REKAYASA PERANGKAT LUNAK

Kinerja Web Service pada Web Server Apache, Ngin-X dan IIS-7

Yogiswara [#1]

[#]*Jurusan Teknologi Informasi ,Politeknik Negeri Jember*
JL Mastrip PO BOX 164 Jember
[1]yogipoltek@gmail.com

Abstract

Teknologi layanan web dikembangkan untuk mengatasi perbedaan platform. Teknologi ini banyak diimplementasikan pada sistem informasi terdistribusi diantaranya untuk mengakumulasi data dari banyak sumber atau server. Penelitian ini difokuskan untuk mendapatkan kinerja terbaik dari tiga web server yang memiliki platform berbeda yaitu Apache, Ngin-X dan IIS-7 dalam mengimplementasikan layanan web dengan mengukur nilai latency masing masing web server. Berdasarkan hasil evaluasi kinerjanya didapatkan penggunaan web server "*nginx*" dengan menggunakan metode "*rest*" memiliki nilai latency terbaik ketika diimplementasikan pada transaksi penambahan, perubahan, pemilihan dan penghapusan data.

Keywords— **Web Server, Apache, Ngin-X, IIS7, Xmlrpc, Soap, Rest**

I. PENDAHULUAN

Isu utama teknologi layanan web adalah permasalahan kualitas layanan atau Qos (*Quality of service*). Menurut Kuyoro Shade, 2012 kemampuan kinerja layanan web direpresentasikan dengan kecepatan layanan dalam menyelesaikan permintaan klien. Pengukuran kinerja layanan web salah satunya dapat dilakukan dengan mengukur *latency* yang merupakan lama waktu yang diperlukan dari pengiriman permintaan oleh klien sampai menerima hasil permintaan tersebut dari server. Aspek yang mempengaruhi kinerja layanan web adalah platform web server. Menurut Toyotaro 2008, implementasi layanan web pada web server yang berbeda platform akan menghasilkan kinerja yang berbeda.

Penelitian ini dilakukan dengan mengukur kinerja tiga jenis platform web server yang berbeda yaitu Apache Web Server, Ngin-X Web Server dan IIS7 Web Server dalam mengimplementasi metode layanan web dengan mengukur nilai latency.

Dari penelitian ini diharapkan dapat menjadi referensi dan kontribusi bagi para pengembang aplikasi terdistribusi dalam meningkatkan layanan data yang berbasis layanan web atau web service

II. TINJAUAN PUSTAKA

A. Layout Halaman

Layanan web (Web Service) adalah layanan yang tersedia melalui Internet yang menggunakan sistem pesan dengan standar XML dan tidak terikat pada satu sistem operasi atau bahasa pemrograman. Ada beberapa alternatif untuk menjalankan pesan XML yaitu dengan menggunakan Remote Procedure Calls XML (XML-RPC) atau SOAP atau bisa menggunakan HTTP GET / POST yang secara sistem akan leluasa melewatkan segala bentuk dokumen XML. Meskipun tidak wajib, layanan web juga memiliki dua sifat. Pertama Sebuah layanan web harus self-describing, intinya Jika mempublikasikan sebuah layanan web baru, sebaiknya juga harus menerbitkan antarmuka publik ke layanan. Minimal, layanan anda harus mencakup dokumentasi yang dapat terbaca manusia sehingga pengembang lain dapat lebih mudah mengintegrasikan layanan tersebut. Jika kita telah membuat layanan SOAP, idealnya anda juga harus membuat antarmuka publik yang ditulis dalam tata bahasa XML umum.

Tata bahasa XML dapat digunakan untuk mengidentifikasi semua metode umum, metode argumen, dan nilai-nilai timbal baliknya. Kedua, sebuah layanan web harus discoverable. Jika anda membuat sebuah layanan web, harus ada mekanisme yang sederhana untuk mempublikasikan data tersebut dan harus ada mekanisme yang sederhana agar pihak yang berkepentingan dapat menemukan layanan tersebut. karakteristik layanan web sebagai berikut :

B. Web Server

Web server merupakan perangkat lunak untuk memberikan layanan data dengan menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *webbrowser* dan mengirimkan kembali hasilnya dalam bentuk halaman halaman web yang umumnya berbentuk dokumen HTML.Hubungan antara Web Server dan Browser Internet merupakan gabungan atau jaringan komputer yg ada di seluruh dunia. Setelah terhubung secara fisik, Protocol TCP/IP (networking protocol) memungkinkan semua komputer dapat berkomunikasi satu dengan yg lainnya. Pada saat browser meminta data web page ke server maka instruksi permintaan data oleh browser tersebut di kemas di dalam TCP yg merupakan protocol transport dan dikirim ke alamat menggunakan protocol berikutnya yaitu Hyper Text Transfer Protocol (HTTP). HTTP ini merupakan protocol yg digunakan dalam World Wide Web (WWW) antar komputer yg terhubung dalam jaringan di dunia ini.Data yg di passing dari browser ke web server disebut sebagai HTTP request. Data yg dikirim dari server ke browser disebut sebagai HTTP response. Jika data yg diminta oleh browser tidak ditemukan oleh si Web server maka akan menampilkan error yg sering anda lihat di web page yaitu *Error : 404 Page Not Found*. Didalam penelitian ini terdapat tiga produk web server yang digunakan yaitu :

1. Apache

Secara umum arsitektur apache bekerja dengan model *thread-based* atau berbasis proses. Web server berbasis proses menggunakan proses (*thread*) untuk menerima dan merespon permintaan. Setiap permintaan akan diciptakan sebuah thread yang disimpan dalam sebuah pool pada alokasi memori tertentu dan dilakukan proses untuk menjawab, setelah proses dieksekusi kemudian hasil proses dikirimkan kembali ke klien serta dicatat dalam sebuah log

2. IIS 7

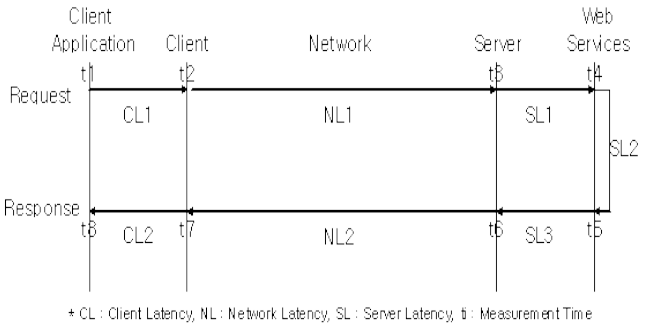
Secara struktur model kinerja arsitektur IIS melayani permintaan melalui application pool yang merupakan komponen *host worker process*. *Application pool* merupakan improvisasi dari arsitektur iis. Karena situs web yang terpisah dan aplikasi web yang terpisah dapat dilayani melalui konfigurasi *application pool.* Didalam application pool permintaan dari klien diproses oleh *event-handler* dan permintaan tersebut diterjemahkan dan dieksekusi dengan mengoleksi informasi yang diperlukan sebagai jawaban permintaan melalui ASPX (*active server page*). Proses dilanjutkan dengan mengirimkan respon ke klien dan mencatat log bahwa permintaan telah direspon.

3. NginX

Secara umum web server nginx menerapkan model *event-based*. Model *event-based* hanya menggunakan sebuah thread untuk memproses permintaan dan mengelola *event* dengan menggunakan multiplexing dan banyak notifikasi *event*. Setiap koneksi diproses secara sangat efisien dalam sejumlah proses *single threaded* yang disebut *workers*. Dalam setiap nginx, *workers* dapat menangani ribuan koneksi bersamaan dan permintaan per detik

C. Latency

latency adalah waktu antara pengiriman suatu permintaan dan menerima tanggapan. Latency merupakan salah satu parameter pengukuran kinerja yang digunakan pada aplikasi perangkat lunak [4]. Terdapat dua jenis latency. Latency koneksi dan latency permintaan. latency koneksi mencerminkan waktu yang dibutuhkan untuk membuat sambungan, latencypermintaan mencerminkan waktu untuk menyelesaikan transfer data sekali sambungan. Dalam penelitian ini latency yang diukur menggunakan *user perceived latency* yang meliputi jumlahkoneksidari permintaanlatency, ditambah latency jaringan karena koneksi wan (*wireless area network*) atau router.



Gambar 1. komponen latency

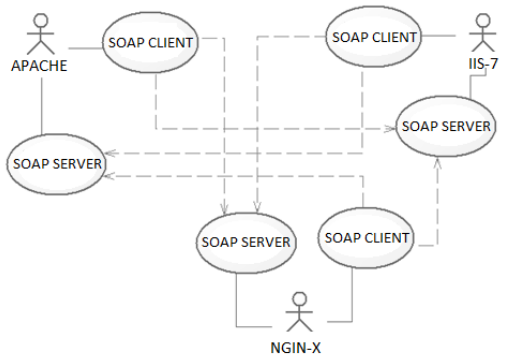
Tiga komponen latency dalam pengukuran latency terlihat pada diatas. CL (*Client Latency*) waktu latency yang diperlukan oleh klien saat mengirim ke jaringan dan menrima data dari jaringan. NL (*Network Latency*) waktu yang diperlukan jaringan dalam mengirimkan data. SL (*Server Latency*) waktu yang diperlukan server menerima permintaan melayani permintaan dan mengirimkan hasil permintaan ke jaringan.

III. PERANCANGAN SISTEM

Sistem yang dirancang dalam penilaian kinerja dilakukan dengan membuat desain sistem dalam bentuk sebuah model aplikasi berbasis pesan yang dapat dieksekusi menggunakan aplikasi lain menggunakan metoda xmlrpc, soap, rest. Rancangan model yang didesain untuk penilaian kinerja pada tiga buah server terlihat pada gambar 1.

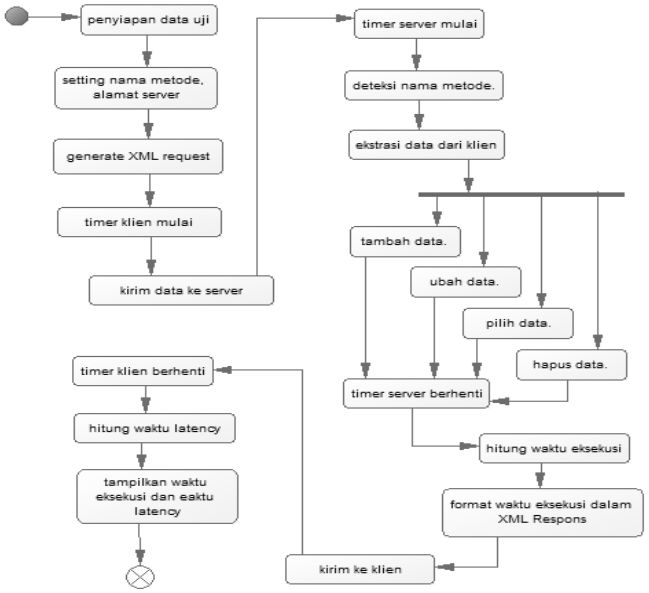
use case model yang direncanakan bertujuan untuk mengukur *latency* ketiga metode layanan web pada tiga jenis web server yang memiliki perbedaan platform dan arsitektur. Pada gambar 6 dapat dilihat masing masing web server

memiliki modul klien dan modul server layanan web yang dibuat dengan menerapkan metoda xmlrpc, soap dan rest. Kebutuhan penilaian adalah menghasilkan keluaran berupa data latency dalam satuan mili detik untuk semua fungsi yang dimiliki masing masing aktor.



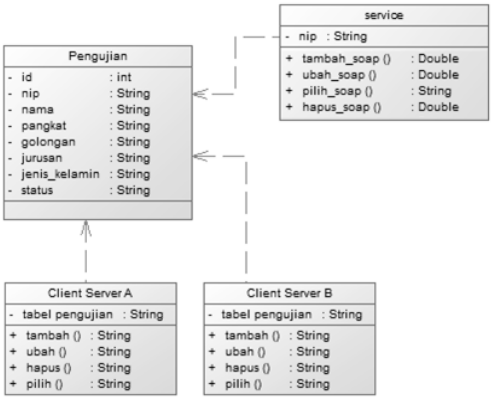
Gambar 2. Use Case Model Penilaian kinerja

Secara lebih rinci aktivitas yang direncanakan pada setiap server atau actor dalam model use case dideskripsikan dalam diagram aktivitas yang terlihat pada gambar 7 dibawah ini.



Gambar 3. Activity diagram penilaian kinerja

berdasarkan diagram aktivitas tersebut, masing masing aplikasi dirancang sebuah database dengan struktur yang sama sehingga setiap aplikasi akan memiliki layanan yang dapat diakses oleh server lain. Adapun transaksi antar server berupa penambahan, perubahan, pemilihan maupun penghapusan data pada tabel yang berada di server. Keluaran dari layanan adalah berupa data waktu eksekusi yang diperlukan oleh server dalam memproses instruksi dari klien. Representasi model database dirancang dalam sebuah model class diagram sebagai berikut



Gambar 4. Class Diagram penilaian kinerja

IV. PENGEMBANGAN SISTEM

Hasil desain sistem yang dirancang melalui pemodelan use case, activity dan class pada bagian sebelumnya dikembangkan dengan membuat aplikasi yang dipasang pada ketiga server. Pengembangan dilakukan dengan menggunakan CodeIgniter yang merupakan framework aplikasis berbasis web dengan bahasa PHP. Sedangkan database yang dibangun menggunakan mysql adapun menu menu yang dikembangkan pada aplikasi menyesuaikan pada kebutuhan masing masing server. Berikut gambaran menu pada masing masing server.

TABEL 1 :
IMPLEMENTASI MENU APLIKASI PENILAIAN KINERJA SOAP

Server	Menu
Apache	Client nginX – Tambah Data
	Client NginX – Ubah Data
	Client NginX – Hapus Data
	Client NginX – Pilih Data
	Client IIS – Tambah Data
	Client IIS – Ubah Data
	Client IIS – Hapus Data
	Client IIS – Pilih Data
IIS-7	Client nginX – Tambah Data
	Client NginX – Ubah Data
	Client NginX – Hapus Data
	Client NginX – Pilih Data
	Client Apache – Tambah Data
	Client Apache – Ubah Data
	Client Apache – Hapus Data
	Client Apache – Pilih Data
NginX	Client Apache – Tambah Data
	Client Apache – Ubah Data
	Client Apache – Hapus Data
	Client Apache – Pilih Data
	Client IIS – Tambah Data
	Client IIS – Ubah Data
	Client IIS – Hapus Data
	Client IIS – Pilih Data

Aplikasi yang dikembangkan mengguna- kan kerangka MVC (Model – View – Controller) dimana kodefikasi yang dikembangkan terdiri dari 2 modul yaitu modul klien dan modul service. Modul tersebut terpasang pada tiga server dan masing masing server akan berfungsi sebagai klien dan sebagai server. Adapun struktur file dalam pengembangan modul ditunjukkan pada tabel 1.

TABEL 2
STRUKTUR FILE PENGEMBANGAN SISTEM PENILAIAN KINERJA

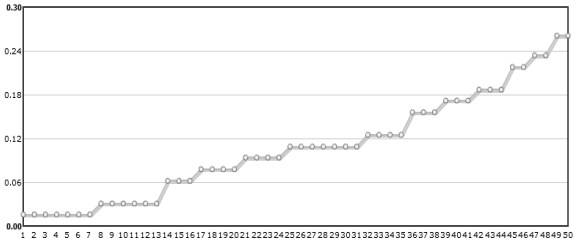
Modul	Nama File	Fungsi
Modul Klien	xmlrpc_tambah	Permintaan tambah data
	xmlrpc_ubah.php	permintaan perubahan data
	xmlrpc_hapus.php	Permintaan penghapusan data
	xmlrpc_pilih.php	Permintaan pemilihan data
Modul Server	Modul_uji_xmlrpc.php	Eksekusi layanan web berdasar permintaan klien
Modul Klien	soap_tambah	Permintaan tambah data
	soap_ubah.php	permintaan perubahan data
	Soap_hapus.php	Permintaan penghapusan data
	Soap_pilih.php	Permintaan pemilihan data
Modul Server	Modul_uji_soap.php	Eksekusi layanan web berdasar permintaan klien
Modul Klien	rest_tambah	Permintaan tambah data
	rest_ubah.php	permintaan perubahan data
	rest_hapus.php	Permintaan penghapusan data
	rest_pilih.php	Permintaan pemilihan data
Modul Server	Modul_uji_rest.php	Eksekusi layanan web berdasar permintaan klien

Data beban kerja yang dibuat terdiri dari tujuh field data dan besaran beban yang dibuat adalah jumlah record dalam data array yang dikirimkan untuk diinputkan pada basis data server penerima. Pada proses perubahan, data array yang dikirimkan dalam *xml request* adalah data perubahan tujuh kolom data dimana salah satu kolomnya menjadi parameter *record* yang dirubah. Besaran beban dalam proses perubahan adalah jumlah *record* yang mengalami perubahan data pada server penerima. Pada proses pemilihan data, data *xml request* hanya berisi satu data parameter sebagai bahan permintaan data dan respon yang akan dihasilkan adalah data lengkap tujuh kolom berdasarkan parameter permintaan. Besaran beban kerja didasarkan jumlah data parameter yang dikirimkan. Hal ini juga berlaku untuk proses penghapusan data.

V. HASIL PENGUKURAN

Use case model yang direncanakan untuk pengujian kinerja web server bertujuan untuk mengukur *latency* tiga jenis web server yang memiliki perbedaan arsitektur. Kebutuhan pengujian adalah menghasilkan keluaran berupa data *latency* dalam satuan mili detik untuk semua fungsi yang dimiliki masing-masing aktor. Didalam implementasinya. Tabel yang digunakan untuk melakukan transaksi dimasukkan dalam database masing masing web server.

Pengukuran waktu *latency* dilakukan melalui program klien. Program tersebut mengirimkan permintaan berupa penambahan data, perubahan data, pemilihan data dan penghapusan data, permintaan tersebut akan dilayani oleh server dan server mengeksekusi permintaan kedalam basis data dan mengirimkan respon data ke klien. Pengiriman permintaan dilakukan berulang ulang berdasarkan rancangan beban data. Beban data berupa sebuah data array yang terdiri baris data yang dirancang dari 1 baris data sampai 50 baris data. Sehingga untuk satu pengujian evaluasi kinerja akan didapatkan 50 nilai latency dari beban 1 baris data hingga 50 baris data. Nilai latency tersebut disimpan dalam basis data dan ditampilkan oleh aplikasi dalam bentuk sebuah grafik nilai latency terhadap jumlah baris data. Berikut salah satu bentuk keluaran grafik yang dihasilkan oleh aplikasi.



Gambar. 5. Grafik nilai latency terhadap beban data

Dari pengamatan dihasilkan berdasarkan konfigurasi yang ditetapkan menunjukkan adanya pengaruh beban data terhadap lamanya waktu latency. Hal ini ditunjukkan pada setiap penambahan jumlah beban data menghasilkan nilai latency yang semakin besar.

Tabel 3 dan tabel 4 adalah urutan data nilai latency penambahan data dan perubahan data berdasarkan 18 konfigurasi yang telah ditetapkan. no urutan pada tabel tersebut merupakan nilai latency tercepat. Tabel 5 dan 6 adalah data urutan nilai latency untuk jenis transaksi, pemilihan data dan penghapusan data. Tabel tersebut berisi data hasil analisis deskriptif menggunakan metode *tendency central* yang menampilkan nilai latency terkecil dan terbesar serta rata rata nilai latency dari 50 data yang dicatat. Dari hasil analisis deskriptif tersebut dapat diamati dan disimpulkan bahwa metode rest memiliki kinerja lebih baik dari metode lainnya, sedangkan konfigurasi yang memiliki nilai latency tercepat yang dihasilkan adalah konfigurasi dengan web server "Apache" sebagai klien, web server "Nginx" sebagai pemilik service atau sebagai server,

sedangkan metode terbaik yang digunakan untuk proses integrasi data adalah metode rest.

TABEL 3
STATISTIK DESKRIPTIF NILAI LATENCY PENAMBAHAN DATA

No	server-klien-metode	N	Min	Max	Mean
1	nginx-apache-rest	50	0.002	0.016	0.0038
2	apache-nginx-rest	50	0.009	0.076	0.01384
3	apache-iis-rest	50	0.009	0.016	0.01562
4	nginx-iis-rest	50	0.001	0.031	0.0187
5	iis-nginx-rest	50	0.014	0.047	0.02238
6	iis-apache-rest	50	0.014	0.047	0.0236
7	apache-nginx-xmlrpc	50	0.016	0.162	0.0416
8	nginx-iis-xmlrpc	50	0.016	0.182	0.04234
9	nginx-apache-xmlrpc	50	0.007	0.192	0.04288
10	nginx-apache-soap	50	0.01	0.16	0.0436
11	apache-iis-xmlrpc	50	0.016	0.094	0.04986
12	iis-apache-xmlrpc	50	0.03	0.098	0.06414
13	apache-nginx-soap	50	0.032	0.11	0.06618
14	nginx-iis-soap	50	0.016	0.125	0.0668
15	iis-nginx-xmlrpc	50	0.03	0.121	0.07004
16	apache-iis-soap	50	0.031	0.139	0.08328
17	iis-apache-soap	50	0.057	0.891	0.48714
18	iis-nginx-soap	50	0.044	1.417	0.52428

TABEL 4
STATISTIK DESKRIPTIF NILAI LATENCY PERUBAHAN DATA

No	server-klien-metode	N	Min	Max	Mean
1	nginx-apache-rest	50	0.006	0.188	0.09472
2	nginx-iis-rest	50	0.016	0.187	0.10144
3	apache-nginx-rest	50	0.016	0.261	0.10812
4	nginx-apache-soap	50	0.012	0.246	0.12626
5	nginx-iis-xmlrpc	50	0.016	0.25	0.12948
6	nginx-apache-xmlrpc	50	0.009	0.38	0.13828
7	nginx-iis-soap	50	0.016	0.265	0.14696
8	apache-iis-rest	50	0.016	0.328	0.15666
9	apache-nginx-xmlrpc	50	0.031	0.31	0.15994
10	apache-nginx-soap	50	0.024	0.352	0.18222
11	apache-iis-xmlrpc	50	0.031	0.343	0.1825
12	apache-iis-soap	50	0.031	0.39	0.21154
13	iis-nginx-rest	50	0.033	1.041	0.48192
14	iis-apache-xmlrpc	50	0.049	0.965	0.50638
15	iis-apache-soap	50	0.058	0.963	0.50946
16	iis-nginx-soap	50	0.054	1.047	0.51032
17	iis-apache-rest	50	0.031	1.025	0.5166
18	iis-nginx-xmlrpc	50	0.05	1.197	0.53012

TABEL 5
STATISTIK DESKRIPTIF NILAI LATENCY PEMILIHAN DATA

No	server-klien-metode	N	Min	Max	Mean
1	nginx-apache-rest	50	0.009	0.023	0.01596
2	nginx-iis-rest	50	0.016	0.031	0.0262
3	apache-nginx-rest	50	0.031	0.042	0.03848
4	apache-iis-rest	50	0.031	0.062	0.0425
5	apache-nginx-xmlrpc	50	0.016	0.075	0.0425
6	nginx-iis-xmlrpc	50	0.016	0.078	0.04272
7	nginx-apache-xmlrpc	50	0.009	0.086	0.04672
8	nginx-apache-soap	50	0.011	0.094	0.05236
9	iis-nginx-xmlrpc	50	0.016	0.086	0.0526
10	apache-iis-xmlrpc	50	0.016	0.094	0.05486
11	iis-apache-xmlrpc	50	0.022	0.087	0.05494
12	nginx-iis-soap	50	0.016	0.109	0.0658
13	apache-nginx-soap	50	0.029	0.107	0.07178
14	iis-nginx-rest	50	0.047	0.086	0.072
15	apache-iis-soap	50	0.031	0.109	0.07298
16	iis-apache-rest	50	0.071	0.088	0.0755
17	iis-nginx-soap	50	0.03	0.181	0.1033
18	iis-apache-soap	50	0.033	0.187	0.10916

Konfigurasi web klien apache, web server nginx dan metode rest berlaku pada semua jenis transaksi yaitu transaksi penambahan data, perubahan data, pemilihan data dan penghapusan data.

TABEL 6
STATISTIK DESKRIPTIF NILAI LATENCY PENGHAPUSAN DATA

No	server-klien-metode	N	Min	Max	Mean
1	nginx-apache-rest	50	0.005	0.015	0.00552
2	iis-nginx-rest	50	0.009	0.015	0.01235
3	iis-apache-rest	50	0.01	0.015	0.01246
4	nginx-apache-xmlrpc	50	0.007	0.022	0.01376
5	nginx-iis-rest	50	0.001	0.031	0.0145
6	nginx-apache-soap	50	0.013	0.025	0.01626
7	apache-iis-rest	50	0.016	0.031	0.0163
8	nginx-iis-xmlrpc	50	0.008	0.031	0.01944
9	apache-nginx-res	50	0.016	0.031	0.0244
10	nginx-iis-soap	50	0.016	0.047	0.02982
11	apache-nginx-xmlrpc	50	0.029	0.031	0.03048
12	apache-iis-xmlrpc	50	0.016	0.031	0.0307
13	apache-nginx-soap	50	0.031	0.052	0.03442
14	iis-nginx-soap	50	0.037	0.054	0.03904
15	iis-nginx-xmlrpc	50	0.03	0.077	0.04274
16	iis-apache-xmlrpc	50	0.031	0.063	0.04516
17	apache-iis-soap	50	0.031	0.062	0.0466
18	iis-apache-soap	50	0.044	0.058	0.05084

Berdasarkan data tersebut secara umum perbedaan arsitektur web server menunjukkan pengaruh yang beragam terhadap nilai latency pada semua jenis transaksi

VI. KESIMPULAN

Web server dan metode pertukaran data memiliki pengaruh terhadap lamanya proses pertukaran data. Konfigurasi web server "Nginx" dengan klien "Apache" menggunakan metode "REST" merupakan konfigurasi dengan nilai latency terbaik untuk diimplementasikan dalam proses integrasi data.

Sebagai saran untuk penelitian lebih lanjut, penambahan parameter throughput, respon time dan execution time sebagai parameter pengujian perlu dilakukan untuk menghasilkan analisis kinerja yang lebih mendalam. Selain itu modul integrasi data disempurnakan dengan modul yang dapat menerima mendeteksi metode pertukaran data sebagai bentuk fleksibilitas teknologi *web service*.

DAFTAR PUSTAKA

[1] Juric, M.B., Loganathan, R., Sarang, P., dan Jennings, F. 2007. SOA Approach to Integration, Packt Publishing, Birmingham, B27 6PA, UK.

[2] Ladan Mohamad Ibrahim,. Ph.D. 2011. *Web Services Metrics A Survey and A Classification*, International Conference on Network and Electronics Engineering IPCSIT vol.11 (2011) © (2011) IACSIT Press, Singapore

[3] Kuyoro Shade. 2012. Quality of Service (Qos) Issues in Web Services, IJCSNS International

[4] Toyotaro Suzumaru .2008. Performance Comparison of Web Service Engines in PHP, Java, and CG. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64