

Analisis Web Server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web Server Apache dengan Nginx

Abdul Aziz, Topan Tampati
Jurusan Teknik Informatika dan Komputer
Politeknik Negeri Jakarta
Jakarta, Indonesia
abdul.aziz@tik.pnj.ac.id, topan.tampati@gmail.com

Diterima: 12 September 2015. Disetujui: 10 Oktober 2015. Dipublikasikan: November 2015

Abstrak - Proses pertukaran data dan informasi masih terhambat dikarenakan belum terpusatnya sumber informasi dan data yang dibutuhkan Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta. Oleh karena itu dibutuhkan sebuah komputer server sebagai wadah dari web server yang dapat dimanfaatkan bagi pengguna individu maupun pengguna kelompok. Dengan perkembangan internet dan pengguna website semakin besar serta salah satu alasan memberikan cara yang mudah dan cepat dalam mengirimkan data dan informasi. Web server memenuhi kebutuhan untuk menampung website, email, dan lain-lain, serta untuk mempercepat dan mengorganisasi informasi dan data secara terpusat yang kemudian diimplementasikan pada jaringan enterprise Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta. Kinerja dari web server Apache dan Nginx untuk mengetahui manakah yang lebih unggul dari kedua web server tersebut. Metode yang digunakan adalah studi literatur mengenai web server, diskusi dan pengamatan upaya untuk menentukan software yang paling tepat digunakan untuk web server, dan pengujian. Dari tahapan yang dilakukan dapat disimpulkan bahwa web server dengan menggunakan Apache lebih unggul dibandingkan dengan web server yang menggunakan Nginx, kinerja Apache dalam transfer data, Connection, dan request data lebih baik sehingga memudahkan client dalam mendapatkan informasi yang dibutuhkan. Dengan kelebihan kecepatan transfer rata-rata 701 Kbytes/sec, web server Apache dapat diimplementasikan di jaringan enterprise Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta.

Kata Kunci: data, informasi, jaringan enterprise, web server, apache, nginx, transfer data, connection, request data.

I. PENDAHULUAN

Seorang pengguna pada suatu institusi melakukan penyimpanan data selalu pada

komputernya masing-masing atau sistem penyimpanan secara personal. Sehingga saat ingin melakukan pertukaran data diperlukan komputer atau perangkat memori penyimpanan data yang terpisah, dan dengan demikian pengguna tersebut adalah seorang yang memiliki akses pada komputer dan memori terpisah agar dapat membuka dan mengakses. Tentunya ini akan membuat pihak yang terkait harus memberikan tenaga dan waktu yang lebih saat akan melakukan pertukaran data. Dan bagaimana jika kejadian tersebut dilakukan oleh beberapa orang pada suatu institusi. Insiden tersebut merupakan gambaran suatu kebutuhan akan sebuah komputer server yang dapat menunjang dan mendukung kinerja dari sistem pertukaran data dan informasi, antara lain dengan menggunakan sistem hosting server yang di dalamnya mencakup database server yang dapat menampung semua file yang diperlukan oleh institusi untuk bertukar data dan informasi.

Hosting server adalah salah satu solusi untuk permasalahan tersebut, karena pada hosting control panel memungkinkan untuk mengelola beberapa server seperti web server dan mail server serta beberapa fitur tambahan seperti DNS dan file transfer. Hosting server dapat mempermudah dalam mengolah penggunaan server seperti penambahan user untuk mail server. Pada hosting sever ini akan mempermudah pengguna mendapatkan informasi yang dibutuhkan, salah satunya melalui website. Dimana website digunakan banyak orang untuk memuat informasi yang dibutuhkan, sehingga mempermudah dalam pertukaran informasi. Teknologi website tidak terlepas dengan data penyimpanan, atau wadah dimana website tersebut disimpan. Website tersimpan dalam web server yang menunjang dan mendukung kinerja dari website [1].

Perangkat lunak web server dikenal dapat melayani permintaan pengguna berupa http dari client yang terhubung dalam jaringan dan

memberikan pelayanan kepada yang meminta informasi berkaitan dengan *website* dan memberikan suatu hasil berupa halaman *web* yang ditampilkan dalam *browser*. *Web server* menggunakan port 80, *web server* sendiri terdiri dari dua komponen, yang pertama adalah komputer dan software *web server* yang digunakan [2]. Dimana pada *web server* inilah *website* yang digunakan untuk memberikan informasi atau bertukar informasi ditempatkan. Aplikasi *web server* dapat diperoleh dengan mudah baik yang berbayar maupun tanpa bayar. Saat akan memilih perangkat lunak aplikasi *web server*, administrator *web* harus memilih *web server* manakah yang akan digunakan untuk melayani para pengguna *website* institusinya, administrator melakukan ini ketika akan melakukan *hosting server* untuk digunakan *website* institusi.

Berkaitan dengan pelayanan *server*, membutuhkan aplikasi pada sistem komputer (*computer server dedicated*) yang berfungsi melayani permintaan akses dari komputer pengguna. Beberapa aplikasi pelayanan *server* antara lain *Web Server*, *FTP Server*, *DHCP Server*, *Mail Server*, *DNS Server*, *FTP Server*, dan *Database Server* [3]. Bila *web server* dan *website*-nya yang berisi tampilan informasi-informasi dapat diakses menggunakan *web browser* seperti Mozilla Firefox atau Google Chrome.

Berikut merupakan jenis *web server* antara lain: Apache *Web Server*, Apache Tomcat, MS Windows *server* 2003 *Internet Information Service* (IIS), *Light HTTP*, Sun Java *System Web Server*, Zerus *Web Server* serta Nginx. Studi implementasi *hosting server*, khusus fokus pada bagian implementasi *web server* akan bertemu pada masalah bagaimana kinerja *web server* itu sendiri. Dan studi kinerja dari *web server* pada permasalahan menganalisis kinerja *web server* yang akan diimplementasi. Studi *web server* akan bertemu dengan berbagai aplikasi perangkat lunak *web server* seperti tersebut di atas. Analisis mengangkat perbandingan *web server* Apache dengan *web server* Nginx.

Beberapa hal dasar yang perlu diketahui berkaitan dengan perbandingan aplikasi *web server* yang dianalisis. *Hosting Server*, Apache *HTTP server*, *web server* Nginx, dan hasil studi perbandingan yang telah dilakukan beberapa peneliti lainnya, studi perbandingan kinerja *web server* menjadi menarik. Kontribusi analisis *web server* untuk pengembangan *hosting server* institusi, khusus fokus pada bagian implementasi *web server* yang bertemu pada masalah bagaimana kinerja *web server*, berujung pada perbandingan kinerja *web server*: 1) Memberikan alternatif pemilihan *web server* yang dapat melayani kecepatan transfer data, waktu *request*, dan koneksi, 2) Memberikan solusi bagi administrator untuk menentukan topologi

jaringan. Dan terakhir, gambaran dan penyesuaian penggunaan *web server* dilihat dari kelebihan dan kekurangannya akan diperoleh administrator.

Hosting server merupakan komputer khusus yang terhubung dengan internet secara *real time*, dan secara terus-menerus agar pengguna internet dapat mengakses perangkat lunak yang menjadi *backbone* dari *World Wide Web* dikenal dengan *web server* merupakan perangkat lunak *server* [3]. *Web server*, merupakan perangkat lunak untuk berkomunikasi dengan *client* (*web browser*) dan mempunyai protokol sendiri yaitu *Hyper Text Transfer Protocol* [4]. Dengan protokol ini, komunikasi antar *web server* dengan *client*-nya (*browser*) dapat saling dimengerti dan lebih mudah. Proses yang dimulai dari permintaan *client* (*browser*), diterima *web server*, diproses, dan dikembalikan hasil prosesnya oleh *web server* ke *web client* lagi dilakukan secara transparan [5]. HTTPS memiliki pengertian yang sama dengan HTTP hanya saja HTTPS memiliki fungsi di bidang keamanan (*secure*). HTTPS menggunakan *Secure Socket Layer* (SSL) atau *Transport Layer Security* (TLS) sebagai *sub layer* dibawah HTTP aplikasi *layer* [6].

HTTP di-enkripsi dan deskripsi dari halaman yang diminta oleh pengguna dan halaman yang dikembalikan oleh *web server*. Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eaves droppers* dan *man in the middle attacks*. Pada umumnya *port* yang digunakan HTTPS adalah *port* 443. Tingkat keamanan tergantung pada ketepatan dan mengimplementasikan pada *browser* dan perangkat lunak *server* dan didukung oleh algoritma penyandian yang aktual. Oleh karena itu, pada halaman *web* digunakan HTTPS dan URL yang digunakan dimulai dengan *https://* [7].

Apache *HTTP server* adalah perangkat lunak dengan *platform oprating system* (OS) yang mendukung *multi-tasking*, dan menyediakan layanan untuk aplikasi lain yang terhubung ke dalamnya, seperti *web browser*. Apache pertama kali dikembangkan untuk bekerja dengan sistem operasi Linux/Unix, tetapi kemudian diadaptasi untuk bekerja di bawah sistem lain, termasuk Windows dan Mac [4].

Nginx adalah *software open-source* yang memiliki kinerja tinggi sebagai *server* HTTP dan *reverse proxy*. Nginx dengan cepat memberikan konten statis dengan penggunaan efisien sumber daya sistem. Hal ini dapat menyebarkan dinamis HTTP konten di jaringan menggunakan *FastCGI handler* untuk *script*, dan dapat berfungsi sebagai perangkat lunak yang sangat mampu menyeimbangkan beban. Nginx dibangun secara modular dan dengan demikian mampu mendukung berbagai fitur seperti *Load Balancing* dan *Reverse Proxying*, *Virtual hosts*

berbasis nama dan IP, Fast CGI, akses langsung ke *cache*, SSL, Flash Video Streaming dan sejumlah fitur-fitur standar lainnya. Nginx dapat dijalankan dan tersedia untuk platform Unix, Linux, varian dari BSD, MacOS X, Solaris, dan Microsoft Windows [8].

Nginx menjadi pilihan berkat kinerjanya yang tinggi, stabilitas, kekayaan akan fitur, mudah dikonfigurasi dan terutama hemat sumber daya [9]. Berbeda dengan *server* tradisional, Nginx tidak sepenuhnya bergantung pada 'thread' 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 dalam jumlah yang bisa diprediksikan.

Awalnya Nginx dibangun di Rusia untuk memenuhi kebutuhan mesin pencari skala besar Rambler yang tetap memanfaatkannya sampai sekarang. Berkat berbagai kemampuan yang dimilikinya, termasuk kinerja yang tinggi dan fleksibilitas dalam konfigurasi, Nginx banyak digunakan untuk mendukung layanan *web* skala besar antara lain WordPress.com, SourceForge, Hulu, ComputerBase [8];[10]. Nginx menggunakan pendekatan *asynchronous-event* untuk menangani permintaan yang diprediksi memberikan kinerja yang lebih bawah beban, kontras dengan Apache HTTP *server* [11];[8]. Fitur yang lain: *Handling of static files, index files, and auto-indexing, Reverse proxy with caching, Load balancing, Fault tolerance, Web page access authentication, gzip compression, Ability to handle more than 10000 simultaneous connections, URL rewriting, Custom logging, Server-side includes*. Fitur Mail proxy: SMTP, POP3, and IMAP proxy, STARTTLS dan STLS support, SSL http [8].

DNS (*Domain Name System*), sebuah sistem yang menyimpan informasi tentang nama *host* maupun nama *domain* dalam bentuk basis data tersebar (*distributed database*) di dalam jaringan komputer, misalkan internet. Fungsi utama dari sebuah sistem DNS adalah menerjemahkan nama-nama *host* (*hostnames*) menjadi nomor IP (*IP address*) ataupun sebaliknya, sehingga nama tersebut mudah diingat oleh pengguna internet [5]. Fungsi lainnya adalah untuk memberikan suatu informasi tentang suatu *host* ke seluruh jaringan internet.

Beberapa hasil penelitian berkaitan web server Apache dan Nginx melaporkan, bahwa Nginx + php-fpm jauh lebih baik daripada Apache + mod_php5. "Jauh lebih baik" didefinisikan sebagai hasil akhir di mana Nginx bisa melayani jumlah http request jauh lebih banyak daripada Apache. Kemudian penelitian lain bahwa php fpm yang *listen*

di tcp *socket* performanya kerjanya lebih buruk daripada apabila dijalankan di Unix *socket* [11]. Percobaan tersebut dibangun menggunakan PHP dengan *database* PostgreSQL. Kerja script mengambil beberapa ratus *record* dari *database*, kemudian mengirimkan hasil html ke *browser*. Lokasi *database* terpisah dari *server web*. *Server web* yang digunakan punya RAM 4GB, *harddisk* SAS 15k 136GB, dan 8 *processor cores*. Php-fpm berjalan menggunakan Unix *socket*.

Konfigurasi Apache dan Nginx standar, hanya mengubah *timeout* untuk *keep alive* menjadi 3 detik. OS *server* yang digunakan adalah Ubuntu 12.04 tanpa *custom* repo. Tes beban dilakukan dengan menggunakan Siege atau Apache *benchmark* (ab), prinsipnya sama saja. Hasil analisis Apache vs Nginx adalah perbedaan konsumsi RAM. Pada percobaan (-c 150), Apache bisa memakai memory RAM hingga 715MB, sedangkan menggunakan RAM hingga 575MB. Selisih sekitar 24%. Untuk *web* yang ditempatkan di VPS dengan RAM kecil boleh jadi ini ada bedanya. Perbedaan lain adalah kecepatan turunnya penggunaan RAM setelah tes. Tampaknya Nginx melepas RAM lebih cepat daripada apache. Dan menjelaskan Nginx tidak memakan RAM sebanyak Apache; Performa keduanya dianggap sama, beda pun sangat tipis; Pada jumlah *concurrency* tinggi (-c 350), php-fpm dengan tcp *socket* bisa lebih stabil daripada menggunakan Unix *socket* [11].

Hasil penelitian yang telah dipaparkan di atas menunjukkan hasil perbandingan *web server* Apache dengan Nginx pada pengukuran tes kinerja menggunakan PHP dengan *database* PostgreSQL. Kerja script mengambil beberapa ratus *record* dari *database*, kemudian mengirimkan hasil HTML ke *browser*, dimana lokasi *database* terpisah dari *server web*.

II. METODE PENELITIAN PENGEMBANGAN HOSTING SERVER

Pada metode penelitian kami dengan menggunakan dua *web server* yang sama yaitu Apache dan Nginx, kedua *web server* ini ditujukan untuk wadah dari http yang cenderung kepada *website* dan *email*. Pengujian yang dilakukan adalah menguji performa kecepatan kedua *web server* untuk mengetahui kinerja masing-masing. Pengujian ini lebih kepada masalah cepat waktu yang digunakan untuk mengakses sebuah *web*. Manakah *web server* yang lebih efektif dalam meng-handle setiap aktifitas yang dilakukan oleh *user* secara bersamaan. Kemudian dijabarkan bagaimana melakukan pengujian ini, setelah itu berlanjut pada hasil pengujian. Dan melakukan analisis kinerja *web server* Apache dan Nginx.

A. Kebutuhan Perangkat Keras dan Perangkat Lunak

Analisis *hosting server* sistem operasi yang digunakan Linux, Ubuntu. Ubuntu termasuk sistem operasi *open source* dan *free software* dan sistem operasi *server* yang stabil. Perangkat lunak *hosting server* yang akan diimplementasikan pada institusi perangkat lunak adalah *web server*, *database server*, *DNS server*, *email server*, *file transfer*. Server basis data yang akan dibangun pada *hosting server* infrastruktur institusi adalah *server* basis data MySQL *server* dan PostgreSQL *server*. Sebagai perangkat lunak untuk analisis pengembangan *web server* basis data, MySQL *server* dan PostgreSQL *server* memiliki kelebihan dan kekurangan masing-masing. Pada bagian lain penelitian pengembangan *web server* institusi, kedua perangkat lunak ini diperbandingkan untuk mengetahui perangkat lunak sistem *database server* mana yang lebih tangguh dan lebih baik.

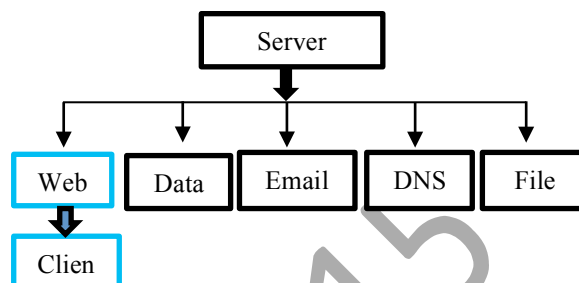
Selanjutnya perangkat lunak DNS *server* yang memiliki fungsi khusus yang digunakan sebagai *master DNS server* sehingga dalam proses konfigurasi dan penggunaannya lebih mudah. Dan mendukung penyimpanan data DNS dengan menggunakan basis data sehingga memudahkan dalam pengelolaan nama *domain* dan replikasi data DNS, spesifikasi tersebut antara lain dimiliki oleh Power DNS. Sedangkan untuk *slave DNS*, menggunakan *Bind* atau bisa disebut dengan penyimpanan kedua (*secondary*) yang mana data yang ada dalam Power DNS langsung tersinkronkan pada *Bind* sehingga ketika salah satu dari *software* DNS mati maka *software* yang lainnya yang akan membackupnya.

Email server menggunakan *webmail roundcube* dengan protokol IMAP dan Microsoft Outlook dengan protokol POP3. IMAP memungkinkan pengguna memilih pesan yang akan diambil, membuat *folder* di *server*, mencari pesan tertentu, maupun menghapus pesan yang ada dan terjadi komunikasi dua arah, sehingga terjadi sinkronisasi data. Sedangkan POP3 yang hanya memperbolehkan kita mengambil atau *download* semua pesan yang ada tanpa kecuali dan hanya bekerja dengan satu arah saja.

File transfer dibutuhkan *software* protokol SFTP dan FTP. SFTP dan FTP adalah suatu protokol dengan persamaan dapat tukar-menukar *file* dalam suatu *network* yang men-support TCP/IP. Terdapat dua hal penting agar terjalannya komunikasi data yaitu adanya sisi *server* dan *client*. Fungsi utama aplikasi ini adalah untuk mengakses mesin secara *remote*. Bentuk akses *remote* yang bisa diperoleh adalah akses pada mode teks maupun mode grafis/X apabila konfigurasinya mengizinkan.

B. Rencana Pelaksanaan Pengujian dan Analisis

Gambar 1 adalah diagram blok kerja alat, merupakan dinamika kerja sistem perangkat lunak pada *hosting server*, merupakan blok diagram pelaksanaan pengujian.



Gambar 1. Diagram Blok Hosting Server

Blok diagram di atas percobaan *web server* yang diimplementasikan pada *hosting server* menggunakan program dengan *software open source* dan akan berjalan diatas sistem operasi Ubuntu. Cara kerja *web server* terhubung dengan *server*, *email server*, *database server*, DNS *server*, serta *file transfer* untuk *upload* dan *download* berkas-berkas yang berada didalam *server*. *Web server* yang berkerja sebagai *backbone* dari *World Wide Web*. *Web server* digunakan untuk berkomunikasi dengan *client*-nya (*web browser*). Pelaksanaan pengujian dan menganalisis untuk pengembangan *hosting server* institusi menggunakan perangkat keras dan perangkat lunak dengan spesifikasi perangkat keras *server* merupakan komponen *server* komputer berfungsi operasional, penyimpan, dan proses sistem dengan spesifikasi seperti pada Tabel 1. Sedangkan spesifikasi perangkat lunak *hosting server*, dibutuhkan *software* seperti Sistem Operasi Linux Ubuntu *version* 14 dengan aplikasi didalamnya seperti :

1. *Web Server* : Apache, Nginx, dan PHP5
2. *Database Server* : Mysql dan PostgreSQL
3. *DNS Server* : Power DNS dan Bind
4. *Email Server* : Postfix, Roundcube, dan MS Outlook
5. *File Transfer* : SFTP dan FTP

III. PENGUJIAN DAN ANALISIS DATA

Pengujian dilakukan pada pengujian kecepatan akses layanan kedua *web server* melalui Ubuntu *server*, mulai dari waktu *request*, waktu *transfer*, dan waktu koneksi. Dan dengan menggunakan 5 sample pengaksesan halaman *website* yang akan diuji pada kedua *web server* tersebut.

A. *Prosedur dan pelaksanaan pengujian*

1. Membuka *putty* untuk *me-remote server*, lalu melakukan *login* pada *ubuntu server* dengan *account* yang telah dibuat sebelumnya, pastikan berada dalam *user "root"* ketika menjalankan *Ubuntu server*.

2. Setelah masuk kedalam *ubuntu server*, dilanjutkan dengan melakukan pengujian dengan waktu pengujian setiap *web server* dan *http* yang akan diujikan selama 60 *second* atau 1 menit dengan *request* pengaksesan secara bersamaan yaitu 10 koneksi. Pertama pengujian untuk *web server Apache* pada *Ubuntu server* dengan menggunakan perintah yang dijalankan pada terminal *Ubuntu server* "ab -c 10 -t60 http://ip.domain/xxx/xxx.php". Dimana "ab" ini adalah sebuah *utilities* untuk membaca *script benchmark*, c 10 adalah *request* secara bersamaan dalam satu waktu, dan t60 adalah lama waktu yang digunakan dan selanjutnya adalah *file* atau target yang ingin di uji. Perintah *benchmark* untuk *web server apache* yang diuji dengan akses alamat *web index.php* dan perintah *benchmark* untuk *web server Nginx* dengan akses alamat *web* yang serupa yaitu *index.php*. Dan mendapatkan hasil *benchmark*. Hasil ini menjelaskan tentang rentang waktu yang digunakan untuk akses *request*, *transfer*, dan koneksi. Hasil pengujian dari *halaman index* pertama dari kedua *web server* dari *benchmark apache server* untuk halaman *web index* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request*: 394.955 (ms)
- *Transfer rate*: 948.61 (Kbytes/sec)
- *Connection Times*: 393 (ms)

Hasil pengujian berikut dari *benchmark Nginx server* untuk halaman *web index* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 516.168 (ms)
- *Transfer rate* : 719.75 (Kbytes/sec)
- *Connection Times* : 513 (ms)

3. Selanjutnya pengujian dengan mengakses halaman *web* yang lain dengan metode yang sama seperti halaman *index* sebelumnya. Halaman *web* yang akan diuji dengan kedua *web server* ini yaitu *administrator* dari *website* tersebut dengan *script* "ab -c 10 -t60 http://172.16.0.XX/prestashop/adminXXX/index.php?controller=AdminLogin". Hasil yang didapat dari pengujian pertama untuk *server Nginx*. Hasil dari *benchmark Apache server* untuk halaman *web*

admin login yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 102.725 (ms)
- *Transfer rate* : 60.28 (Kbytes/sec)
- *Connection Times* : 103 (ms)

Hasil dari *benchmark Nginx server* untuk halaman *web admin login* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 148.332 (ms)
- *Transfer rate* : 40.09 (Kbytes/sec)
- *Connection Times* : 148 (ms)

4. Selanjutnya pengujian dengan mengakses halaman *web category* yang terdapat pada *website* yang akan diuji oleh kedua *web server* yang digunakan. *Script* yang digunakan untuk mengeksekusi halaman *web* tersebut yaitu "ab -c 10 -t60 http://172.16.0.XX/prestashop/index.php?id_category=5". Hasil yang didapat dari pengujian pertama halaman *web category* dijabarkan. Hasil dari *benchmark Apache server* untuk halaman *web category* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 482.511 (ms)
- *Transfer rate* : 773.83 (Kbytes/sec)
- *Connection Times* : 479 (ms)

Hasil dari *benchmark Nginx server* untuk halaman *web category* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 649.507 (ms)
- *Transfer rate* : 573.27 (Kbytes/sec)
- *Connection Times* : 648 (ms)

5. Pengujian halaman *website* yang keempat yaitu *authentication account* yang terdapat pada *website* yang diakses. Kedua *web server* menguji halaman *web authentication* dengan perintah *script* yang digunakan pada *Ubuntu server* yaitu "ab -c 10 -t60 http://172.16.0.xx/prestashop/index.php?controller=authentication". Hasil dari pengujian pertama *authentication account*. Hasil dari *benchmark apache server* untuk halaman *web authentication* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 474.366 (ms)
- *Transfer rate* : 771.70 (Kbytes/sec)
- *Connection Times* : 472 (ms)

Hasil dari *benchmark Nginx server* untuk halaman *web authentication* yang menggunakan waktu tes 60 *second* mendapatkan hasil seperti :

- *Time per request* : 675.777 (ms)
- *Transfer rate* : 539.91 (Kbytes/sec)
- *Connection Times* : 672 (ms)

6. Pengujian terakhir untuk kedua *web server* yaitu dengan pengaksesan halaman *web price* atau

diskon yang masih menggunakan metode yang sama yaitu dengan lama waktu yang digunakan yaitu 60 second dengan 10 koneksi pengaksesan secara bersamaan. Script yang digunakan pada Ubuntu server untuk pengujian terakhir yaitu "ab -c 10 -t60 http://172.16.0.XX/prestashop/index.php? controller=prices-drop". Hasil dari pengujian pertama dari halaman terakhir yaitu pada halaman *web price* atau diskon ini ditampilkan untuk *web server Apache* dan pada *web server Nginx*. Hasil dari pengujian *benchmark Apache server* untuk halaman *web* terakhir yaitu *price* yang menggunakan waktu tes 60 second mendapatkan hasil seperti :

- *Time per request* : 565.559 (ms)
- *Transfer rate* : 659.82 (Kbytes/sec)
- *Connection Times* : 564 (ms)

Hasil dari pengujian *benchmark Nginx server* untuk halaman *web* terakhir yaitu *price* yang menggunakan waktu tes 60 second mendapatkan hasil seperti :

- *Time per request* : 727.601 (ms)
- *Transfer rate* : 509.76 (Kbytes/sec)
- *Connection Times* : 725 (ms)

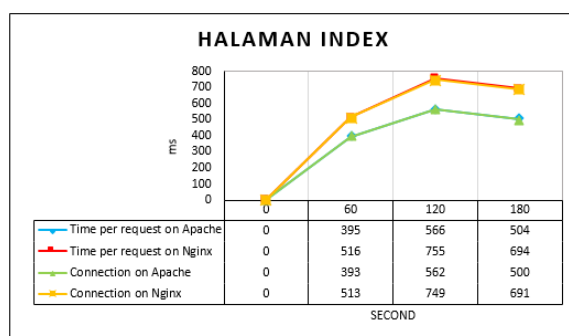
B. Data Hasil pengujian

Dari hasil pengujian yang telah dilakukan sebelumnya antara *web server Apache* dan *web server Nginx* yang diuji menggunakan *ubuntu server*, dengan waktu pengujian selama 60 second dan 10 request koneksi secara bersamaan dan pelaksanaan pengujian dilakukan 3 kali percobaan untuk setiap halamannya, maka didapatkan hasil seperti Tabel 1 – Tabel 5. Hasil didapatkan menggunakan pembulatan angka, jika di belakang koma muncul angka 1-5 maka dijadikan 0, sedang angka 6-9 dijadikan 1.

Hasil pengujian halaman *index* :

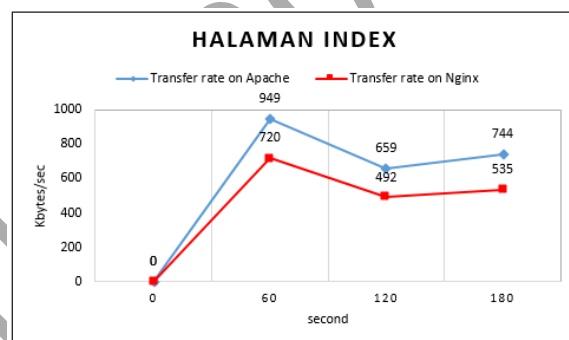
TABEL 1. PENGUJIAN HALAMAN INDEX

No	Variable	Percobaan 1	Percobaan 2	Percobaan 3
1	<i>Time per Request on Apache</i>	394.955 = 395 (ms)	565.654 = 566 (ms)	504.702 = 504 (ms)
2	<i>Time per Request on Nginx</i>	516.168 = 516 (ms)	754.924 = 755 (ms)	694.492 = 694 (ms)
3	<i>Connection Times on Apache</i>	393 (ms)	562 (ms)	500 (ms)
4	<i>Connection Times on Nginx</i>	513 (ms)	749 (ms)	691 (ms)
5	<i>Transfer Rate on Apache</i>	948.61 = 949 (Kbytes/s)	659.14 = 659 (Kbytes/s)	743.70 = 744 (Kbytes/s)
6	<i>Transfer Rate on Nginx</i>	719.75 = 720 (Kbytes/s)	492.09 = 492 (Kbytes/s)	535.05 = 535 (Kbytes/s)



Gambar 2. Grafik Pengujian Halaman Index

Pada Gambar 2 menjelaskan grafik dari hasil pengujian yang dilakukan pada halaman *index* untuk setiap variabelnya dari 60 second pertama sampe 60 second ketiga.

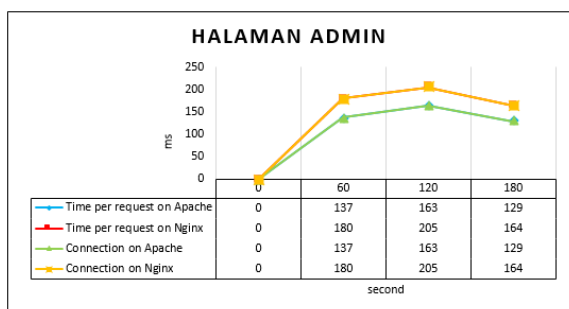


Gambar 3. Grafik Pengujian Halaman Index – Transfer Rate

Pada Gambar 3 menjelaskan grafik dari hasil pengujian *transfer rate* yang didapat dari halaman *index*. Hasil pengujian halaman *admin login I* seperti pada Tabel 2.

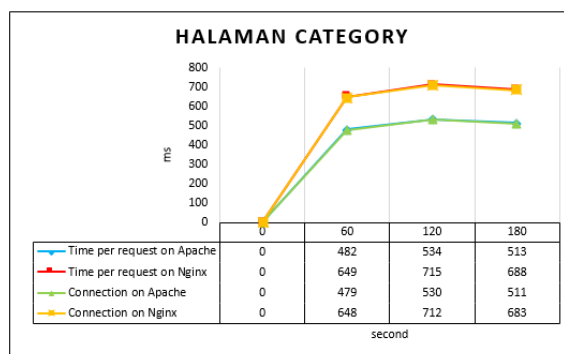
TABEL 2. PENGUJIAN HALAMAN ADMIN LOGIN

No	Variable	Percobaan 1	Percobaan 2	Percobaan 3
1	<i>Time per Request on Apache</i>	136.911 = 137 (ms)	163.195 = 163 (ms)	128.833 = 129 (ms)
2	<i>Time per Request on Nginx</i>	179.976 = 180 (ms)	205.421 = 205 (ms)	164.302 = 164 (ms)
3	<i>Connection Times on Apache</i>	137 (ms)	163 (ms)	129 (ms)
4	<i>Connection Times on Nginx</i>	180 (ms)	205 (ms)	164 (ms)
5	<i>Transfer Rate on Apache</i>	353.63 = 354 (Kbytes/s)	296.75 = 297 (Kbytes/s)	375.80 = 376 (Kbytes/s)
6	<i>Transfer Rate on Nginx</i>	268.12 = 268 (Kbytes/s)	235.01 = 235 (Kbytes/s)	293.61 = 294 (Kbytes/s)



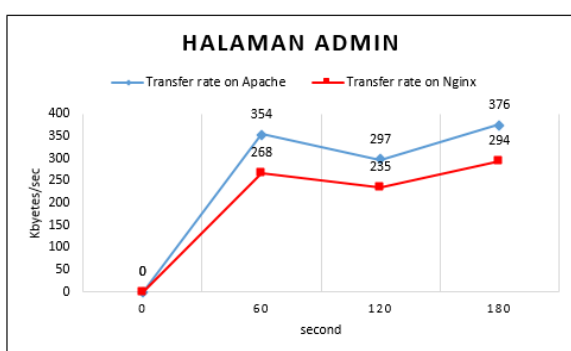
Gambar 4. Grafik Pengujian Halaman Admin Login

Pada Gambar 4 grafik yang dihasilkan dari pengujian halaman admin yang dilakukan sebanyak 3 kali percobaan.



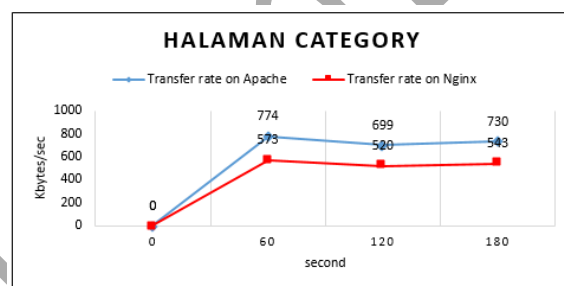
Gambar 6. Grafik Pengujian Halaman Category

Pada Gambar 6 grafik yang didapatkan dari halaman category pada pengujian web server pada variabel time per request dan connection.



Gambar 5. Grafik Pengujian Halaman Admin Login - Transfer Rate

Pada Gambar 5 grafik yang dihasilkan dari pengujian halaman admin yang dilakukan sebanyak 3 kali percobaan pada variable transfer rate. Hasil pengujian halaman category seperti pada Tabel 3.



Gambar 7. Grafik Halaman Category - Transfer Rate

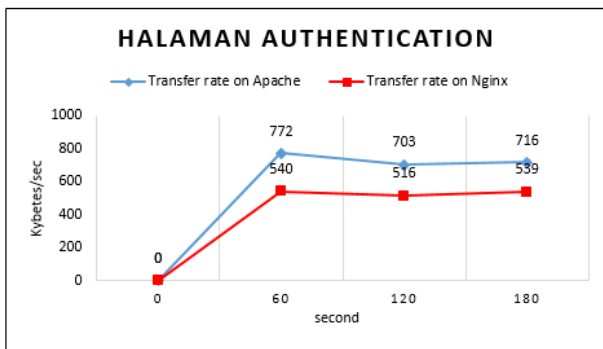
Pada Gambar 7 grafik yang didapatkan dari halaman category pada pengujian web server pada variable transfer rate. Hasil pengujian halaman authentication ditunjukkan pada Tabel 4.

TABEL 3. PENGUJIAN HALAMAN CATEGORY

No	Variable	Percobaan 1	Percobaan 2	Percobaan 3
1	Time per Request on Apache	482.511 = 482 (ms)	533.910 = 534 (ms)	512.826 = 513 (ms)
2	Time per Request on Nginx	649.507 = 649 (ms)	715.515 = 715 (ms)	687.647 = 688 (ms)
3	Connection Times on Apache	479 (ms)	530 (ms)	511 (ms)
4	Connection Times on Nginx	648 (ms)	712 (ms)	683 (ms)
5	Transfer Rate on Apache	773.83 = 774 (Kbytes/s)	699.34 = 699 (Kbytes/s)	729.95 = 730 (Kbytes/s)
6	Transfer Rate on Nginx	573.29 = 573 (Kbytes/s)	520.47 = 520 (Kbytes/s)	543.36 = 543 (Kbytes/s)

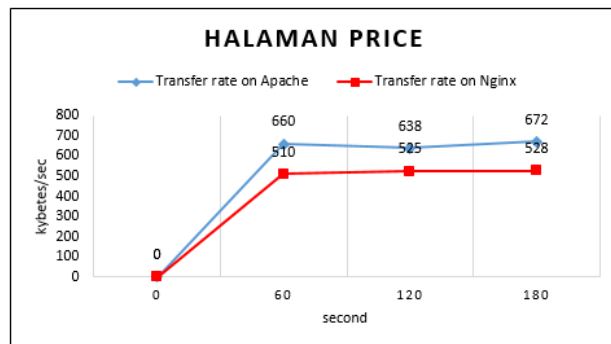
TABEL 4. PENGUJIAN HALAMAN AUTHENTICATION

No	Variable	Percobaan 1	Percobaan 2	Percobaan 3
1	Time per Request on Apache	474.366 = 474 (ms)	522.754 = 523 (ms)	514.894 = 515 (ms)
2	Time per Request on Nginx	675.777 = 676 (ms)	707.143 = 707 (ms)	676.734 = 677 (ms)
3	Connection Times on Apache	472 (ms)	521 (ms)	512 (ms)
4	Connection Times on Nginx	672 (ms)	704 (ms)	671 (ms)
5	Transfer Rate on Apache	771.70 = 772 (Kbytes/s)	702.98 = 703 (Kbytes/s)	716.12 = 716 (Kbytes/s)
6	Transfer Rate on Nginx	539.91 = 540 (Kbytes/sec)	516.15 = 516 (Kbytes/s)	539.35 = 539 (Kbytes/s)



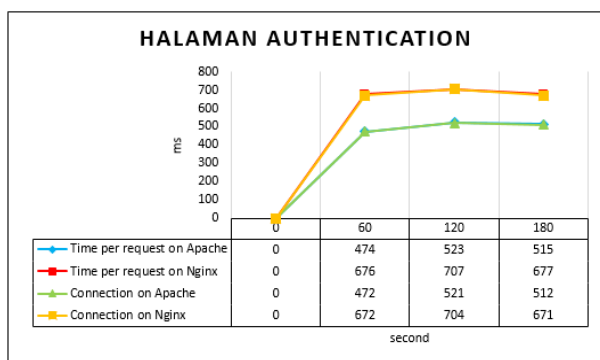
Gambar 8. Grafik Pengujian Halaman *Authentication* – *Transfer Rate*

Pada Gambar 8 grafik di atas menjelaskan nilai dari hasil pengujian *authentication* yang didapatkan dari pengujian *transfer rate*.



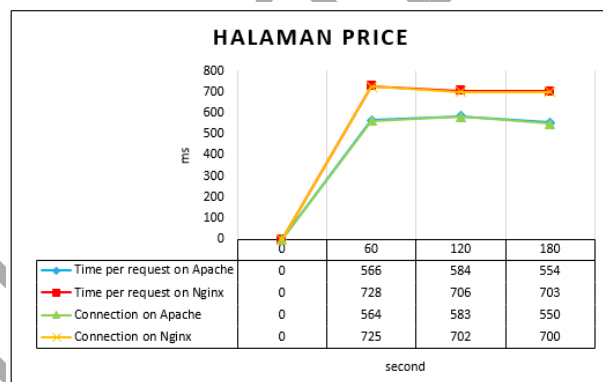
Gambar 10. Grafik Pengujian Halaman *Price* – *Transfer Rate*

Pada Gambar 10 grafik pengujian halaman *price* yang pengujiannya beracuan pada *variable transfer rate*.



Gambar 9. Grafik Pengujian Halaman *Authentication*

Pada Gambar 9 grafik di atas menjelaskan nilai dari hasil pengujian *authentication* yang didapatkan dengan 3 kali pengujian. Hasil pengujian halaman *price* seperti pada Tabel 5.



Gambar 11. Grafik Pengujian Halaman *Price*

Pada Gambar 11 grafik menjelaskan hasil yang didapat dari pengujian *web server*, angka yang ditampilkan sudah dibulatkan sebelumnya.

TABEL 5. PENGUJIAN HALAMAN *PRICE*

No	Variable	Percobaan 1	Percobaan 2	Percobaan 3
1	<i>Time per Request on Apache</i>	565.559 = 566 (ms)	583.658 = 584 (ms)	554.105 = 554 (ms)
2	<i>Time per Request on Nginx</i>	727.601 = 728 (ms)	706.432 = 706 (ms)	702.671 = 703 (ms)
3	<i>Connection Times on Apache</i>	564 (ms)	583 (ms)	550 (ms)
4	<i>Connection Times on Nginx</i>	725 (ms)	702 (ms)	700 (ms)
5	<i>Transfer Rate on Apache</i>	659.82 = 660 (Kbytes/s)	637.79 = 638 (Kbytes/s)	671.82 = 672 (Kbytes/s)
6	<i>Transfer Rate on Nginx</i>	509.76 = 510 (Kbytes/s)	525.25 = 525 (Kbytes/s)	528.08 = 528 (Kbytes/s)

C. Evaluasi hasil pengujian

Evaluasi hasil pengujian kecepatan *performance* dari kedua *web server* dari segi *time per request* Apache lebih unggul dari pada Nginx karena kecepatan *request* dari Apache lebih cepat dibandingkan dengan Nginx, itu terbukti dengan lebih kecilnya angka yang keluar dari Apache *server*. Bisa dikatakan semakin kecil angka yang dihasilkan semakin cepat proses *request*. Selain itu hasil proses *request* dari Apache lebih banyak dibandingkan Nginx.

Connection times web server Apache lebih baik dengan nilai rata-rata per *millisecond* yang kecil, sehingga koneksi yang dilakukan *user* ketika membuka *situs website* yang berada di *web server* Apache stabil. Sedangkan *connection times web server* Nginx nilai rata-rata yang dihasilkan lebih besar dibandingkan dengan Apache.

Dan terakhir dari segi *transfer rate* pada Apache memiliki nilai *transfer* yang cukup tinggi, sehingga proses *transfer* data atau *files* yang dilakukan di *web server* Apache lebih cepat.

Sebaliknya *transfer rate* yang dihasilkan Nginx lebih rendah dibandingkan dengan Apache, sehingga proses transfer data sedikit lambat.

IV. KESIMPULAN

Berdasarkan hasil penelitian kami dengan judul Analisis *Web server* Untuk Pengembangan *Hosting Server* Institusi, dan sub judul perbandingan kinerja *web server* Apache dengan *web server* Nginx, dapat disimpulkan bahwa :

Analisa kinerja dari kedua *web server* dengan parameter *transfer rate*, *time per request*, dan *connection time* sebagai indikator pengukuran untuk pengujian kinerja dari kedua *web server*. Dan metode pengujian menggunakan proses *benchmark* didapatkan hasil bahwa *transfer rate* pada *web server* Apache adalah 949 Kbytes/sec, 354 Kbytes/sec, 774 Kbytes/sec, 772 Kbytes/sec, 660 Kbytes/sec dan *web server* Nginx = 720 Kbytes/sec, 268 Kbytes/sec, 573 Kbytes/sec, 540 Kbytes/sec, 510 Kbytes/sec. *Time per request* pada *web server* Apache = 395 ms, 137 ms, 482 ms, 474 ms, 566 ms. Dan pada *web server* Nginx = 516 ms, 180 ms, 649 ms, 676 ms, 728 ms. *connection time* pada *web server* Apache = 393 ms, 137 ms, 479 ms, 472 ms, 564 ms dan *web server* Nginx = 513 ms, 180 ms, 648 ms, 672 ms, 725 ms.

Kesimpulan analisis dari hasil pengujian tersebut ditemukan *web server* Apache memiliki kinerja lebih baik dari pada *web server* Nginx, dimana kemungkinan pengguna *web server* Apache belum mengetahui. Hal tersebut bahwa *web server* Apache memiliki kecepatan *transfer rate*, *time per request* dan *connection time* lebih cepat dibandingkan dengan Nginx, dimana dengan nilai transfer rata-rata dari Apache 701 Kbytes/sec sedangkan Nginx 522 Kbytes/sec.

Kritik dan saran dari hasil penelitian ini dapat dikaji beberapa hal dalam melaksanakan pengembangan penggunaan *web server* agar menjadi bahan evaluasi sehingga penerapan pada institusi menjadi lebih baik. Kritik dan saran adalah sebagai berikut :

1. Pada sistem *hosting server* ini khususnya pada *web server* memfokuskan pada pembahasan mengenai pengujian kecepatan transfer data, waktu *request* dan koneksi saja dan belum memerhatikan aspek lainnya. Oleh karena itu peneliti menyarankan agar sistem ini dapat dikembangkan kembali dimasa yang akan datang.
2. Dalam proses *hosting server* Infrastruktur jaringan institusi, *web server* yang digunakan disesuaikan dengan kebutuhan, dilihat dari kelebihan dan kekurangan masing-masing *web server*.

3. Melakukan *maintenance* secara berkala yang bertujuan untuk menjaga kinerja *hosting server*, dan menjaga keamanan *hosting server* dari serangan *hacking* dan lain lain.

REFERENSI

- [1] Xiuquan Qiao, Guoshun Nan, Wei Tan, Lei Guo, Junliang Chen, Wei Quan, Yukai Tu, "CCNxTomcat: An extended web server for Content-Centric Networking", Computer Networks, Volume 75, Part A, 24 December 2014, Pages 276-296.
- [2] Andre Felipe Monteiro, Marcus Vinicius Azevedo, Alexandre Sztajnberg, "Virtualized Web server cluster self-configuration to optimize resource and power use", Journal of Systems and Software, Volume 86, Issue 11, November 2013, Pages 2779-2796.
- [3] Sopandi, Dede, *Instalasi dan Konfigurasi Jaringan Komputer*, Penerbit Informatika, Bandung, 2010.
- [4] Apache Web Server Complete Guide, Dedoimedo www.dedoimedo.com
- [5] Abdullah, Dahlan, "Perancangan dan Implementasi Virtual Hosting Menggunakan Linux". Jurnal Penelitian Teknik Informatika Universitas Malikussaleh, Lhokseumawe, Aceh, 2013.
- [6] Dwiyanakuntoko, Arsyad, *Protokol SSL dan TLS*. Jakarta, 2013.
- [7] Simarmata, Janner, "Rekayasa Web : Analisa dan Desain Sistem, Rekayasa Informasi, Rekayasa Kebutuhan, Data Mining, Manajemen Proyek", ANDI OFFSET, Yogyakarta, 2010.
- [8] Clément Nedelcu, *Nginx HTTP Server*, Packt Publishing, Birmingham, UK, ISBN 978-1-849510-86-8, 2010.
- [9] Xiaoni Chi; Bichuan Liu; Qi Niu; Qiuxuan Wu, "Web Load Balance and Cache Optimization Design Based Nginx under High-Concurrency Environment.". Third International Conference on Digital Manufacturing & Automation, 2012 Pages: 1029 – 1032.
- [10] Sverdlov, Etel, <https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-on-ubuntu-12-04>, 2012. [20 Mei 2015].
- [11] Sokam, Apache vs Nginx, <http://sokam.or.id/log/592/apache-vs-nginx>, 08 Mei 2014