbandingan yang ditampilkan oleh gambar 4.25 terlihat bahwa nilai *reply* Litespeed lebih kecil daripada Nginx. Hal ini menunjukkan bahwa Litespeed melayani permintaan lebih cepat daripada Nginx, sehingga dalam hal ini menunjukkan bahwa Litespeed lebih unggul.



Gambar 4.26. Hasil *request time* bukalapak.html

Dari hasil perbandingan yang ditunjukkan gambar 4.26 terlihat bahwa kedua web server memiliki nilai *request* yang sama dan perbedaan antar Nginx dan Litespeed tidak terlalu signifikan, hal ini menunjukkan bahwa dalam hal *request time* kedua web server memiliki kinerja yang sama-sama baik.

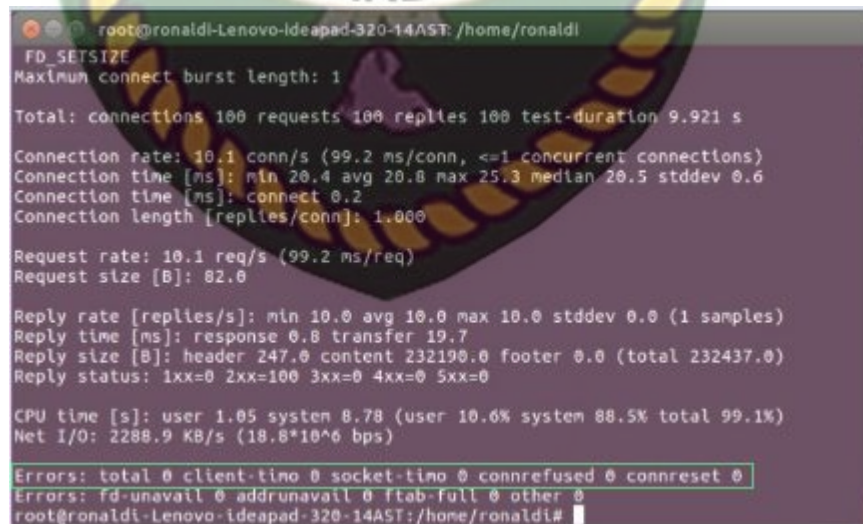


```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1
Total: connections 100 requests 100 replies 100 test-duration 9.921 s
Connection rate: 10.1 conn/s (99.2 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 20.6 avg 21.0 max 29.2 median 20.5 stddev 0.8
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000
Request rate: 10.1 req/s (99.2 ms/req)
Request size [B]: 81.0
Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 1.0 transfer 19.7
Reply size [B]: header 252.0 content 232190.0 footer 0.0 (total 232442.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0
CPU time [s]: user 0.93 system 8.90 (user 9.4% system 89.8% total 99.1%)
Net I/O: 2288.8 KB/s (18.7*10^6 bps)
Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@ronaldi-Lenovo-ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.27. Hasil *error* bukalapak.html Nginx



```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1
Total: connections 100 requests 100 replies 100 test-duration 9.921 s
Connection rate: 10.1 conn/s (99.2 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 20.4 avg 20.8 max 25.3 median 20.5 stddev 0.6
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000
Request rate: 10.1 req/s (99.2 ms/req)
Request size [B]: 82.0
Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 0.8 transfer 19.7
Reply size [B]: header 247.0 content 232190.0 footer 0.0 (total 232437.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0
CPU time [s]: user 1.05 system 8.78 (user 10.6% system 88.5% total 99.1%)
Net I/O: 2288.9 KB/s (18.8*10^6 bps)
Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
root@ronaldi-Lenovo-ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.28. Hasil *error* bukalapak.html Litespeed

Dari hasil perbandingan yang ditunjukkan gambar 4.27 dan 4.28 tidak terjadi *error* dalam proses pengujian pada kedua web server, hal ini menunjukkan bahwa Nginx dan Litespeed memiliki kinerja yang sama-sama baik karena tidak ada terjadi *error* pada saat pengujian.

Table 4.7 Hasil pengujian Bukalapak

Beban		Nginx				Litespeed			
nums/conn	rate/seq	C (ms)	Req (ms)	Rep (ms)	Erorr	C (ms)	Req (ms)	Rep (ms)	Erorr
100	10	0.3	99.2	1	0	0.2	99.2	0.8	0
	50	0.9	20	1	0	1.3	20.1	1.6	0
	100	21	19.8	23.2	0	9.7	19.8	10.5	0
500	10	0.3	99.8	1	0	0.2	99.8	0.9	0
	50	1.2	20	1.3	0	1.2	20	1.2	0
	100	115.5	19.8	122.2	0	47.1	19.8	48.6	0
1000	10	0.3	99.9	0.9	0	0.2	99.9	0.8	0
	50	1	20	1	0	2.3	20	2.2	0
	100	301.2	19.8	241.4	0	601.5	19.8	106.2	0

Hasil yang ditampilkan oleh table 4.7 adalah rangkuman dari pengujian halaman bukalapak, di dalam tabel 4.7 *connection*, *request*, *reply*, dan *error* adalah parameter yang di ukur menggunakan aplikasi httpperf, nums/conn adalah banyak beban koneksi yang akan diberikan oleh httpperf pada saat pengujian dan rate/seq adalah banyak *request*/detik yang akan di kirimkan pada pengujian. Berikut adalah penjelasan dari hasil pengujian.

1. *Connection*, semakin kecil nilai pada pengujian atau mendekati angka 0 maka lebih baik pula kinerja web server, pada tabel 4.7 nilai *connection* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request*/detik 10, 50, dan 100 maka hasil yang di dapat bahwa pada saat beban 100 *request*/detik akan terlihat perbedaan yang

signifikan dimana saat beban 100 dan 500 koneksi Litespeed unggul akan tetapi pada saat beban 1000 koneksi Nginx lebih unggul dibandingkan Litespeed, secara keseluruhan Litespeed lebih unggul daripada Nginx.

2. *Request*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik pula kinerja *web server* dalam menampung permintaan. Pada tabel 4.7 nilai *request* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request/detik* 10, 50, dan 100 maka di dapatkan hasil bahwa Nginx dan Litespeed sama baik, perbedaan yang ditampilkan tidak terlalu signifikan.
3. *Reply*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik dan cepat pula kinerja *web server* dalam melayani permintaan. Pada tabel 4.7 nilai *reply* Litespeed jauh lebih unggul dibandingkan Nginx. Sehingga Litespeed lebih cepat dalam melayani permintaan.
4. *Error*, pada tabel 4.7 menampilkan bahwa kedua *web server* tidak mengalami kesalahan pada saat proses pengujian, hal ini menunjukkan bahwa kinerja kedua *web server* sama baik. Apabila terjadi error pada salah satu *web server* maka kinerja *web server* tersebut kurang baik.

Tabel 4.8 Hasil *throughput* bukalapak

Beban		Nginx	Litespeed
client	waktu/menit	T (mb)	T (mb)
1 Menit	10	0.9	0.84
	100	7.82	7.73
5 Menit	10	0.93	0.8
	100	8.02	7.92

Hasil yang ditampilkan oleh tabel 4.8 adalah hasil dari pengujian halaman bukalapak, di dalam tabel 4.8 *throughput* adalah parameter yang di ukur menggunakan aplikasi siege. Pada pengujian menggunakan aplikasi siege berbeda dengan httpperf karena hasil *request* akan diketahui setelah pengujian, beban yang akan diberikan adalah 10 dan 100 *client* dengan durasi masing-masing pengujian selama 1 menit dan 5 menit. Setelah dilakukan pengujian maka hasilnya adalah *throughput* Litespeed lebih besar dibandingkan Nginx karena semakin besar nilai *throughput* maka akan semakin banyak pula data yang bisa ditampung oleh *web server*. Pada pengujian ini Litespeed lebih unggul diandingkan Nginx.

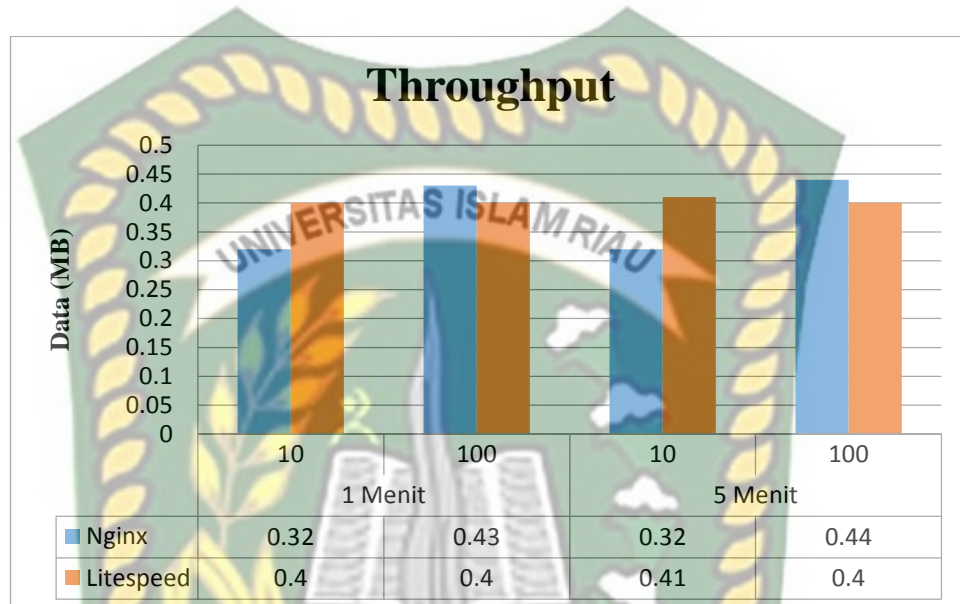
4.1.5 Pengujian terhadap halaman web wordpress

Pengujian terhadap halaman *web* menggunakan wordpress ini bertujuan untuk melihat kinerja *web server* terhadap halaman *web* yang kompleks. Halaman *web* wordpress sudah dirancang sehingga memiliki banyak konten. Berikut adalah tampilan dari halaman wordpress yang akan digunakan untuk pengujian dan hasil pengujiannya.



Gambar 4.29. Tampilan halaman web Wordpress

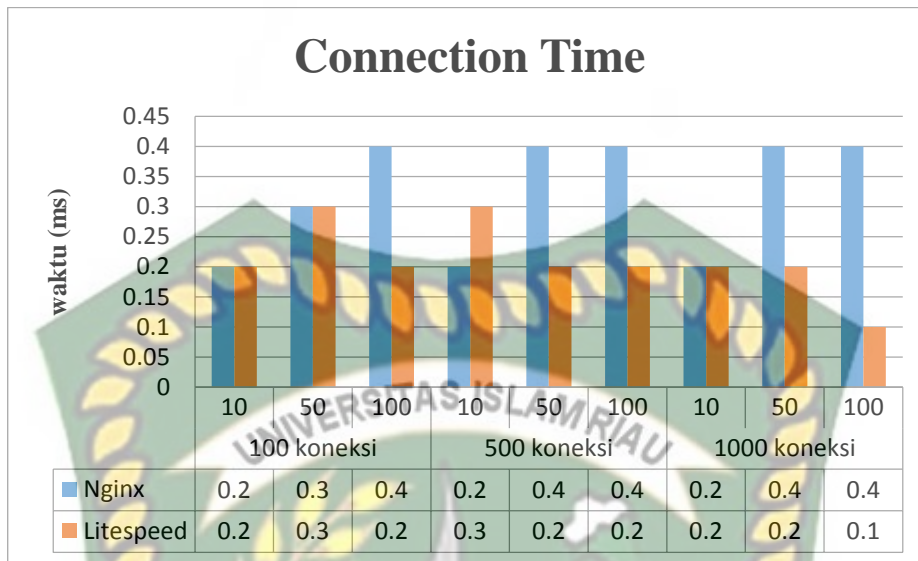
Berikut adalah hasil pengujian *throughput* menggunakan aplikasi Siege terhadap halaman web wordpress dengan memberikan beban sebanyak 1 dan 100 dan lama waktu pegujiannya adalah 1 dan 5 menit.



Gambar 4.30. Hasil *throughput* wordpress

Dari hasil perbandingan yang ditampilkan gambar 4.30 Nginx memiliki nilai *throughput* yang lebih besar saat di berikan beban 100 *client* dengan durasi waktu 1 menit dan 5 menit. Sedangkan Litespeed unggul terhadap Nginx pada saat diberikan beban 10 *client* dengan durasi waktu pengujian 1 menit dan 5 menit. Hasil pada pengujian ini adalah Nginx lebih unggul dibandingkan Litespeed, hal ini dilihat dari Nginx yang unggul saat diberikan beban 100 *client*, dimana *bandwith* yang dihasilkan jauh lebih baik sehingga mampu menampung banyak data. Namun perbedaan yang ditunjukkan diantara kedua *web server* tidak terlalu signifikan.

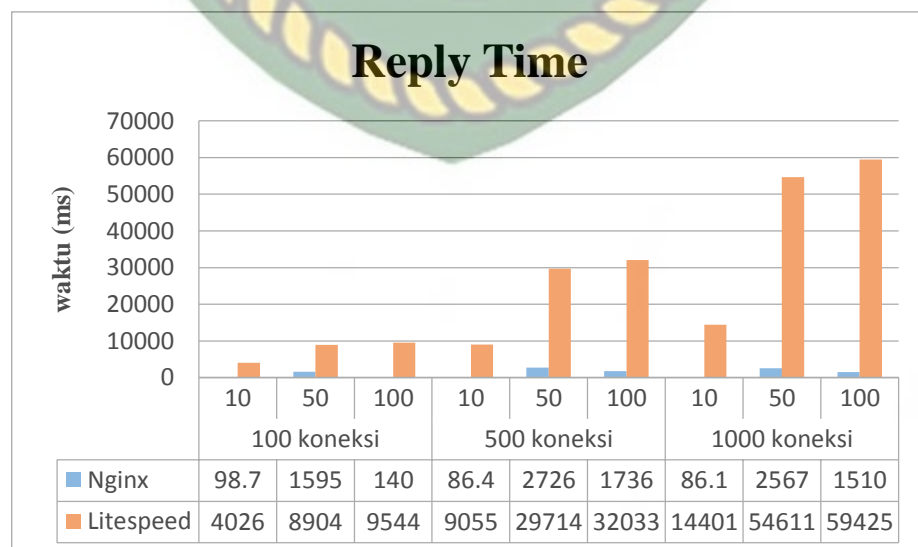
Connection Time



Gambar 4.31. Hasil *connection time* wordpress

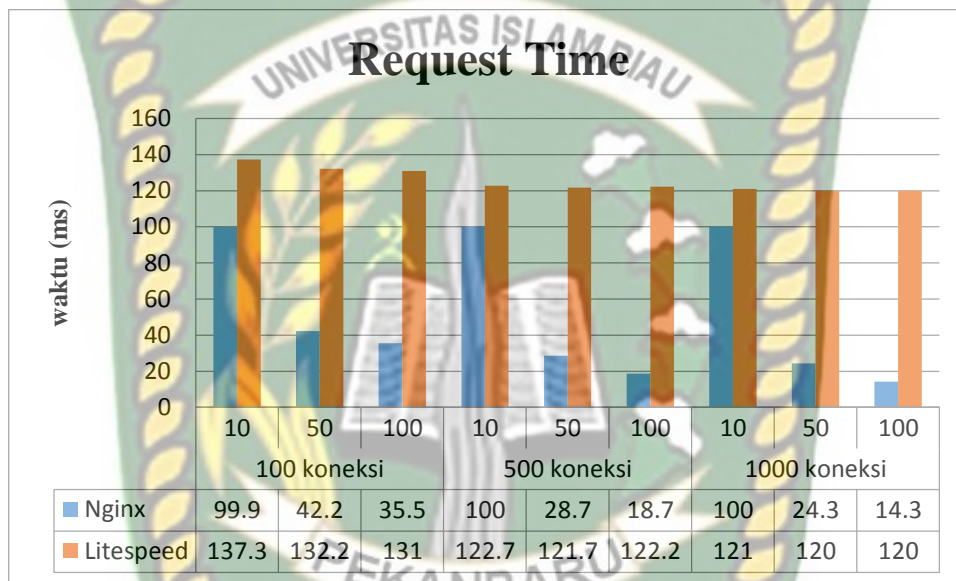
Dari hasil perbandingan yang ditampilkan gambar 4.31 Litespeed lebih unggul dibandingkan Nginx karena mempunyai nilai *connection time* yang lebih kecil. Namun pada saat diberikan beban 100 koneksi dengan 100 request maka yang unggul adalah Nginx. Secara keseluruhan Litespeed lebih unggul dimana semakin kecil *connection* yang ditunjukkan maka akan semakin bagus kinerja dari web server tersebut.

Reply Time



Gambar 4.32. Hasil *reply time* wordpress

Dari hasil perbandingan yang ditampilkan oleh gambar 4.32 terlihat bahwa nilai *reply* Nginx lebih kecil daripada Litespeed. Hal ini menunjukkan bahwa Nginx melayani permintaan lebih cepat daripada Litespeed, sehingga dalam hal ini menunjukkan bahwa Nginx lebih unggul.



Gambar 4.33. Hasil *request time* wordpress

Dari hasil perbandingan yang ditampilkan oleh gambar 4.33 terlihat bahwa nilai *request* Nginx lebih kecil daripada Litespeed. Hal ini menunjukkan bahwa Nginx melayani permintaan lebih banyak daripada Litespeed, sehingga dalam hal ini menunjukkan bahwa Nginx lebih unggul.

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
FD_SETSIZE
Maximum connect burst length: 1

Total: connections 100 requests 100 replies 100 test-duration 9.988 s

Connection rate: 10.0 conn/s (99.9 ms/conn, <=3 concurrent connections)
Connection time [ms]: min 85.1 avg 103.9 max 1468.8 median 89.5 stddev 137.9
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 10.0 req/s (99.9 ms/req)
Request size [B]: 81.0

Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 samples)
Reply time [ms]: response 98.7 transfer 5.0
Reply size [B]: header 345.0 content 51350.0 footer 2.0 (total 51697.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=100 5xx=0

CPU time [s]: user 9.90 system 9.07 (user 9.1% system 90.9% total 99.9%)
Net I/O: 500.2 KB/s (4.1*10^6 bps)

Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 flab-full 0 other 0
root@ronaldi-Lenovo-ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.34. Hasil *error* wordpress pada Nginx

```

root@ronaldi-Lenovo-Ideapad-320-14AST: /home/ronaldi
Maximum connect burst length: 1

Total: connections 1000 requests 1000 replies 1000 test-duration 121.032 s

Connection rate: 8.3 conn/s (121.0 ms/conn, <=210 concurrent connections)
Connection time [ms]: min 196.8 avg 14756.0 max 27073.0 median 15319.5 stddev 6386.0
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 8.3 req/s (121.0 ms/req)
Request size [B]: 75.0

Reply rate [replies/s]: min 6.0 avg 8.1 max 10.0 stddev 1.2 (24 samples)
Reply time [ms]: response 14401.5 transfer 354.3
Reply size [B]: header 330.0 content 51945.0 footer 2.0 (total 52277.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=1000 5xx=0

CPU time [s]: user 2.40 system 118.42 (user 2.0% system 97.8% total 99.8%)
Net I/O: 422.4 KB/s (3.5*10^6 bps)

Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 flab-full 0 other 0
root@ronaldi-Lenovo-ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.35. Hasil *error* wordpress pada Litespeed

Dari hasil perbandingan yang ditunjukkan gambar 4.34 dan 4.35 tidak terjadi *error* dalam proses pengujian pada kedua web server, hal ini menunjukkan bahwa Nginx dan Litespeed memiliki kinerja yang sama-sama baik karena tidak ada terjadi *error* pada saat pengujian.

Tabel 4.9 Hasil pengujian Wordpress

Beban		Nginx				Litespeed			
nums/conn	rate/seq	C (ms)	Req (ms)	Rep (ms)	Erorr	C (ms)	Req (ms)	Rep (ms)	Erorr
100	10	0.2	99.9	98.7	0	0.2	137.3	4026	0
	50	0.3	42.2	1595	0	0.3	132.2	8904	0
	100	0.4	35.5	140	0	0.2	131	9544	0
500	10	0.2	100	86.4	0	0.3	122.7	9055	0
	50	0.4	28.7	2726	0	0.2	121.7	29714	0
	100	0.4	18.7	1736	0	0.2	122.2	32033	0
1000	10	0.2	100	86.1	0	0.2	121	14401	0
	50	0.4	24.3	2567	0	0.2	120	54611	0
	100	0.4	14.3	1510	0	0.1	120	59425	0

Hasil yang ditampilkan oleh table 4.9 adalah rangkuman dari pengujian wordpress, di dalam tabel 4.9 *connection*, *request*, *reply*, dan *error* adalah parameter yang di ukur menggunakan aplikasi httpperf, nums/conn adalah banyak beban koneksi yang akan diberikan oleh httpperf pada saat pengujian dan rate/seq adalah banyak *request*/detik yang akan di kirimkan pada pengujian. Berikut adalah penjelasan dari hasil pengujian.

1. *Connection*, semakin kecil nilai pada pengujian atau mendekati angka 0 maka lebih baik pula kinerja web server, pada tabel 4.9 nilai *connection* Nginx dan Litespeed pada saat diberikan beban koneksi 100, 500, dan 1000 dengan *request*/detik 10, 50, dan 100 maka hasil yang di dapat bahwa nilai *connection* Litespeed lebih kecil dibandingkan Nginx. Hal ini menunjukkan bahwa Litespeed lebih unggul dari Nginx.
2. *Request*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik pula kinerja *web server* dalam menampung permintaan. Pada tabel 4.9 nilai *request* Nginx dan Litespeed pada saat diberikan

beban koneksi 100, 500, dan 1000 dengan *request*/detik 10, 50, dan 100 maka di dapatkan hasil bahwa Nginx jauh lebih unggul dibandingkan Litespeed, karena nilai *request* Nginx lebih kecil daripada Litespeed, sehingga mampu menampung banyak data.

3. *Reply*, semakin kecil nilai pada pengujian atau mendekati 0, maka semakin baik dan cepat pula kinerja *web server* dalam melayani permintaan. Pada tabel 4.9 nilai *reply* Nginx jauh lebih unggul dibandingkan Litespeed. Sehingga Nginx lebih cepat dalam melayani permintaan.
4. *Error*, pada tabel 4.9 menampilkan bahwa kedua *web server* tidak mengalami kesalahan pada saat proses pengujian, hal ini menunjukkan bahwa kinerja kedua *web server* sama baik. Apabila terjadi *error* pada salah satu *web server* maka kinerja *web server* tersebut kurang baik.

Tabel 4.10 Hasil *throughput* wordpress

Beban		Nginx	Litespeed
waktu/menit	client	T (mb)	T (mb)
1 Menit	10	0.32	0.4
	100	0.43	0.4
5 Menit	10	0.32	0.41
	100	0.44	0.4

Hasil yang ditampilkan oleh tabel 4.10 adalah hasil dari pengujian halaman bukalapak, di dalam tabel 4.10 *throughput* adalah parameter yang di ukur menggunakan aplikasi siege. Pada pengujian menggunakan aplikasi siege berbeda dengan httpperf karena hasil *request* akan diketahui setelah pengujian, beban yang


```

root@ubuntu:/var/log/nginx# nano access.log
GNU nano 2.5.3                                     File: access.log

192.168.100.9 - [29/Oct/2019:20:14:05 +0700] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:14:05 +0700] "GET /favicon.ico HTTP/1.1" 404 152 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:22:20 +0700] "GET /info.php HTTP/1.1" 200 23431 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:22:21 +0700] "GET /favicon.ico HTTP/1.1" 404 152 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:47 +0700] "GET / HTTP/1.1" 302 5 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/install.php HTTP/1.1" 200 4299 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-includes/css/dashicons.min.css?ver=5.2.4 HTTP/1.1" 200 47558 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-includes/css/buttons.min.css?ver=5.2.4 HTTP/1.1" 200 6312 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/css/forms.min.css?ver=5.2.4 HTTP/1.1" 200 22174 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/css/l10n.min.css?ver=5.2.4 HTTP/1.1" 200 2479 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/css/install.min.css?ver=5.2.4 HTTP/1.1" 200 4930 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-includes/js/jquery/jquery.min.js?ver=3.1.1 HTTP/1.1" 200 10056 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/js/language-chooser.min.js?ver=5.2.4 HTTP/1.1" 200 374 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-includes/js/jquery/jquery.js?ver=1.12.4 HTTP/1.1" 200 96373 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/images/wordpress-logo.svg?ver=20131107 HTTP/1.1" 200 1521 "http://192.168.100.10/wp-admin/css/install.min.css" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:52:50 +0700] "GET /wp-admin/images/spinner.gif HTTP/1.1" 200 3656 "http://192.168.100.10/wp-admin/css/install.min.css?ver=5.2.4" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "POST /wp-admin/install.php?step=1 HTTP/1.1" 200 2363 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "GET /wp-includes/js/xcvbn-synchronous.min.js?ver=1.0 HTTP/1.1" 200 324 "http://192.168.100.10/wp-admin/install.php?step=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "GET /wp-admin/js/password-strength-meter.min.js?ver=5.2.4 HTTP/1.1" 200 769 "http://192.168.100.10/wp-admin/install.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "GET /wp-includes/js/underscore.min.js?ver=1.8.3 HTTP/1.1" 200 16184 "http://192.168.100.10/wp-admin/install.php?step=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "GET /wp-includes/js/wp-util.min.js?ver=5.2.4 HTTP/1.1" 200 1646 "http://192.168.100.10/wp-admin/install.php?step=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "GET /wp-admin/js/user-profile.min.js?ver=5.2.4 HTTP/1.1" 200 6321 "http://192.168.100.10/wp-admin/install.php?step=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:53:08 +0700] "GET /wp-includes/js/xcvbn.min.js HTTP/1.1" 200 822202 "http://192.168.100.10/wp-admin/install.php?step=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.10 - [29/Oct/2019:20:54:38 +0700] "POST /wp-cron.php?doing_wp_cron=1572357278.7860000133514404296875 HTTP/1.1" 200 31 "http://192.168.100.10/wp-cron.php?step=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.10 - [29/Oct/2019:20:54:39 +0700] "GET /2019/10/29/hello-world/ HTTP/1.1" 200 5777 "http://192.168.100.10/2019/10/29/hello-world/" "WordPress/5.2.4; https://192.168.100.9"
192.168.100.9 - [29/Oct/2019:20:54:39 +0700] "POST /wp-admin/install.php?step=2 HTTP/1.1" 200 1157 "http://192.168.100.10/wp-admin/install.php?step=2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:44 +0700] "GET /wp-login.php HTTP/1.1" 200 1937 "http://192.168.100.10/wp-admin/install.php?step=2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:44 +0700] "GET /wp-includes/css/dashicons.min.css?ver=5.2.4 HTTP/1.1" 200 44937 "http://192.168.100.10/wp-login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:44 +0700] "GET /wp-admin/css/login.min.css?ver=5.2.4 HTTP/1.1" 200 3964 "http://192.168.100.10/wp-login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:49 +0700] "POST /wp-login.php HTTP/1.1" 302 5 "-" "http://192.168.100.10/wp-login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-admin/ HTTP/1.1" 200 17220 "http://192.168.100.10/wp-login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/js/thickbox/thickbox.css?ver=5.2.4 HTTP/1.1" 200 2658 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/css/editor.min.css?ver=5.2.4 HTTP/1.1" 200 27203 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/js/dist/hooks.min.js?ver=5.2.4 HTTP/1.1" 200 4905 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/js/heartbeat.min.js?ver=5.2.4 HTTP/1.1" 200 6017 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/js/dist/vendor/wp-polyfill.min.js?ver=7.0.0 HTTP/1.1" 200 96193 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/js/wp-auth-check.min.js?ver=5.2.4 HTTP/1.1" 200 4783 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-admin/load-scripts.php?c=0&load=5B45D=jquery-core,jquery-migrate,utils&ver=5.2.4 HTTP/1.1" 200 108861 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"
192.168.100.9 - [29/Oct/2019:20:54:52 +0700] "GET /wp-includes/js/wplink.min.js?ver=5.2.4 HTTP/1.1" 200 11262 "http://192.168.100.10/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0"

```

Gambar 4.37. Hasil log web server Nginx

Dari hasil perbandingan dari gambar 4.36 dan 4.37 maka bisa dilihat bahwa log Nginx lebih lengkap dibandingkan dengan Litespeed, pada Litespeed tidak terdapat keterangan apa saja yang sudah di akses oleh user atau pun yang telah dilakukan oleh penyedia layanan server, berbeda dengan yang ada pada log Nginx terdapat informasi lengkap dengan apa saja yang telah di akses oleh user maupun yang dilakukan penyedia layanan server. Hal ini menunjukkan bahwa system log pada Nginx lebih baik dan lengkap dibandingkan Litespeed.

4.3 Pengujian security web server Nginx dan Litespeed

Pada pengujian security kedua web server ini akan menggunakan aplikasi Nikto, aplikasi ini akan mencoba menembus system keamanan yang ada di Nginx dan Litespeed dengan cara mengscan semua file yang ada pada web server dan

akan menampilkan port dari web server tersebut. Berikut adalah hasil dari *scanning* nikto.

```

root@ronaldi-Lenovo-Ideapad-320-14AST:/home/ronaldi# nikto -h 192.168.137.10
- Nikto v2.1.5
-----
+ Target IP:      192.168.137.10
+ Target Hostname: 192.168.137.10
+ Target Port:    80
+ Start Time:    2019-11-16 00:21:46 (GMT7)
-----
+ Server: nginx/1.10.3 (Ubuntu)
+ RFC-1918 IP address found in the 'link' header. The IP is "192.168.100.12".
+ The anti-clickjacking X-Frame-Options header is not present.
+ Uncommon header 'link' found, with contents: <http://192.168.100.12/wp-json/?_rel="https://api.w.org/"
+ Uncommon header 'x-redirect-by' found, with contents: WordPress
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server leaks inodes via ETags, header found with file /testing/, fields: 0x5dc194e5 0xe5
+ OSVDB-3092: /testing/: This might be interesting...
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-5292: /info.php?file=http://cirt.net/rflinc.txt?: RFI from ASnake's list (http://h3.ckers.org/weird/rfi_locations.dat) or from http://0
svdb.org/
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested up to' version usually matches the WordPress version
+ OSVDB-3092: /license.txt: License file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ Uncommon header 'x-frame-options' found, with contents: SAMEORIGIN
+ Cookie wordpress_test_cookie created without the httponly flag
+ 6544 items checked: 6 error(s) and 13 item(s) reported on remote host
+ End Time:      2019-11-16 00:27:06 (GMT7) (320 seconds)
-----
+ 1 host(s) tested
root@ronaldi-Lenovo-Ideapad-320-14AST:/home/ronaldi#

```

Gambar 4.38. Hasil *scanning* web server Nginx

```

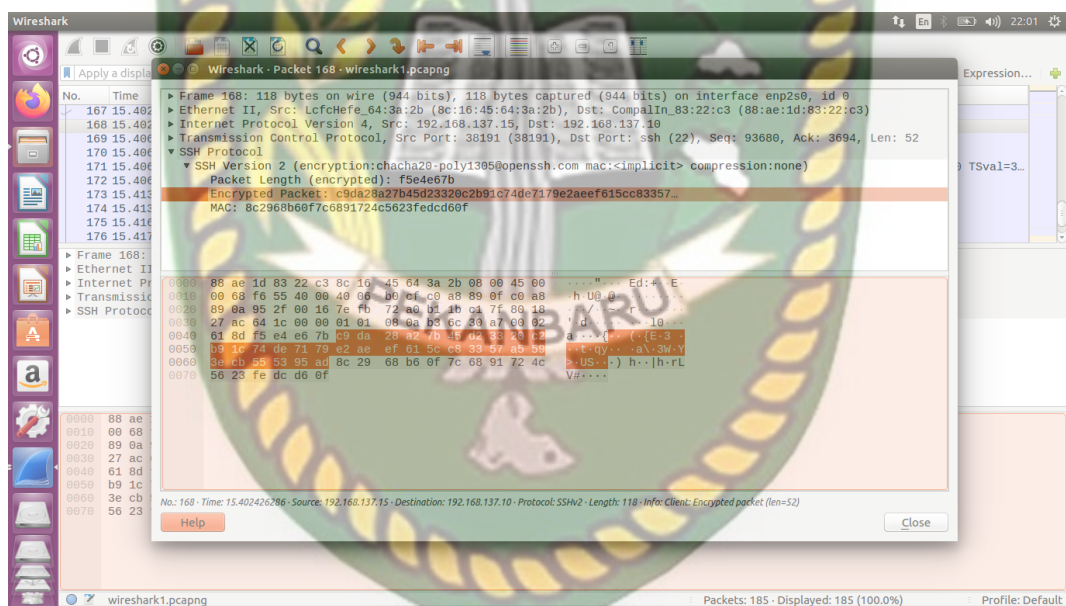
root@ronaldi-Lenovo-Ideapad-320-14AST:/home/ronaldi# nikto -h 192.168.137.100
- Nikto v2.1.5
-----
+ Target IP:      192.168.137.100
+ Target Hostname: 192.168.137.100
+ Target Port:    80
+ Start Time:    2019-11-16 00:18:05 (GMT7)
-----
+ Server: LiteSpeed
+ Server leaks inodes via ETags, header found with file /, inode: 13538, size: 1
573650644, mtime: 0x540936
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-3233: /phpinfo.php: Contains PHP configuration information
+ OSVDB-3092: /testing/: This might be interesting...
+ 6544 items checked: 6 error(s) and 4 item(s) reported on remote host
+ End Time:      2019-11-16 00:20:18 (GMT7) (133 seconds)
-----
+ 1 host(s) tested
root@ronaldi-Lenovo-Ideapad-320-14AST:/home/ronaldi#

```

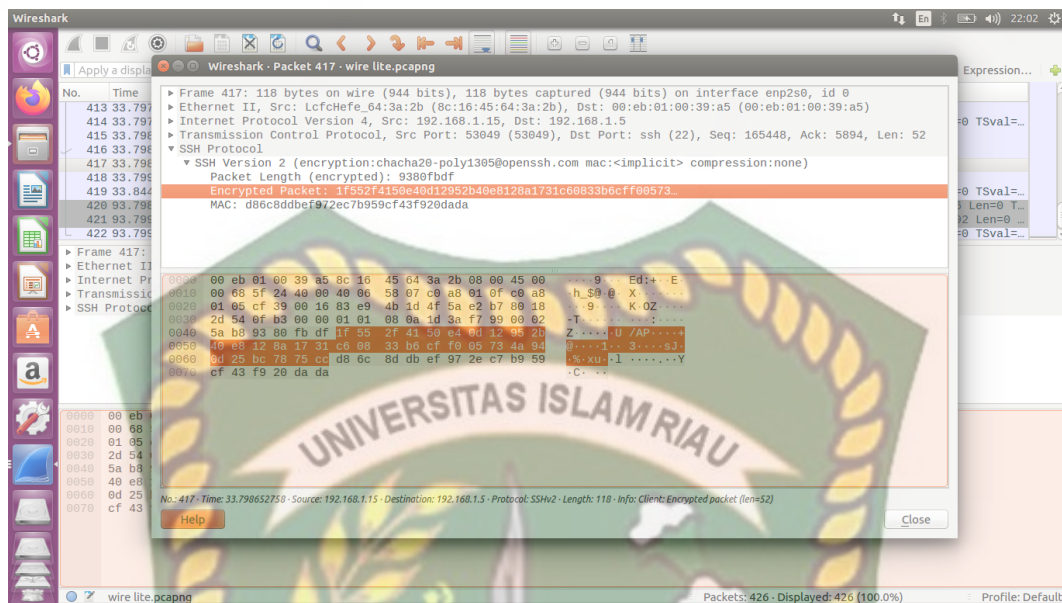
Gambar 4.39. Hasil *scanning* web server Litespeed

Dari hasil *scanning* yang ditampilkan oleh gambar 4.38 dan 4.39 terlihat bahwa sistem keamanan Litespeed lebih unggul dibandingkan Nginx, karena pada Nginx semua file yang ada bisa diketahui oleh nikto dan sistem operasi yang

digunakan juga bisa diketahui serta tidak ada error pada saat proses *scanning* yang berarti tidak ada sistem keamanan yang berusaha memblokir penetrasi yang dilakukan oleh nikto, berbeda dengan Litespeed file yang ada di dalamnya tidak semua diketahui oleh nikto contohnya wordpress dan sistem operasi yang digunakan oleh Litespeed tidak dapat terdeteksi oleh nikto serta terjadinya error pada saat proses *scanning*. Hal ini menunjukkan bahwa dalam hal system keamanan Litespeed lebih unggul dibandingkan Nginx.



Gambar 4.40. Hasil *capture packet* Nginx



Gambar 4.41. Hasil *capture packet* Litespeed

Pengujian ini bertujuan untuk melihat keamanan saat *client* mengupload data ke server atau saat server merespon permintaan dari *client*. Pada gambar 4.40 dan 4.41 menampilkan hasil *capture packet* menggunakan aplikasi wireshark, hasil yang di dapat bahwa Nginx dan Litespeed akan mengenkripsi setiap data yang masuk maupun data yang keluar dari server, hal ini menunjukkan kedua web server sama baiknya pada saat proses transfer atau transaksi data karena bisa melakukan proses enkripsi dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisa dan kesimpulan dari pengujian terhadap kinerja web server Nginx dan Litespeed:

1. Dari perbandingan pengujian throughput antara *web Server* Nginx dan Litespeed sebanyak lima kali pengujian didapatkan hasil dimana hasil dari *throughput* Nginx memiliki kinerja yang lebih unggul dari pada kinerja throughput Litespeed, dimana bandwidth yang dihasilkan jauh lebih baik sehingga mampu menampung banyak data.
2. Pengujian *Connection Section (Time)* yang dilakukan antara *web Server* Nginx dan Litespeed didapatkan hasil Litespeed lebih unggul dalam mengkoneksikan *Client – Server* dan *Server* ke *Client* dimana semakin kecil *connection* yang ditunjukkan semakin bagus kinerja dari suatu *Server*.
3. Perbandingan dari *Request* antara *web Server* Nginx dan Litespeed memiliki kinerja yang hampir sama tergantung dari kemampuan *Client* yang mengakses *Web Server* tersebut.
4. Perbandingan pengujian *reply section* antara *web Server* Nginx dan Litespeed, kinerja Litespeed lebih baik dari Nginx, karena *reply time*

dari *web server* Litespeed lebih kecil, semakin kecil waktu *reply time* semakin cepat data yang bisa ditransfer

5. Pada pengujian *error section* terhadap kinerja *web Server* Nginx dan Litespeed memiliki kinerja yang sama-sama baik karena tidak terjadinya *error* selama pengujian dari kedua *web Server* Nginx dan Litespeed.
6. Pada pengujian *system log* terhadap *web server* Nginx dan Litespeed, hasil yang di dapat adalah *log* dari Nginx lebih unggul karena menampilkan informasi lebih lengkap dibandingkan Litespeed.
7. Pada pengujian *security* dari *web server* Nginx dan Litespeed, hasilnya adalah Litespeed lebih tahan tahan terhadap penyusupan atau pun peretasan dibandingkan Nginx.

5.2 Saran

Berdasarkan hasil penelitian dan pengujian maka dapat dikembangkan lagi untuk pengujian *web server* dengan menggunakan *software* lain dan parameter yang digunakan hasil pengujian akan lebih baik.

DAFTAR PUSTAKA

- Adnan, Fariq., 2016, *Analisis Perbandingan Performa Web Server Apache Dan Nginx Menggunakan Httpperf Pada VPS Dengan Sistem Operasi Centos*, Skripsi, STMIK AMIKOM Yogyakarta, Yogyakarta.
- Amri, Ahmad Ariful., 2015, *Analisis Kinerja Web Server Apache dan Nginx Menggunakan Siege dan Httpperf*, Skripsi, Universitas Syiah Kuala Darussalam, Banda Aceh.
- Andica, Intan Yuli., 2017, *Peforma Kinerja Web Server Berbasis Ubuntu Linux Dan Turnkey Linux*, Jurnal Penelitian Ilmu komputer, Sistem Embedded Dan Logic, Vol.5.
- Dawood, Rahmad., dkk., 2014, *Kelayakan Raspberry sebagai Web Server: Perbandingan Kinerja Nginx, Apache, dan Lighttpd pada Platform Raspberry Pi*, Jurnal Rekayasa Elektrika, Vol.11.
- Hasibuan, Zainal A. 2007, *Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi*, Universitas Indonesia, Depok.
- Hasim, Ma'fud., Riadi, Imam., 2013, *Analisis Perbandingan Kinerja Server Pada Data Center IIX Dan International Menggunakan Teknologi Cloud Server*, Jurnal Sarjana Teknik Informatika, Vol.1.
- Irza, Intan Ferina., dkk., 2017, *Analisis Perbandingan Kinerja Web Server Apache Dan Nginx Menggunakan Httpperf Pada Portal Berita*, Jurnal Vokasional Teknik Elektronika Dan Informatika, Vol.5.

Is, Mardianto., Adriant, M. Ferdy, 2015, *Implementasi Wireshark Untuk Penyadapan(Sniffing) Paket Data Jaringan*, Jurnal Seminar Nasional Cendekiawan, Vol.1.

Juliharta, I Gede Putu Krisna, 2015, *Distribusi Konten Web Server Menggunakan Metode Content Delivery Network*, Jurnal Sistem Dan Informatika, Vol.0.

Wijaya, Ardi., Harjuna, Army Martia, 2017, *Perancangan Program Aplikasi Tabungan Menggunakan Server Nginx*, Jurnal Pseudocode, Vol.4.

Yusuf, Effendi, 2013, *Implementasi Teknologi Load Balancer Dengan Web Server Nginx Untuk Mengatasi Beban Server*, STMIK AMIKOM Yogyakarta, Yogyakarta.

