

**ANALISIS KINERJA KOMPUTASI TERDISTRIBUSI  
DENGAN PLATFORM WEB SERVICE MENGGUNAKAN METODE REST  
REPRESENTATIONAL STATE TRANSFER**

Oleh :  
Yogiswara \*)

**ABSTRAK**

Teknologi Komputasi terdistribusi seperti *Common Object Request Broker Architecture (CORBA)*, *Distributed Component Object Model (DCOM)*, *Remote Method Invocation (RMI)* memiliki ketergantungan terhadap platform sistem operasi dan tidak memiliki kemampuan interoperabilitas sehingga memerlukan infrastruktur yang kompleks untuk diimplementasikan pada proses integrasi data. Teknologi layanan web (*web service*) adalah teknologi yang dapat menyelesaikan permasalahan interoperabilitas antar aplikasi yang berbeda platform baik perangkat lunak, sistem operasi maupun bahasa pemrograman. Penelitian ini difokuskan untuk menganalisis kinerja sebuah web service dengan mengukur kemampuan web service metode rest pada proses transaksi data. Parameter yang digunakan dalam mengevaluasi kinerja adalah pengukuran lama waktu yang diperlukan dari pengiriman permintaan oleh klien sampai klien menerima hasil permintaan tersebut dari server atau biasa disebut *user perceived latency*. Sedangkan pengujian dilakukan pada 3 buah web server yang memiliki perbedaan arsitektur. Hasil Analisis menunjukkan metode *REST* memerlukan desain yang memiliki struktur data secara terperinci agar eksekusi klien dapat dijalankan dengan baik oleh server, selain itu web server nginx menjadi web server yang terbaik dalam mengeksekusi service terutama web service dengan metoda *REST*.

**Kata Kunci :** Web service, Rest

**PENDAHULUAN**

Teknologi web merupakan salah satu bidang teknologi informasi yang memiliki perkembangan begitu pesat dibandingkan dengan teknologi lainnya pada bidang yang sama. Teknologi web yang sudah memasuki generasi kedua dan lebih dikenal dengan nama web 2.0 telah memberi dampak perubahan yang nyata pada pembuatan dan pengembangan aplikasi web. Hal ini terlihat jelas pada kemunculan sejumlah aplikasi berbasis web yang kaya akan isi serta kemampuan mengintegrasikan data yang cukup handal.

Web service merupakan teknologi yang memungkinkan sistem heterogen untuk berkomunikasi satu sama lain menggunakan pesan dengan format XML (*Extensible Mark Up Language*). Kondisi sistem yang bisa heterogen mengakibatkan munculnya kebutuhan akan interoperabilitas. Oleh karena itu XML sering digunakan untuk memformat pesan, hal ini dikarenakan XML terbentuk dalam format teks sehingga hampir semua sistem dapat memahami pesan dan bekerja dengan satu sama lain. Pesan akan digunakan untuk komunikasi antar aplikasi yang berjalan di mesin yang berbeda. Sebagai contoh, dalam aplikasi chat, teks yang diketik oleh pengguna di kemas dalam bentuk pesan, bersama dengan data yang akan menjelaskan mana pesan harus di kirim dan bagaimana pesan tersebut harus diinterpretasikan antara aplikasi server.

*Representational State Transfer (REST)* merupakan pendekatan yang menjadikan fungsi-fungsi aplikasi menjadi *service*, yang dipaket sebagai bagian yang dapat digunakan ulang. Pada teknologi ini setiap sumber daya akan diberikan sebuah ID yang unik (misalnya, URL dokumen) yang memungkinkan sumber daya dapat berhubungan menggunakan standar (HTTP, HTML, XML). *REST* terletak pada kemampuannya mengakses sumber (*source*) melalui alamat unik yang dapat mengidentifikasi data yang dibutuhkan. Hanya dengan mengkonstruksikan URL (*Uniform Resource Locator*) maupun URI (*Uniform Resource Identifier*) pada suatu layanan web (*web service*).

Isu utama teknologi layanan web adalah permasalahan kualitas layanan atau Qos (*Quality of service*). Menurut Mohamad Ibrahim Ladan, 2011 matrik kualitas layanan pada layanan web terbagi dalam beberapa hal diantaranya adalah kemampuan layanan web dalam melayani permintaan klien (*accessibility*), kemam puan layanan web untuk melaksanakan fungsinya pada interval waktu tertentu (*reliability*), dan kemampuan kinerja (*performance*). Menurut Kuyoro Shade, 2012 kemampuan kinerja layanan web direpresentasikan dengan kecepatan layanan dalam menyelesaikan permintaan klien. Pengukuran kinerja layanan web dapat dilakukan berdasarkan *response time* yaitu lama waktu yang dibutuhkan untuk menyelesaikan permintaan, *throughput* yaitu jumlah permintaan yang dapat diselesaikan dalam periode tertentu,

\*) Staf Pengajar Jurusan Teknologi Informasi, Politeknik Negeri Jember

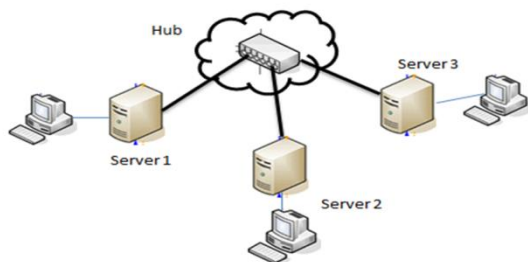
latency yang merupakan lama waktu yang diperlukan dari pengiriman permintaan oleh klien sampai menerima hasil permintaan tersebut dari server. *Respon time dan throughput* lebih banyak dipengaruhi faktor kualitas perangkat keras dan jaringan sedangkan *latency* lebih mengarah pada pengukuran kinerja aplikasi perangkat lunak.

Berdasarkan hal tersebut maka penelitian ini bertujuan untuk menilai kinerja web service menggunakan metode REST dalam melakukan transaksi data dalam hal ini adalah transaksi penambahan, perubahan, penghapusan dan pemilihan data..

### METODOLOGI

Metodologi dalam penelitian ini dilakukan dengan mengembangkan sebuah model integrasi data antar server yang menerapkan teknologi web service menggunakan metode REST sebagai media transaksi antar servernya. proses pengembangan sistem diawali dengan perancangan desain prototipe aplikasi melalui identifikasi kebutuhan sistem dan perancangan desain aplikasi. Proses dilanjutkan dengan instalasi dan konfigurasi server. Tahap terakhir dengan membuat aplikasi serta melakukan pengujian. Berikut alur kegiatan penelitian yang dilakukan

- (a) Tahap Perancangan desain prototipe aplikasi dengan membuat model transaksi yang akan menjadi beban kerja bagi service yang akan dilakukan oleh masing masing server.
- (b) Melakukan proses instalasi dan konfigurasi web server pada tiga buah server yang berbeda dimana ketiga web server tersebut memiliki perbedaan arsitektur sehingga perilaku dalam mengeksekusi service juga berbeda.
- (c) tahap pengembangan modul merupakan tahap pengembangan service yang diperlukan untuk pengukuran kinerja masing masing service akan dipasang pada setiap server . server tersebut juga berfungsi sebagai klien.
- (d) hasil pengukuran akan dikoleksi dan dianalisis menggunakan analisis deskriptif untuk mendapatkan kinerja terbaik metode REST terhadap jenis web server dan perbandingan kecepatan berdasarkan jenis transaksinya. Berikut adalah gambar topologi implementasi pengujian kinerja web service.



Gambar 1 Topologi instalasi server

Konfigurasi pengujian dilakukan dengan mengukur nilai latency masing masing transaksi dengan beban data bertingkat dari data satu record hingga 50 record data. Hasil data akan dikoleksi berdasarkan jenis server dan jenis transaksi. Berikut tabulasi jumlah pengujian yang akan dilakukan

No	SERVER	KLIEN	JENIS TRANSAKSI
1	lis 7	apache	4
2	lis 7	nginx	4
3	apache	lis 7	4
4	apache	nginx	4
5	nginx	apache	4
6	nginx	lis 7	4
			<b>24</b>

Masing masing jenis transaksi akan dianalisis secara deskriptif dan diambil kesimpulan tentang kelebihan dan kekurangan masing masing transaksi terhadap jenis web server.

### HASIL DAN PEMBAHASAN

#### a. Gambaran Sistem

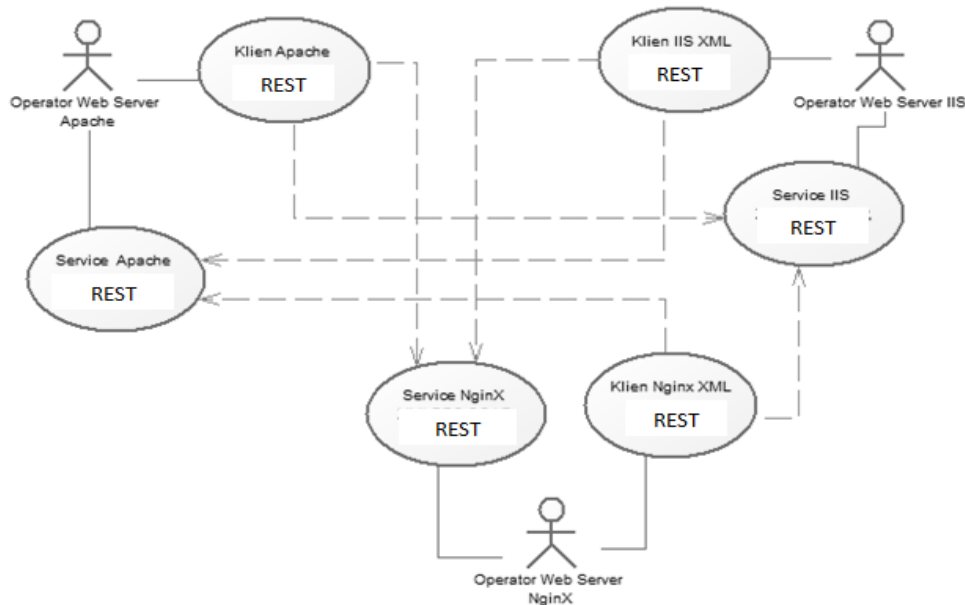
Desain aplikasi yang dikembangkan pada sistem penilaian kinerja web service. data input berupa data array yang diciptakan pada sisi klien dan dikirimkan kepada server dalam format XML melalui layanan web yang disediakan server. Setelah server mengeksekusi permintaan klien server akan memberikan data respon sebagai jawaban kepada klien dalam format XML dan mengirimkan data respon kepada klien. Output yang dihasilkan adalah data latency yang diterima oleh klien (1) Hasil analisis kebutuhan sistem yang diperlukan adalah Setiap aplikasi berfungsi sebagai klien dan server. (2) Sebagai server masing masing aplikasi memiliki sebuah layanan web berupa layanan yang dapat melakukan penambahan, perubahan, dan pemilihan data pada sistem basis data server. (3) Sebagai server layanan web yang dimiliki dapat diakses menggunakan metode REST. (4) Sebagai klien masing masing aplikasi dapat melakukan transaksi penambahan, perubahan, penghapusan data dengan memanfaatkan layanan web yang dimiliki aplikasi lainnya. (5) Sebagai klien pada setiap transaksi dapat menghasilkan keluaran berupa nilai latency dan waktu eksekusi serta data lain yang diminta. (6) Setiap aplikasi menampilkan data hasil pengujian berdasarkan beban data uji yang direncanakan.

#### b. Desain Model Sistem

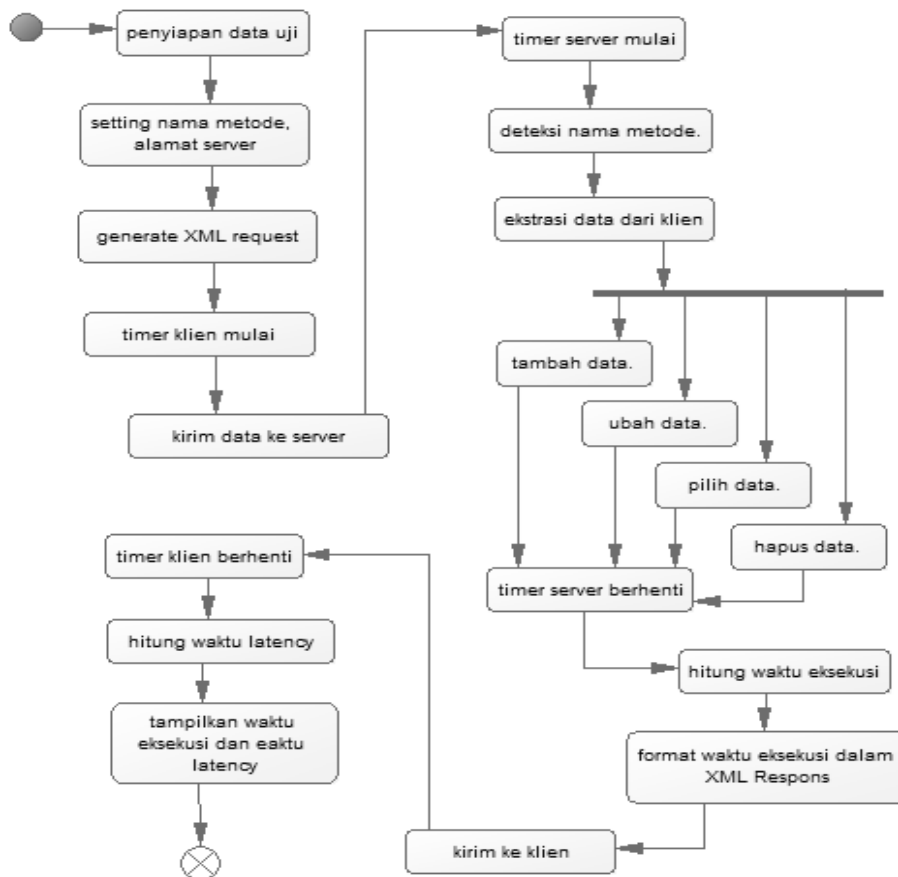
Use case model kinerja web service menggambarkan fungsi-fungsi dalam sebuah skema integrasi data yang mengimplementasikan metode REST. Pada gambar 2 dapat dilihat bahwa masing masing aktor memiliki sebuah layanan web yang melayani permintaan data bagi aplikasi lainnya

sehingga aplikasi tersebut berperan sebagai *provider*. Selain itu masing masing aktor juga memiliki modul untuk meminta data kepada aplikasi lain yang menunjukkan peran operator sebagai

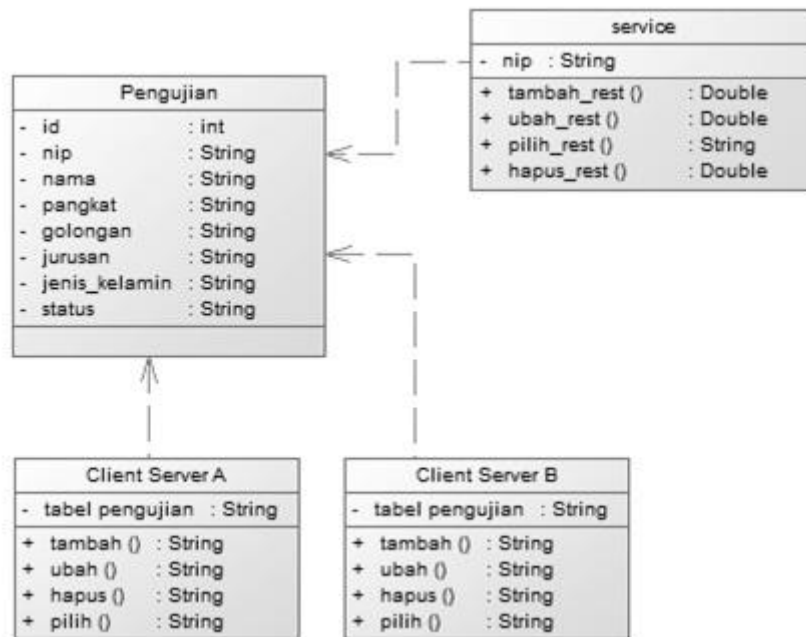
klien. Pada gambar tersebut model direncanakan agar tiga aplikasi tersebut menerapkan metode REST.



Gambar 2. Desain Use Case Model Pengujian Kinerja



Gambar 3. Gambar activity diagram model pengujian web service



Gambar 4. Class Diagram model Penguajian kinerja web service

Gambar 2 dan gambar 3 merupakan gambar diagram pemodelan yang direncanakan sebagai model aplikasi yang dikembangkan dalam penelitian ini. Pada gambar 2 diperlihatkan bahwa aplikasi ini akan membangun beberapa service yang akan terpasang pada masing masing server. Sehingga setiap server akan berfungsi sebagai klien untuk mengukur kedua server lain. Sedangkan transaksi yang diukur adalah transaksi penambahan, perubahan, penghapusan dan pemilihan data. Data yang akan ditransaksikan didesain memiliki beban data yang bertingkat. Gambaran aktivitas pengujian terlihat pada gambar 3.

### c. Implementasi sistem

Hasil pengembangan sistem berupa tiga buah aplikasi yang terpasang pada tiga server dimana setiap aplikasi memiliki menu yang dapat mengukur kinerja web service dua web server lainnya.

implementasi hasil pengembangan model untuk metode rest dilakukan dengan membuat 2 modul yaitu modul klien dan modul server. Modul klien akan menjadi modul yang meminta data atau mengeksekusi data pada kedua server yang lainnya. Sedangkan modul server adalah modul service yang dimiliki server untuk melayani permintaan klien. Berikut kode program yang dikembangkan untuk mengukur nilai latency untuk satu beban data. Metode rest yang dikembangkan dalam penelitian ini menggunakan pustaka cURL. cURL adalah perangkat lunak yang digunakan untuk mentransfer data dari dan ke server. cURL memiliki fitur yang lengkap dibanding perangkat lainnya. cURL telah diintegrasikan secara standar

pada berbagai produk web server. Berikut contoh kode program pada klien

```
$url = 'http://rpcserver.loc/mod_uji_rest.php/tambah';
$xml = "<data><member>
  <nip>2013\".$count.\"</nip>
  <nama>Client Apache</nama>
  <pangkat>Data Ke \".$count.\"</pangkat>
  <golongan>Jumlah Rec\".$count.\"</golongan>
  <jurusan>SKI UB</jurusan>
  <kelamin>L</kelamin>
  <status>REST</status>
</member></data>";

print "request <pre>\".htmlentities($xml).\"</pre>";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
$response = curl_exec($ch);
curl_close($ch);
echo "Hasil Feedback <pre>\".$response.\"</pre>";
```

Metode rest memerlukan sebuah string dengan format xml untuk dikirimkan melalui putaka cURL. Variabel string tersebut akan menjadi data yang dimasukkan dalam atribut cURL kemudian dikirimkan kepada server yang dituju. Keluaran yang ditampilkan pada sisi klien terlihat pada hasil permintaan yang dihasilkan sebagai berikut

```
request
    <data><member>
        <nip>20131</nip>
        <nama>Client Apache</nama>
        <pangkat>Data Ke 1</pangkat>
        <golongan>Jumlah Rec 1</golongan>
        <jurusan>SKI UB</jurusan>
        <kelamin>L</kelamin>
        <status>REST</status>
    </member></data>
```

**Hasil Feedback**

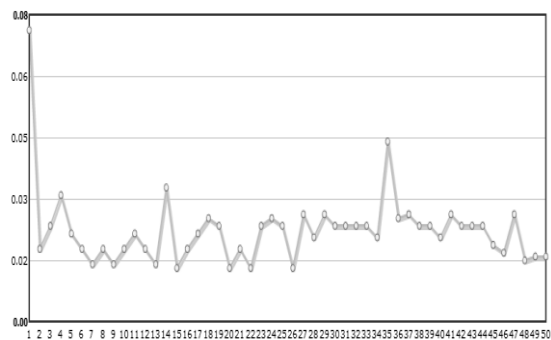
Successfully Save

Server akan menerima data permintaan tersebut dalam bentuk file xml. Langkah berikutnya server melalui fungsi simple xml membaca struktur data dari klien tersebut dan kemudian menyimpan data tersebut dalam basis data lokal pada server. Server kemudian membuat data string dengan format xml sebagai data respon untuk klien. berikut kode program pada sisi klien

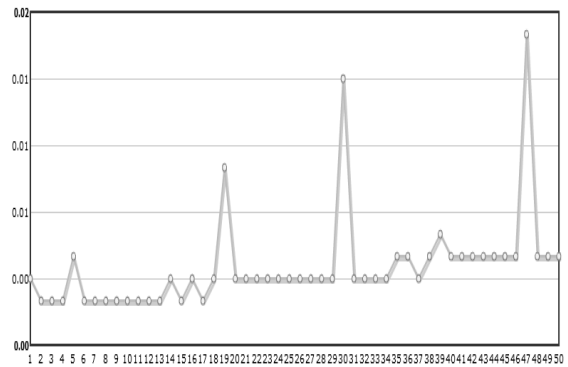
```
//header untuk format XML
header('Content-Type: text/xml; charset=ISO-8859-1');
$con = mysql_connect("localhost:3307","root","admin");
mysql_select_db("db_uji_performance",$con);
//Cek elemen path
$path = $_SERVER[PATH_INFO];
if($path != null) {
    $path_params = split("/", $path);
}
//METODE REQUEST untuk POST
if($_SERVER['REQUEST_METHOD'] == 'POST') {
    if($path_params[1] == 'tambah') {
        $input = file_get_contents("php://input");
        $xml = simplexml_load_string($input);
        $query = "INSERT INTO tm_peneliti(nip,nama,pangkat,golongan,jurusan,kelamin,status) values";
        $sum = 0;
        foreach($xml->member as $member) {
            $query = $query."('$member->nip','$member->nama','$member->pangkat',
                '$member->golongan','$member->jurusan','$member->kelamin','$member->status')";
            $sum++;
            if($sum < count($xml->member)) { $query = $query.','; }
        }
        $result = mysql_query($query) or die('Query failed: '.mysql_error());
        echo "<data>";
        echo "<member>";
        echo "<key>Successfully Save</key>";
        echo "</member>";
        echo "</data>";
        mysql_free_result($result);
    }
}
```

**d. Hasil Pengujian**

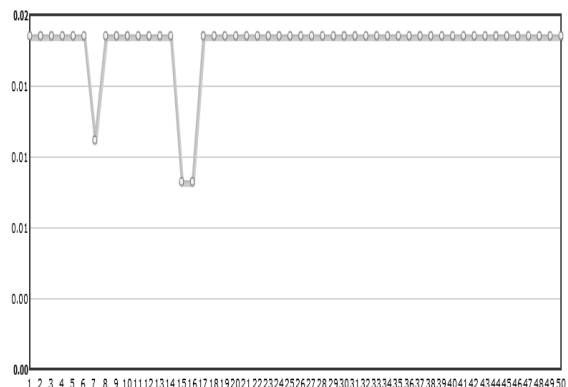
Hasil pengujian dalam aplikasi dapat ditampilkan dalam bentuk grafik latency dari permintaan transaksi dengan beban data 1 set data hingga 50 set data.berikut beberapa hasil grafik pengujian latency.



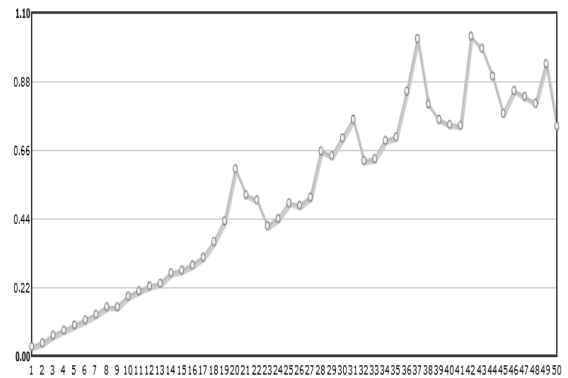
Gambar 5 : Penambahan data pada server iis



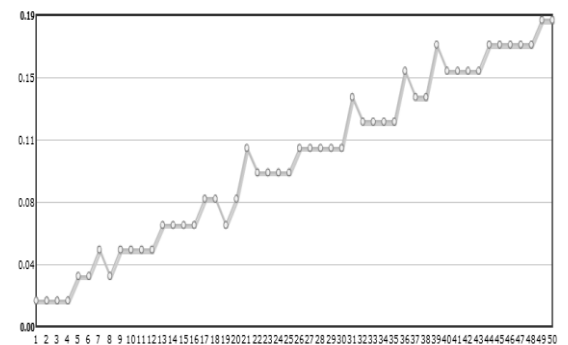
Gambar 6 : Penambahan data pada server nginx



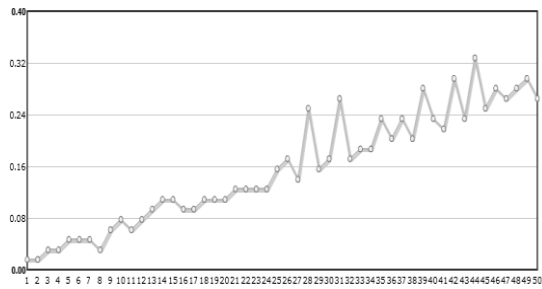
Gambar 7 : Penambahan data pada server apache



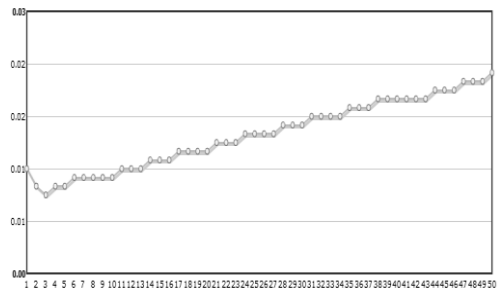
Gambar 8 : Perubahan data pada server iis



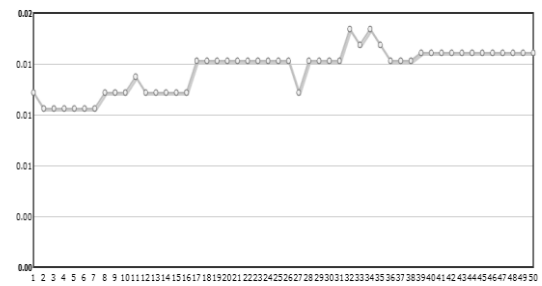
Gambar 9 : Perubahan data pada server nginx



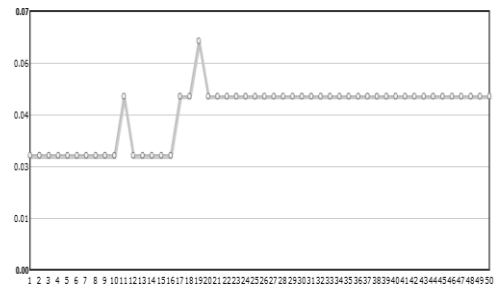
Gambar 10 : Perubahan data pada server apache



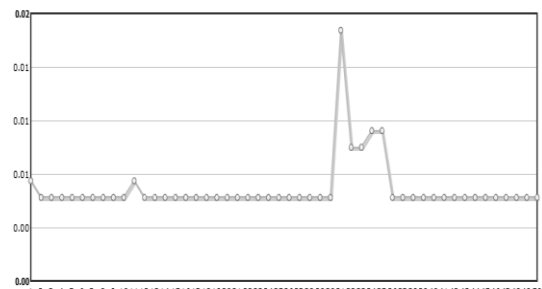
Gambar 15. Pemilihan data pada server nginx



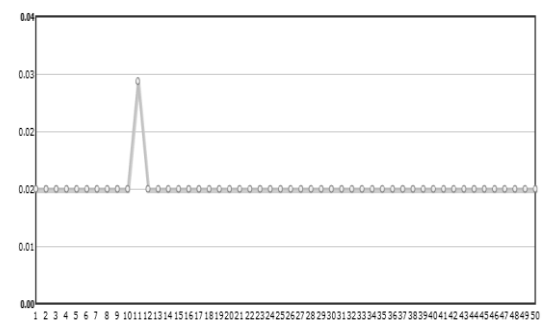
Gambar 11 : Penghapusan data pada server iis



Gambar 16. Pemilihan data pada server apache



Gambar 12 : Penghapusan data pada server nginx



Gambar 13 : Penghapusan data pada server apache



Gambar 14 : Pemilihan data pada server iis

Berdasarkan gambar grafik diatas dat diketahui bahwa pengaruh beban data terhadap lamanya latency tidak selalu linier akan tetapi memiliki karakteristik semakin besar data yang ditransaksikan semakin besar waktu yang diperlukan berikut hasil tabel nilai latency yang dikoleksi.

**Descriptive Statistics**

	N	Min	Max	Mean
nginx-apache-rest	50	.002	.016	.00380
apache-nginx-rest	50	.009	.076	.01384
apache-iis-rest	50	.009	.016	.01562
nginx-iis-rest	50	.001	.031	.01870
iis-nginx-rest	50	.014	.047	.02238
iis-apache-rest	50	.014	.047	.02360
Valid N (listwise)	50			

Tabel 2 : Nilai Latency penambahan data

**Descriptive Statistics**

	N	Min	Max	Mean
nginx-apache-rest	50	.006	.188	.09472
nginx-iis-rest	50	.016	.187	.10144
apache-nginx-rest	50	.016	.261	.10812
apache-iis-rest	50	.016	.328	.15666
iis-nginx-rest	50	.033	1.041	.48192
iis-apache-rest	50	.031	1.025	.51660
Valid N (listwise)	50			

Tabel 3 : Nilai Latency Perubahan data

**Descriptive Statistics**

	N	Min	Max	Mean
nginx-apache-rest	50	.009	.023	.01596
nginx-iis-rest	50	.016	.031	.02620
apache-nginx-rest	50	.031	.042	.03848
apache-iis-rest	50	.031	.062	.04250
iis-nginx-rest	50	.047	.086	.07200
iis-apache-rest	50	.071	.088	.07550
Valid N (listwise)	50			

Tabel 4 : Nilai Latency pemilihan data

**Descriptive Statistics**

	N	Min	Max	Mean
nginx-apache-rest	50	.005	.015	.00552
iis-nginx-rest	50	.009	.015	.01235
iis-apache-rest	50	.010	.015	.01246
nginx-iis-rest	50	.001	.031	.01450
apache-iis-rest	50	.016	.031	.01630
apache-nginx-rest	50	.016	.031	.02440
Valid N (listwise)	50			

Tabel 5 : Nilai Latency penghapusan data

Berdasarkan data nilai latency yang ditunjukkan pada tabel 2 hingga tabel 5 dapat disimpulkan bahwa web server nginx memiliki kecepatan eksekusi lebih baik dalam melayani permintaan klien hal ini terlihat dari semua jenis transaksi web server nginx memiliki kecepatan terbaik.

**KESIMPULAN DAN SARAN**

**Kesimpulan**

Metode Rest dapat diimplementasikan pada semua jenis arsitektur web server. Desain sumberdaya informasi yang diakses melalui URL dapat dibuat sesuai pemilik service dan klien wajib mengikuti format ketentuan desain service yang dibuat. Web server NginX merupakan web server yang paling tepat digunakan untuk teknologi web service terutama metode REST.

**Saran**

Pengukuran akan lebih ideal jika parameter pengukuran ditambahkan throughput dan dilakukan pada jaringan yang memiliki jumlah route lebih banyak.

**DAFTAR PUSTAKA**

Adina Ploscar. 2012. *XML-RPC vs. SOAP vs. REST web services in Java uniform using WS Wrapper*, International Journal of computer Issue 4, Volume 6, 2012

Eko Budi Cahyono. 2007. *Model dan Teknologi Komputasi Terdistribusi dengan platform Web Service*, Publikasi Ilmiah Universitas Muhammadiyah Malang.

Ethan Cerami .2002. *Web Service Essential First Edition* ISBN: 0-596-00224-6, O'Reilly.

Matjaz B. Juric., Ramesh Loganathan., Poornacandra Sarang., dan Frank Jennings. 2007. *SOA Approach to Integration*, Packt Publishing, Birmingham, B27 6PA, UK

Kuyoro Shade O. 2012. *Quality of Service (Qos) Issues in Web Services*, IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.1, January 2012

Mohamad Ibrahim Ladan., 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

Toyotaro Suzumaru .2008. *Performance Comparison of Web Service Engines in PHP, Java, and C*, IEEE International Conference on Web Services

Xiaoming cai, josefine Ottoson. 2009. *Performance & Implementation Best Practices for Information Passing Through Web Services*, SSN: 1651-4769 University of Gothenburg Department of Applied Information Technology