

# ANALISIS PERBANDINGAN *RESPONSE TIME* STATISTICS JAX-WS DAN RESTFUL WEB SERVICE

Reinhard Komansilan<sup>1)</sup> Eko Sedyono<sup>2)</sup>

<sup>1)2)</sup> Teknik Informatika Universitas “Universitas Kristen Satya Wacana”  
 Jl. Diponegoro 52-60 Salatiga - Indonesia 50711  
 email : <sup>1)</sup>deadcaster@yahoo.com, <sup>2)</sup>ekosed1@yahoo.com

## ABSTRACT

*Analysis of response time statistics comparison between SOAP-based web service which is JAX-WS and RESTful-based, through the implementation of the web-based applications of university academic system, using ServiceMon as the software that generates the response time statistics and availability value to measure the ability of the system to process  $\pm 100$ ,  $\pm 500$ , and  $\pm 1000$  request given in every 500ms frequency. Stability test was also carried out by measuring the ability of the system to handle  $\pm 5000$  request. The results showed a significant difference through the T test using SPSS 20 of the average response time of each web service with JAX-WS as a web service that has an average time to process each request given because the client application are uploaded to the web server is lighter than RESTful client applications. Stability test at  $\pm 5000$  shows the stability of each system with an average time of 500ms to 600ms still be among 100 and 1000ms normal time, but the availability value increased to  $> 6000$ ms with a percentage of 99.99% of the total requests tested.*

## Key words

*JAX-WS, Response Time, RESTful, ServiceMon, Web Service*

## 1. Pendahuluan

*Web* saat ini tidak hanya berisi informasi, tetapi juga dipakai untuk sarana komunikasi bisnis, media sosial, dan pendidikan. Untuk melakukan fungsinya sebagai penyedia informasi dan kegiatan lainnya, *web* juga harus didukung dengan sistem terdistribusi yang baik sehingga pengguna dapat merasakan kemudahan dan manfaat dari penggunaan *web* itu sendiri. Salah satu teknologi *web* yang digunakan dalam sistem terdistribusi adalah *web service*.

Teknologi sistem terdistribusi sebelumnya seperti RMI, CORBA, dan DCOM pada awalnya digunakan untuk membuat aplikasi *client server* dalam intranet karena

*platform* dan teknologi yang digunakan dari sisi *client* maupun *server* adalah sama dan cocok antara satu dengan yang lain. Dengan demikian, digunakanlah *web service* sebagai standar untuk mendistribusikan sistem lebih luas melalui internet. Aplikasi *client* tidak harus memiliki *platform* yang sama dengan aplikasi *server* karena standar dari *web service* inilah yang akan memungkinkan komunikasi antar *client server* terjadi. Oleh sebab itu *web service* tidak bergantung pada *platform* tertentu dan bebas digunakan oleh aplikasi *web* apapun.

Terdapat dua jenis *web service*, yaitu berdasarkan prinsip SOAP dan prinsip REST. Pada *web service* berbasis SOAP, XML digunakan untuk mendefinisikan *simple object access protocol*(SOAP) yaitu protokol yang digunakan oleh data dari *client* untuk dapat terhubung dengan *server*. Sedangkan RESTful *web service* adalah membangun *web service* dalam bentuk *representational state transfer*(REST), yaitu data dari *client* memiliki *uniform resource identifier*(URI) unik yang dapat terhubung dengan *server* menggunakan *web method* sederhana seperti HTTP-GET, POST, dan PUT. Keputusan untuk memilih salah satu konsep dari kedua konsep di atas, merupakan tahap awal dalam merancang *web service*, sehingga penting untuk memahami keunggulan dan kelemahan dari kedua konsep tersebut. SOAP dan RESTful *web service* memiliki filosofi yang berbeda satu sama lain. SOAP adalah protokol yang digunakan untuk aplikasi terdistribusi berbasis XML. Sedangkan RESTful *web service* menggunakan perintah HTTP seperti GET, PUT, POST, DELETE, dan lainnya. Sehingga untuk dapat memutuskan *web service* mana yang akan dipakai sesuai kebutuhannya, maka perlu dilakukan perbandingan karena masing-masing *web service* memiliki kelebihan dan kelemahan dalam beberapa aspek seperti fleksibilitas dan kemudahan, penggunaan *bandwidth*, keamanan, *type handling*, kompleksitas di sisi *client-server*, dan *chaching*[3].

Dalam penelitian ini, dipilih JAX-WS sebagai teknologi yang memanfaatkan protokol SOAP untuk

dianalisis perbandingannya dengan *web service* yang menggunakan arsitektur REST. *Response time statistic* digunakan sebagai indikator dalam penelitian ini untuk membuktikan dan membandingkan kemampuan *response time*, dan stabilitas dari kedua objek penelitian. Data statistik untuk *response time* diambil dengan menggunakan *software* ServiceMon pada masing-masing objek penelitian. Analisis perbandingan kedua teknologi tersebut, diperlukan untuk menjadi acuan khususnya kepada para developer yang baru mengenal *web service* agar terbantu dalam menyediakan dan membangun suatu *web service*, menciptakan *web service* yang lebih efisien dan mudah untuk digunakan, serta mengembangkan konsep teknologi *web service* yang baru sebagai perkembangan di masa depan.

## 2. Tinjauan Pustaka

### 2.1 Penelitian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini dikaji hasil penelitian terdahulu:

1. Jurnal Penelitian dari Snehal Mumbaikar dan Puja Padiya “*Web Services Based On SOAP and REST Principles*”[2]

Tujuan dari artikel ini adalah merancang penelitian pada SOAP dan RESTful *web service* untuk menunjukkan bahwa RESTful memiliki kemampuan yang lebih baik. Dengan memberikan gambaran dari implementasi SOAP dan REST, dengan konferensi multimedia dan layanan komputasi mobil sebagai contoh implementasi.

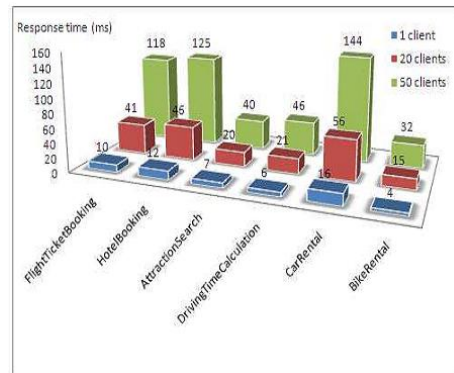
Hasil dari analisis yang telah dilakukan ,disimpulkan bahwa RESTful *web service* adalah alternatif yang lebih baik untuk *web service* berbasis SOAP. *Web service* berbasis SOAP menghasilkan lalu lintas jaringan yang cukup besar, latensi tinggi, dan ukuran *message* yang besar. Kelemahan-kelemahan tersebut tidak didapat pada RESTful. RESTful *web service* memiliki kemampuan yang lebih baik dari SOAP dalam jaringan kabel maupun nirkabel. RESTful *web service* ringan, mudah, dan flexible.[2]

Artikel diatas menggunakan REST dengan tipe data yang berbeda sesuai dengan aplikasinya yaitu konferensi multimedia dengan tipe data yang besar untuk menyimpan suara dan video, sedangkan tipe data dengan ukuran yang lebih kecil seperti teks tidak dilakukan perbandingan sehingga hipotesa yang dapat diambil untuk penelitian ini bahwa RESTful memiliki kemampuan lebih baik dalam mengolah tipe data yang lebih besar. dari SOAP yang pada

penelitian ini menggunakan JAX-WS sebagai *web service* berbasis SOAP. Namun dari sisi implementasi ketika *user* dapat mengakses *client* dari masing-masing *web service* setelah *deployment* ke *web server*, telah dirancang skenario perbandingan yang berbeda, yaitu dengan memanfaatkan fungsi komunikasi antara *user* dengan *client* dari masing-masing *web service* yang telah dibahas sebelumnya.

2. Jurnal Penelitian Maha Driss, Yaassine Jamouss dan Henda Hajjami Ben Ghezala “*QoS Testing of Service-Based Applications*”[4]

Dalam penelitian ini, peneliti memperkenalkan model pendekatan *discrete-events* untuk aplikasi berbasis *service*. Pendekatan ini berorientasi pada evaluasi *QoS* melalui simulasi. Dengan menggunakan aplikasi simulasi jaringan NS-2[5] dengan menambahkan ekstensi hirarki kelas dari C++ untuk dapat melakukan implementasi protokol HTTP, SMTP, dan SOAP. Penelitian ini berfokus pada evaluasi *response time* dari beberapa *web service* seperti yang dapat dilihat pada grafik di bawah ini:



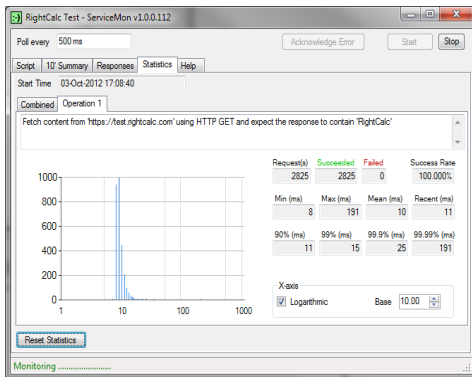
Gambar 1. Hasil *response time* dari simulasi dari beberapa *web service*

Perlu diketahui bahwa objek pada penelitian diatas menggunakan *web service* berbasis SOAP[4], karena tujuan dari penelitian tersebut adalah implementasi dari model simulasi untuk mengukur *response time* dari masing-masing fungsi *web service*. Pada penelitian ini digunakan *web service* dengan model arsitektur berbeda selain SOAP, sebagai pembanding bukan hanya pada tingkat implementasi namun juga pada desain arsitektur teknologi *web service* itu sendiri.

3. Artikel yang dipublikasikan oleh Stuart Wheelwright “*Measuring and Monitoring WCF Web Service*”[7]

Latar belakang masalah dari artikel tersebut adalah perlunya suatu cara atau alat yang dapat digunakan sebagai alat untuk mengukur dan menguji sistem yang sedang dijalankan apakah dapat berjalan dengan stabil dalam berbagai aktivitas jaringan dan jumlah *request* yang besar.

ServiceMon dikembangkan sebagai alat monitoring *website* dan *web service* serta sebagai alat ukur dengan menggunakan uji beban untuk mendapatkan data statistik *response time* secara detail untuk kebutuhan analisis dari sistem yang telah dirancang. Contoh tampilan ServiceMon yang digunakan untuk menghasilkan *response time statistik* dapat dilihat pada gambar dibawah ini:



Gambar 2. Tampilan hasil *response time statistik* dari ServiceMon

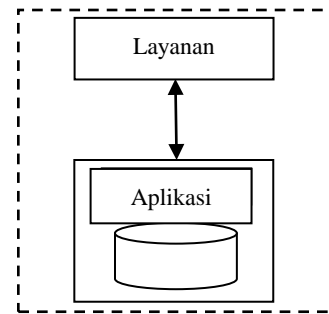
## 2.2 Metode Penelitian

JAX-WS dan RESTful *Web Service* akan dibandingkan berdasarkan indikator yang terdapat pada ServiceMon yaitu *response time statistics*, *stabilty*, dan *availability*. Untuk dapat membandingkannya, maka perlu dibuat aplikasi sederhana yaitu aplikasi *web service* layanan akademik dari masing-masing teknologi. Hasil perbandingan dari kedua aplikasi ini dapat menjadi acuan untuk mengembangkan aplikasi *web service* sesuai dengan kebutuhan sistem dan pengguna.

### 1. Perancangan Sistem

Sasaran pengguna fungsi-fungsi dalam *web services* adalah pengembang aplikasi, bukan *end-user*. Oleh karena itu yang harus disediakan oleh penyedia *web services* adalah fungsi-fungsi yang dibutuhkan oleh pengembang aplikasi dalam domain yang disediakan. Fungsi-fungsi yang disediakan oleh penyedia *web services* tidak harus lengkap, karena penyedia *web services* yang lain dapat menyediakan fungsi-fungsi lain yang pada akhirnya membentuk sistem yang lengkap. Dengan begitu kolaborasi (gotong-royong) antar pengembang aplikasi dapat berlangsung dengan baik [8].

Sistem yang akan dirancang sebagai objek dalam penelitian ini adalah sistem informasi layanan akademik universitas berbasis *web* berupa *prototype* yang memuat dan menjalankan fungsi dari masing-masing teknologi *web service* yang akan dibandingkan yaitu JAX-WS dan RESTful *web service*. Konsep pengembangan ini ditunjukkan pada Gambar 3.

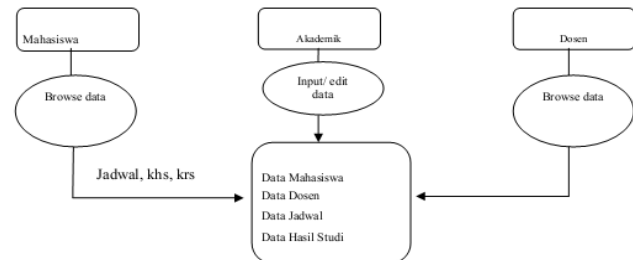


Gambar 3. Konsep Pengembangan Sistem Informasi Universitas

### 2. Rancangan Layanan Akademik

Untuk layanan akademik, maka rancangan yang dibuat mempunyai fungsionalitas sebagai berikut:

- Bagi mahasiswa, yaitu dapat melakukan browsing informasi jadwal kuliah, dosen pengampu, data studinya, dan melakukan proses registrasi mata kuliah.
- Bagi dosen, yaitu dapat melakukan browsing data mahasiswa dan jadwal kuliah.
- Bagi bagian akademik, yaitu dapat melakukan pengisian, pengeditan, dan pengolahan data.



Gambar 4. Rancangan Layanan Akademik

Sesuai dengan kebutuhan di atas, maka dapat dibuat aplikasi yang dapat diimplementasikan dengan memanfaatkan fungsi-fungsi tersebut. Aplikasi yang akan dibuat adalah aplikasi berbasis *web* dengan menggunakan Java. Aplikasi ini ditujukan untuk mahasiswa dan bagian akademik. Fasilitas yang disediakan untuk mahasiswa adalah *browse* data jadwal, ruang, dosen dan melihat data KHS, serta melakukan KRS. Cara kerja secara umum adalah sebagai berikut:

- *Web Service* layanan akademik yang dibuat dan diletakkan pada server.
- Aplikasi *client* berkomunikasi dengan layanan akademik dengan cara memanggil fungsi yang tersedia pada *Web Services* sambil mengirimkan parameter-parameter yang dibutuhkan.
- Pemrosesan data dalam hal ini dilakukan di server, kemudian hasil yang diperoleh dikirimkan kembali ke aplikasi *client*.

### 3. Analisis Perbandingan

Untuk melakukan analisis perbandingan dari kedua teknologi di atas, akan dilakukan pengujian terhadap masing-masing teknologi *web service*, yaitu dengan melakukan *deployment* masing-masing *web service* ke *web server*, sehingga *user* dapat mengakses aplikasi melalui *client* dari masing-masing *web service*. Interaksi(*request*) dari *user* dengan *client* inilah yang akan dimonitor menggunakan *software* monitoring ServiceMon[3] yang akan menghasilkan *response time statistic*. Data yang dihasilkan berupa grafik distribusi model *logarithmic x-axis* yang menggambarkan detail proses yang berlangsung sesuai dengan frekuensi waktu yang ditentukan. Selain grafik, terdapat dua tabel yang menampilkan hasil statistik dari mayoritas *request* yang dikirim, diproses, dan diterima, serta tabel yang berguna untuk menggambarkan kurva secara kuantitatif dengan menspesifikasikan beberapa nilai *availability*[10] hingga 4 nilai 9 untuk mengukur tingkat ketersediaan(*availability*) sistem.

Data yang diperlukan untuk analisis perbandingan diperoleh dengan melakukan sebuah *request HTTP GET* sederhana pada fungsi *web service* untuk registrasi mata kuliah oleh mahasiswa dengan frekuensi 500ms karena menurut Nielsen[9], respon yang diterima dalam 100ms terasa instan, dalam 1000ms (1 detik) masih dapat diterima, tapi lebih dari 10.000 ms (10 detik) sudah cukup bagi *user* untuk kehilangan minat dan melakukan sesuatu yang lain. Jadi berdasarkan hal tersebut, digunakan frekuensi 500ms sebagai nilai tengah antara 100ms dan 1000ms.

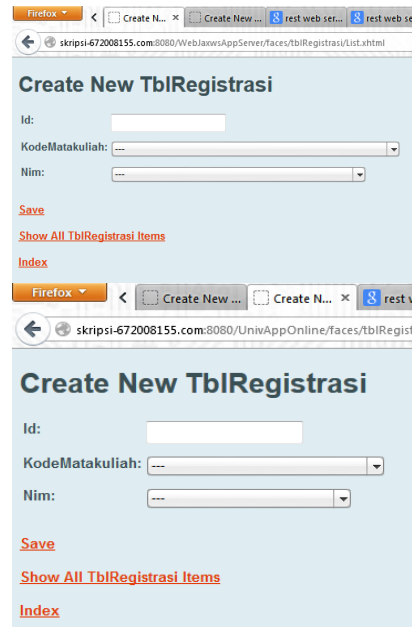
Data yang akan digunakan sebagai perbandingan *response time statistik* adalah nilai rata-rata dari setiap pengujian antara  $\pm 100$ ,  $\pm 500$ , dan  $\pm 1000$  *request* yang diproses, kemudian dilakukan uji T dengan SPSS untuk dapat menyimpulkan apakah terdapat perbedaan yang signifikan antara kedua objek pengujian. Selain hasil perbandingan dari rata-rata pengujian, data lainnya yang telah diperoleh akan dibandingkan dan disimpulkan secara deskriptif seperti hasil analisis stabilitas(*stability*) dan ketersediaan(*availability*) masing-masing *web service* ketika dijalankan.

## 3. Hasil dan Pembahasan

### 1. Web Service layanan Akademik Mahasiswa

Aplikasi *web service* yang dibangun diakses dari *web server* melalui domain skripsi-672008155.com. Keduanya memiliki fungsi yang sama, dengan demikian akan lebih mudah untuk melakukan perbandingan karena tidak ada perbedaan dari segi fungsionalitasnya. Antarmuka dari

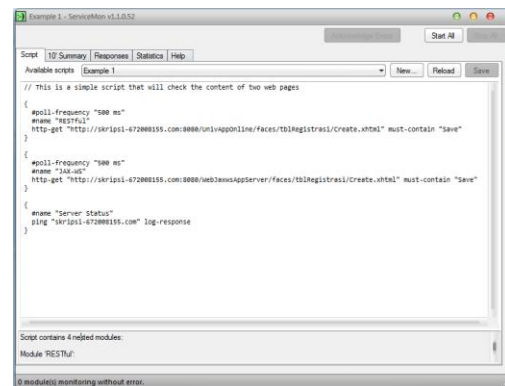
kedua aplikasi ini pun tidak berbeda karena, antarmuka tidak menjadi salah satu indikator dalam pengujian maupun dalam analisis perbandingan. Fungsi yang diuji dalam analisis perbandingan adalah registrasi mata kuliah oleh mahasiswa dari masing-masing *web service* seperti yang dapat dilihat pada gambar di bawah ini:



Gambar 5. Antarmuka fungsi registrasi mata kuliah dari JAX-WS(atas) dan RESTful(bawah)

### 2. Hasil Analisis Perbandingan

Setelah membangun kedua aplikasi *web service* dan melakukan pengujian terhadap seluruh fungsi yang terdapat pada masing-masing *web service*, maka dilakukan analisis dengan menggunakan perangkat lunak ServiceMon untuk menghasilkan data statistik *response time* dari masing-masing percobaan.



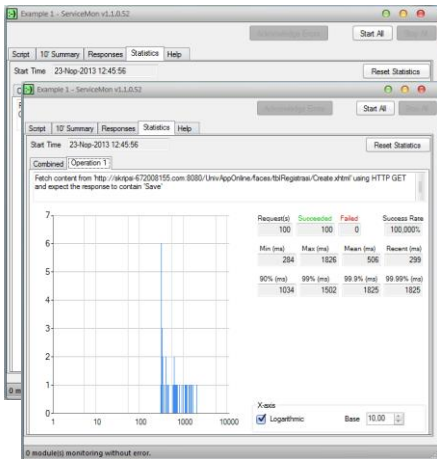
Gambar 6. Tampilan ServiceMon dan script yang digunakan untuk menjalankan monitoring

Pengujian diawali dengan menjalankan 2 *script* untuk melakukan request terhadap kedua web service dan 1 *script* sebagai kontrol jaringan. *Script* untuk melakukan request yang terdiri dari 3 bagian yaitu:

- Menentukan frekuensi waktu 500ms untuk setiap perintah yang akan dijalankan.
- Menentukan nama masing-masing sampel pada panel tampilan dengan nama RESTful dan JAX-WS pada tools simulasi.
- Perintah *http-get* untuk melakukan request pada URL *web service* dengan fungsi registrasi mata kuliah.

Sedangkan bagian *script* lainnya berfungsi sebagai kontrol jaringan yang hanya terdiri dari nama panel dan ping untuk memastikan ketersediaan dan kestabilan jaringan ke server. Setelah *script* dijalankan, ServiceMon akan menghasilkan grafik dan tabel, diambil berdasarkan setiap jumlah *request* yang sudah ditentukan sebelumnya seperti pada gambar dan penjelasan di bawah ini:

1. Hasil pengujian dengan  $\pm 100$  request



Gambar 7 dan 8. Grafik dan tabel *response time statistic* dengan  $\pm 100$  request dari JAX-WS dan RESTful *web service*

Grafik di atas menunjukkan detail dari mayoritas *request* dikirim, diproses dan diterima dengan frekuensi 500ms dan dengan total  $\pm 100$  request. Pada grafik pertama menghasilkan statistik *response time* dari JAX-WS dengan rata-rata *request* diproses dalam 468ms, minimum 284ms dan maksimum 1891ms. Secara detail dapat di deskripsikan dengan melihat nilai *availability* pada tabel di samping grafik: 90% *request* diproses dalam 904ms, 99% dalam 1749ms, 99,9% dan 99,99% dalam 1890ms.

Pada grafik yang kedua menghasilkan statistik *response time* dari RESTful *web service* dengan rata-rata *request* diproses dalam 506ms, minimum 284ms, dan maksimum 1826ms. nilai *high availability* pada tabel di samping grafik: 90% *request* diproses dalam 1034ms, 99% dalam 1502ms, 99,9% dan 99,99% dalam 1825ms. Selanjutnya cara ini dilakukan masing-masing untuk 500 dan 1000 request, dan hasilnya dikelompokkan pada Tabel 1, untuk dilakukan analisis uji T.

2. Uji T

Uji T dilakukan untuk melihat apakah ada perbedaan yang signifikan antara nilai rata-rata kedua sampel yang dihasilkan dari ketiga pengujian di atas. Secara keseluruhan nilai-nilai tersebut dapat dilihat dari tabel di bawah ini:

Tabel 1. Nilai rata-rata dari setiap pengujian dari kedua sampel

Jenis sampel	Rata-rata waktu(ms)		
	$\pm 100$ request	$\pm 500$ request	$\pm 1000$ request
JAX-WS	468	545	894
RESTful	506	690	1058

Dengan menggunakan *tools* statistik SPSS 20, hasil dari uji T diperoleh sebagai berikut:

Tabel 2. Hasil dari uji T menggunakan *tools* SPSS 20

		Group Statistics			
Web Service		N	Mean	Std. Deviation	Std. Error Mean
Response	JAX-WS	3	635,67	227,012	131,065
Time	RESTful	3	751,33	281,065	162,273



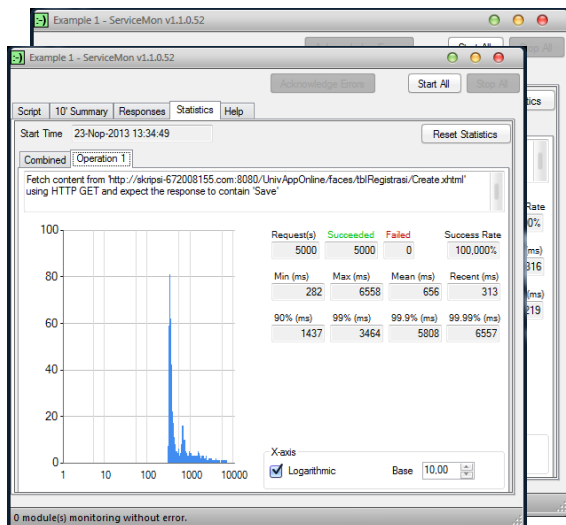
		Independent Samples Test								
		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
								Lower	Upper	
Response Time	Equal variances assumed	,133	,733	-,555	4	,609	-115,667	208,592	-694,810	463,477
	Equal variances not assumed			-,555	3,830	,610	-115,667	208,592	-705,062	473,729

Interpretasi hasil uji anova:

- Uji Levene digunakan untuk mengetahui homogenitas sampel. Jika sig > 0.05 maka sampel homogen
- Hasil Uji T menunjukkan sig 0.609 atau kurang dari 0.05, maka dapat disimpulkan ada perbedaan rata-rata waktu yang signifikan antara penggunaan  $\pm 100$ ,  $\pm 500$ , dan  $\pm 1000$  request dari kedua web service.

### 3. Stabilitas

Selain analisis perbandingan *response time* dan *availability* yang telah dijelaskan sebelumnya, hasil dari *response time statistik* juga dapat digunakan untuk analisis stabilitas sistem. Pengujian untuk analisis *response time* dilakukan pada  $\pm 100$ ,  $\pm 500$ , dan  $\pm 1000$  request. Sedangkan untuk analisis stabilitas ini, pengujian dilakukan hingga  $\pm 5000$  request. Hasil dari pengujian ini dapat dilihat pada gambar di bawah ini:



Gambar 9 dan 10. Grafik dan tabel *response time statistic* dengan  $\pm 5000$  request dari JAX-WS dan RESTful web service

Pengujian dengan  $\pm 5000$  request menghasilkan rata-rata *response time* untuk JAX-WS 559ms dan RESTful 656ms. Menunjukkan bahwa pada  $\pm 5000$  request, rata-rata waktu yang dibutuhkan sistem masih pada angka 100 hingga 1000ms yaitu menurut Nielsen, 2009[9], ada di

antara waktu instan dan waktu normal yang dirasakan oleh manusia. Namun pada nilai maksimum *response time*, keduanya menunjukkan nilai >6000ms atau lebih dari 6 detik, dengan presentase 99,99%(nilai *availability*) diproses keduanya dalam waktu lebih dari 6 detik.

Dengan demikian hasil dari uji stabilitas sederhana di atas, dapat dideskripsikan bahwa JAX-WS dan RESTful masih stabil pada 5000 request dengan rata-rata waktu kurang dari 1 detik, tetapi 99,99% dari request di proses dalam waktu lebih dari 6 detik. Sehingga demikian, jika jumlah request semakin meningkat, nilai presentase *availability* akan semakin meningkat, dan jika rata-rata *response time* meningkat hingga melebihi 10 detik, proses dari sistem akan dianggap terlalu lama, dan cukup waktu untuk user dapat beralih dari penggunaan sistem.

Hasil dari penelitian ini tergantung dari jenis aplikasi yang dibuat, karena *performance* JAX-WS dan RESTful dipengaruhi oleh besar transaksi data yang dilakukan oleh aplikasi melalui jaringan, sehingga semakin besar transaksi data, semakin tinggi *response time* yang dihasilkan dan semakin tinggi pula nilai *availability* yang akan berpengaruh pada stabilitas sistem. Perlu diketahui bahwa untuk aplikasi berbasis web yang menggunakan komunikasi data antar *client server*, nilai *response time*, *availability*, dan stabilitas juga dipengaruhi oleh: jumlah pengguna, kecepatan transmisi, jenis transmisi, jenis perangkat keras, dan perangkat lunak yang digunakan.

Pada jenis aplikasi JAX-WS dan RESTful lainnya yang membutuhkan transaksi dengan ukuran data dan skala yang lebih besar, maka akan membutuhkan tunjangan sistem yang lebih mumpuni untuk mendapatkan hasil *response time*, *availability*, dan stabilitas yang sesuai dengan kebutuhan sistem dan pengguna.

#### 4. Kesimpulan

Kesimpulan yang dapat diperoleh dari penelitian di atas adalah sebagai berikut:

1. Program yang dibangun sebagai sampel perbandingan adalah *web service* registrasi mahasiswa, masing-masing dengan arsitektur yang berbeda yaitu JAX-WS dan RESTful *web service*.
2. Yang menjadi perbandingan dari kedua teknologi *web service* adalah hasil implementasi dari masing-masing *web service* yang telah dijalankan di *remote server*.
3. Analisis perbandingan dilakukan dengan menggunakan *software* ServiceMon untuk mendapatkan hasil *response time statistics* dari setiap sample pengujian.
4. Hasil *response time statistics* diperoleh dengan melakukan *request http-get* dengan frekuensi 500ms untuk setiap sample dengan jumlah masing-masing  $\pm 100$ ,  $\pm 500$ , dan  $\pm 1000$  *request*.
5. Dari hasil pengujian, diambil rata-rata *response time* setiap sample, kemudian dilakukan uji T dengan bantuan *software* SPSS 20, terbukti bahwa terdapat perbedaan yang signifikan antara hasil rata-rata kedua sample setelah pengujian.
6. Berdasarkan perbandingan nilai rata-rata *response time*, dapat dilihat bahwa JAX-WS Service dapat memproses *request* lebih cepat dari RESTful untuk setiap jumlah *request* yang diujikan pada kondisi *request* dilakukan setiap 500ms.
7. Pada uji stabilitas, masing-masing sistem dapat memproses  $\pm 5000$  *request* dengan rata-rata waktu yang cukup stabil ( $\pm 500$ -600ms), namun nilai *availability* meningkat hingga 99,99% *request* diproses dalam waktu hingga  $\pm 6000$ ms (6 detik).
8. Pada pengujian, terdapat beberapa faktor yang menentukan JAX-WS pada *response time statistic* dapat lebih unggul dari RESTful, antara lain:
  - Ukuran *client* JAX-WS ketika *deployment* ke *web server* lebih kecil dibandingkan dengan RESTful, sehingga jumlah *bandwidth* yang dibutuhkan untuk mengakses *client* lebih sedikit dan dapat menghasilkan *response time* yang lebih baik.
  - Kedua sistem dikembangkan dengan menggunakan EJB untuk mendapatkan struktur program yang lebih baik, dan kebanyakan kontainer EJB memiliki dukungan untuk *load balancing* dan *failover* dengan tingkat kehandalan yang tidak mudah dicapai dengan *web service*.

Adapun saran yang dapat disampaikan sebagai acuan untuk perkembangan penelitian ini yaitu:

1. Teknologi *web service* saat ini berkembang dengan pesat, sehingga perlu adanya kombinasi penelitian baik dari segi arsitektur, implementasi, dan pengujian untuk

- mendapatkan hasil perbandingan yang semakin akurat.
2. Metode ini masih perlu diuji untuk sistem dengan skalabilitas yang lebih tinggi.
3. Uji stabilitas masih dapat dilakukan dengan meningkatkan jumlah *request* untuk mendapatkan titik beban maksimal yang dapat dicapai oleh masing-masing sistem, namun pengujian yang terlalu agresif dapat dianggap sebagai serangan *denial of service* yang dapat mengakibatkan lumpuhnya kerja *web server*.
4. Penggunaan *response time statistik* juga sebaiknya digunakan pada aplikasi berbasis teknologi web lainnya.
5. Analisis yang dilakukan tidak terbatas hanya pada *response time* saja, tetapi juga dapat dilakukan pada elemen QoS lainnya.

#### REFERENSI

- [1] Wulandari, Lily., Wicaksana, I Wayan Simri., 2006, Toward Web Service, lily.staff.gunadarma.ac.id/Publications/.../fullPaper\_KOM MIT06.pdf. Diakses tanggal 20 Maret 2013
- [2] Mumbaikar, Snehal; Padiya, Puja; 2013, Web Services Based On SOAP and REST Principles, *International Journal of Scientific and Research Publications*, 3:1-4
- [3] Rizki, Aditya; 2012, Web Service SOAP vs REST, Mana yang Lebih Baik?, <http://www.adityarizki.net/2012/06/web-service-soap-vs-rest-mana-yang-lebih-baik/>. Diakses tanggal 22 November 2013.
- [4] Driss, Maha.; Yassine, Jamoussi; Ghezala, Henda Hajjami Ben; 2008, QoS Testing of Service-Based Application. Proceedings of the 3rd IEEE International Design and Test Workshop, 2008: 45-50
- [5] K. Fall and K. Varadhan. The ns manual. Online: <http://www.isi.edu/nsnam/ns/doc/ns-doc.pdf>
- [6] Driss, Maha., Yassine Jamoussi, Jean-Marc Jezeques, dkk, 2008, A Discrete-Events Simulation Approach for Evaluation of Service-Based Applications. ECOWS'08 : Proceedings of the 6th European Conference on Web Services, 2008
- [7] Wheelwright, Stuart, 2012, Article: Measuring and Monitoring WCF Web Service <https://www.rightcalc.com/blog/servicemon/articles-measuring-and-monitoring-wcf-web-service-performance/>. Diakses tanggal 8 Agustus 2013
- [8] Santosa, Budi, 2008, Analisa dan Perancangan Web Services Untuk Sistem Informasi Universitas, *Jurusan Teknik Informatika UPN "Veteran" Yogyakarta*.
- [9] Nielsen, Jacob; 2009, Powers of 10: Time Scales in User Experience, <http://www.nngroup.com/articles/powers-of-10-time-scales-in-ux/>. Diakses tanggal 13 Agustus 2013
- [10] No name; 2011, System Availability and Downtime, <http://www.activeserver.com/pdf/SystemAvailabilityandDowntime.pdf>, Diakses tanggal 27 November 2013