Uji Kinerja Sistem Web Service Pembayaran Mahasiswa Menggunakan Apache JMeter (Studi Kasus: Universitas AMIKOM Yogyakarta)

Kamarudin[1], Kusrini[2], Andi Sunyoto[3]
[1,2,3]Magister Teknik Informatika, Universitas AMIKOM Yogyakarta
[1]kamarudin@amikom.ac.id, [2]kusrini@amikom.ac.id, [3]andi@amikom.ac.id

INTISARI

Web Service sudah terbukti banyak diimplementasikan pada proses integrasi sistem. Ada dua metode web service yang sering digunakan untuk membangun sebuah sistem web service yaitu Simple Object Access Protocol (SOAP) dan Representational State Transfer (REST). Masing-masing metode ini memiliki spesifikasi dan algoritma yang berbeda dalam implementasinya.

Penelitian ini difokuskan untuk melakukan uji kinerja sistem web service yang dibangun menggunakan metode SOAP dan REST untuk mendapatkan informasi metode mana yang lebih baik kinerjanya sehingga bisa menjadi solusi integrasi sistem informasi pembayaran mahasiswa di Universitas AMIKOM Yogyakarta. Proses uji kinerja dilakukan dengan menggunakan tool Apache JMeter.

Dari hasil penelitian ini didapatlah Response Time dari sistem lama (SOAP) lebih cepat dari pada prototipe sistem baru (REST). Untuk parameter Received/Sent, prototipe sistem baru lebih baik dari sistem lama. Sedangkan untuk parameter Throughput, sistem lama sedikit lebih baik dari prototipe sistem baru.

Kata Kunci: *web service, soap, rest, throughput, response time, uji kinerja sistem*

ABSTRACT

Web Service has proven to be widely implemented in system integration process. There are two web service methods that are often used to build a web service system that is, Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). Each of these methods has different specifications and algorithms in its implementation.

This research is focused to test the performance of web service system built using SOAP and REST method to get information which method better performance. Performance test process is done by using Apache JMeter tool.

This research is focused to test the performance of web service system built using SOAP and REST method to get information which method better performance so that can be solution of integration for student payment information system at AMIKOM University Yogyakarta. Performance test process is done by using Apache JMeter tool.

From the results of this study obtained Response Time from the old system (SOAP) faster than the prototype new system (REST). For the Received / Sent parameter, the new system prototype is better than the old system. As for the throughput parameter, the old system is slightly better than the new system prototype.

Keywords: *web service, soap, rest, response time, throughput, performance testing*

I. PENDAHULUAN

Sistem informasi pembayaran mahasiswa di Universitas AMIKOM Yogyakarta sudah mulai dikembangkan sejak tahun 2004, hingga sampai saat ini berdasarkan informasi dari tim pengembang software Innovation Center Universitas AMIKOM Yogyakarta ada tiga sistem internal yang saling bekerja sama untuk keperluan pembayaran mahasiswa yaitu:

1) Sistem Informasi Pembayaran yang digunakan langsung oleh petugas keuangan. Sistem ini mencakup semua module yang dibutuhkan untuk mengelola pembayaran data pembayaran mahasiswa.
2) Sistem Informasi Pembayaran versi kasir yang diinstall di masing-masing komputer teller bank mitra. Sistem ini hanya mempunyai module inquiry (cek tagihan), payment (pembayaran) dan laporan transaksi masing-masing teller.
3) Sistem Informasi Pembayaran versi online berbasis web service yang merupakan penghubung antara sistem bank mitra dan database amikom. Sistem ini hanya mempunyai module inquiry (cek tagihan),

payment (pembayaran) dan reversal (pembatalan pembayaran).

Salah satu permasalahan penting dan mendesak yang perlu dicarikan solusinya oleh tim pengembang software Innovation Center Universitas AMIKOM Yogyakarta adalah bagaimana mengintegrasikan inter-sistem pembayaran mahasiswa yang ada, karena sistem ini walaupun mengakses sumber data dan fungsi yang sama tetapi secara sistem, sistem-sistem ini berdiri sendiri sehingga menjadi masalah tersendiri bagi tim pengembang software Innovation Center Universitas AMIKOM Yogyakarta seperti:

1) Kesulitan dalam pemeliharaan sistem jika ada permintaan fitur baru atau perubahan logic sistem karena mereka akan melakukan perubahan yang sama di tiga lokasi sistem yang berbeda.
2) Tidak platform independence artinya kesulitan bagi sistem lain yang berbeda platform seperti web atau mobile untuk mengakses informasi yang dihasilkan oleh sistem ini.

Berdasarkan permasalah tersebut maka pada penelitian ini, web service akan diterapkan untuk memecahkan masalah tersebut yaitu mencari solusi integrasi sistem dengan membandingkan beberapa metode web service yang sudah terbukti bisa menyelesaikan masalah integrasi sistem yaitu metode *SOAP (Simple Object Access Protocol)* dan *REST (Representational state transfer)* yang untuk selanjutkan akan disingkat dengan istilah *SOAP* dan *REST*.

Pemilihan *web service* ini dikarenakan dari beberapa jurnal-jurnal yang penulis baca seperti penelitian yang berjudul Strategi Pengembangan Web Service Untuk Integrasi Inter Sistem E-Government di Pemerintah Kabupaten Bantul Yogyakarta [1]. Dalam penelitian ini mengembangkan sistem web service dengan metode *REST* untuk mengintegrasikan sistem aplikasi dan website yang ada di lingkungan pemkab Bantul yang berpotensi diintegrasi menggunakan database kependudukan nasional.

Penelitian lain yang berjudul Pemanfaatan Web Service Sebagai Integrasi Data Farmasi di RSU Banyumas [2]. Dalam penelitian ini web service dibangun menggunakan metode SOAP.

Penelitian lain yang berjudul yang berjudul Kinerja Web Service pada Proses Integrasi Data [3]. Dalam penelitan ini melakukan uji kinerja web service untuk beberapa metode yang digunakan yaitu *Extensible Markup Language Remote Procedure Call (XML-RPC)*, *SOAP* dan *REST*. Penelitian ini difokuskan untuk mengembangkan sebuah model integrasi data yang mengimplementasikan seluruh metode web service pada aplikasi sistem informasi terpisah serta melakukan analisis dengan mengevaluasi kinerja untuk mendapatkan konfigurasi terbaik.

II. TINJAUAN PUSTAKA

Pada penelitan yang berjudul Strategi Pengembangan Web Service Untuk Integrasi Inter Sistem E-Government di Pemerintah Kabupaten Bantul Yogyakarta [1], melakukan identifikasi sistem aplikasi yang berjumlah 27 aplikasi dan website yang berjumlah 34 website di lingkungan pemkab Bantul yang berpotensi diintegrasi menggunakan database kependudukan nasional. Integrasi sistem menggunakan web service dengan metode *REST*.

Berikutnya penelitian yang berjudul Pemanfaatan Web Service Sebagai Integrasi Data Farmasi di RSU Banyumas [2]. Dalam penelitian ini web service dibangun menggunakan metode *Simple Object Access Protocol (SOAP)* untuk membangung layanan web (web service) sebagai sarana integrasi data pada Farmasi RSU Banyumas.

Penelitian lain yang berjudul yang berjudul Kinerja Web Service pada Proses Integrasi Data [3]. Dalam penelitan ini melakukan uji kinerja web service untuk beberapa metode yang digunakan yaitu *Extensible Markup Language Remote Procedure Call (XML-RPC)*, SOAP dan REST. Penelitian ini difokuskan untuk mengembangkan sebuah model integrasi data yang mengimplementasikan seluruh metode web service pada aplikasi sistem informasi terpisah serta melakukan analisis dengan mengevaluasi kinerja untuk mendapatkan konfigurasi terbaik

III. METODOLOGI PENELITIAN

Penelitian ini secara umum merupakan penelitian yang bertujuan untuk melakukan uji coba kinerja sistem yang dibangun menggunakan teknologi web service sehingga hasil akhir dari penelitian ini diharapkan bisa memberikan gambaran solusi mana yang sebaiknya dikembangkan untuk menyelesaikan masalah integrasi inter-sistem.

Metode penelitian yang digunakan dalam penelitian ini adalah metode penelitian eksperimen (percobaan). Hakikat penelitian eksperimen adalah meneliti pengaruh perlakuan terhadap prilaku yang timbul sebagai akibat perlakuan [4]. Penelitian

eksperimen pada prinsipnya dapat didefinisikan sebagai metode sistematis guna membangun hubungan yang mengandung fenomena sebab akibat [5].

Berdasarkan definisi dari para ahli tersebut, dapat dipahami bahwa penelitian eksperimen adalah penelitian yang dilakukan untuk mengetahui pengaruh pemberian suatu treatment atau perlakuan terhadap subjek penelitian.

Dalam penelitian ini peneliti membandingkan dua metode yang bisa digunakan untuk membuat sistem web service yaitu *SOAP* dan *REST*. Sehingga dengan perbandingan ini bisa dilihat kinerjanya mana yang lebih baik. Dari penelitian ini diharapkan bisa mendapatkan parameter atau nilai yang biasanya digunakan sebagai parameter uji kinerja web service yaitu Response Time, Throughput, dan Received/Sent. Perubahan ketiga nilai dari parameter ini, sangat tergantung kepada nilai parameter Ramp-up periode yang digunakan untuk mengatur jeda antar request atau pemanggilan layanan web service.

IV. HASIL DAN PEMBAHASAN

Dalam penelitian ini peneliti melakukan uji kinerja (performance) sistem dari dua metode yang bisa digunakan untuk membuat sistem web service yaitu SOAP dan REST. Untuk metode SOAP peneliti menggunakan sistem web service pembayaran Universitas Amikom Yogyakarta yang sudah ada, sedangkan untuk metode REST peneliti membuatkan prototipe sistem berdasarkan sistem lama yang sudah ada.

4.1. API Prototipe REST Web Service (prototipe sistem)

API (Application Programming Interface) merupakan interface dari sebuah web service yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan. API ini dibangun dengan spesifikasi seperti berikut:

1) Menggunakan teknologi .NET
2) Bahasa pemrograman – C#
3) Web Server – IIS (Internet Information Services)
4) Database – SQL Server
5) Teknologi akses data – Micro ORM Dapper.NET

Berikut adalah deskripsi lengkap *API* prototipe *REST web service* yang dibangun:

1) Inquiry

Layanan ini digunakan untuk mengirimkan data informasi dari customer mitra kepada bank mitra berdasarkan request yang dikirimkan. Deskripsi lengkap nama method, format request dan response lengkapnya bisa dilihat pada Tabel I dan Tabel II.

TABEL I.
REQUEST SERVICE INQUIRY

Method	**URL**
POST	api/mpom/inquiry
Type	**Params**
DATA_PARAM	{ "vano": "7575010016013676", "channelID": "1", "refno": "12345", "trxdate": "20170729075646" }

TABEL II.
RESPONSE SERVICE INQUIRY

Status	**Response**
200 - OK	{ "ccy": "360", "custname": "MUHAMMAD DICKY MAHENDRA RAMADH", "bill": "1150000", "description1": "SPP Tetap", "description2": "", "method": "inquiry", "err": "00" }
400 - Bad Request	{ "Message": "request format not valid" }

2) Payment

Layanan ini digunakan untuk memproses transaksi pembayaran yang masuk ke mitra, agar mitra langsung ter-update datanya secara realtime, dan proses pembayaran dikatakan berhasil apabila mitra telah memberikan keterangan bahwa transaksi pembayaran sukses dilakukan. Deskripsi lengkap nama method, format request dan response lengkapnya bisa dilihat pada TabelIII dan Tabel IV.

TABEL III.
REQUEST SERVICE PAYMENT

Method	**URL**
POST	api/mpom/payment
Type	**Params**
DATA_PARAM	{ "vano": "7575010016013676", "trxdate": "20170729075646", "ccy": "360", "channelID": "1", "custname": "MUHAMMAD DICKY MAHENDRA RAMADH", "refno": "12345", "bill": "1150000", "payment": "1150000" }

TABEL IV.
RESPONSE SERVICE INQUIRY

Status	**Response**
200 - OK	{ "method": "payment", "err": "00", "errdesc": "Approve" }
400 - Bad Request	{ "Message": "request format not valid" }

3) Reversal

Layanan ini digunakan untuk menggagalkan proses pembayaran yang terjadi sesuai dengan request yang diberikan oleh bank mitra. Deskripsi lengkap nama method, format request dan response lengkapnya bisa dilihat pada Tabel V dan Tabel VI.

TABEL V.
REQUEST SERVICE REVERSAL

Method	**URL**
POST	api/mpom/reversal
Type	**Params**
DATA_PARAM	{ "vano": "7575010016013676", "paymentdate": "20170729000000", "trxdate": "20170729075646", "channelID": "1", "refno": "12345", "bill": "1150000" }

TABEL VI.
RESPONSE SERVICE REVERSAL

Status	**Response**
200 - OK	{ "method": "reversal", "err": "00", "errdesc": "Approve" }
400 - Bad Request	{ "Message": "request format not valid" }

4.2. Menentukan Parameter Uji Kinerja

Berdasarkan panduan pada buku Performance Testing with JMeter [6], ada beberapa pertanyaan yang harus dijawab sebagai patokan untuk menentukan kriteria/parameter uji kinerja yaitu:

1) Berapa banyak mahasiswa yang melakukan pembayaran dalam sehari?
2) Berapa banyak mahasiswa yang melakukan pembayaran dalam waktu satu jam?
3) Apakah pembayaran ini dilakukan setiap hari atau hanya dalam waktu-waktu tertentu?
4) Apakah jenis kewajiban yang sering dibayarkan?

Untuk menjawab semua pertanyaan di atas, diperlukan data riil transaksi pembayaran yang diambil langsung dari database Universitas AMIKOM Yogyakarta. Sample data yang diambil adalah data pembayaran semester ganjil 2016/2017. Untuk contoh sample datanya bisa dilihat pada Tabel VIIVII, Tabel VIII, Tabel IX, dan Tabel X.

TABEL VII.
SAMPLE DATA PEMBAYARAN PER HARI

No	Tanggal	Jumlah Transaksi
1	15-08-2016	922
2	16-08-2016	882
3	11-08-2016	826
4	18-08-2016	794
5	19-08-2016	727
6	12-08-2016	716
7	07-10-2016	680

8	10-08-2016	658
9	27-07-2016	546
10	28-07-2016	508
11	06-10-2016	482
12	26-07-2016	397
13	05-10-2016	350
14	04-10-2016	311
15	03-10-2016	231
	Rata-rata	602

TABEL VIII.
SAMPLE DATA PEMBAYARAN PER JAM

No	Tanggal	Jam	Jumlah Transaksi
1	15-08-2016	13	202
2	15-08-2016	11	193
3	19-08-2016	9	191
4	12-08-2016	14	190
5	11-08-2016	10	181
6	16-08-2016	11	175
7	10-08-2016	11	173
8	11-08-2016	9	166
9	16-08-2016	9	158
10	10-08-2016	13	155
11	27-07-2016	13	153
12	16-08-2016	10	145
13	15-08-2016	10	144
14	19-08-2016	10	140
15	15-08-2016	14	138
		Rata-rata	167

TABEL IX.
SAMPLE DATA PEMBAYARAN BERDASARKAN BULAN

No	Bulan	Tahun	Jumlah Transaksi
1	Juli	2016	1461
2	Agustus	2016	6709
3	September	2016	2156
4	Oktober	2016	2764
5	November	2016	522
6	Desember	2016	513
7	Januari	2017	191
8	Februari	2017	2
9	Maret	2017	3

TABEL X.
SAMPLE DATA PEMBAYARAN BERDASARKAN JENIS KEWAJIBAN

No	Jenis Kewajiban	Jumlah Transaksi
1	SPP Tetap	5283
2	SPP Variabel	3231
3	Biaya Perkuliahan Perbaikan	1356
4	Biaya Wisuda	840
5	Pendadaran	542

Berdasarkan beberapa sample data di atas, maka parameter yang digunakan untuk uji kinerja adalah sebagai berikut:

1) Jumlah Mahasiswa
   Berdasarkan data pada Tabel , rata-rata jumlah pembayaran mahasiswa dalam satu jam adalah 167 orang, dan jika dikonversi dalam satuan menit maka dalam satu menit rata-rata mahasiswa yang melakukan pembayaran adalah 3 orang. Dalam penelitian ini jumlah data mahasiswa yang diambil adalah 150 orang x dengan jumlah service yang dipanggil (inquiry, payment dan reversal), sehinggal total samplenya adalah 450 data.
2) Jenis Kewajiban
   Berdasarkan data pada Tabel , jenis kewajiban yang paling banyak dibayarkan adalah SPP Tetap.
3) Jeda antar *request*
   Jeda antar request yang digunakan bervariasi sesuai dengan kondisi pada saat testing yaitu 1, 0.5, dan 0.2 detik.

4.3. Membuat Skenario Testing

Pengujian dilakukan mengunakan tool Apache JMeter dengan menggunakan skenario testing sebagai berikut:

TABEL XI.
JUMLAH SAMPLE DATA MAHASISWA YANG MENGAKSES SISTEM WEB SERVICE

No	Service/Layanan yang diakses	Jumlah Mahasiswa
1	Inquiry	150
2	Payment	150
3	Reversal	150
	Total sample	450

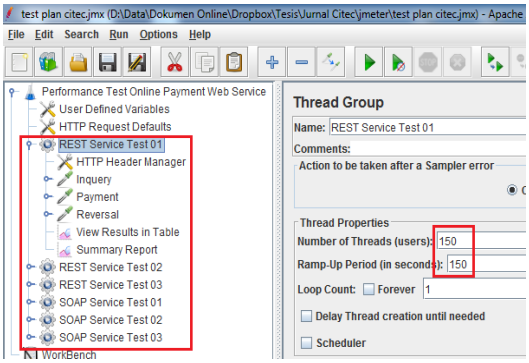
Data pada Tabel XI akan di tes sebanyak 3 kali dengan beberapa variasi nilai ramp-up periode sebagai berikut:

TABEL XII.
NILAI RAMP-UP PERIODE
UNTUK SKENARIO TESTING

No	Ramp-up Period	Delay Request
1	150	1
2	75	0.5
3	30	0.2

Ramp-up period adalah waktu yang dibutuhkan untuk membuat dan menjalankan semua request yang ada. Sebagai contoh untuk layanan inquiry yang diakses sebanyak 150 mahasiswa, dengan nilai ramp-up periode 150, berarti JMeter membutuhkan waktu sekitar 150 detik untuk membuat dan menjalankan semua request yang ada, di mana jeda antar request akan dimulai 1 detik setelah request sebelumnya dimulai.

Kemudian data-data skenario ini akan diinputkan ke dalam JMeter Test Plan, seperti pada Gambar 1.



Gambar 1. JMeter Test Plan

4.4. Menggenerate Data Testing

Setelah membuat skenario testing, langkah berikutnya adalah menyiapkan data-data yang digunakan untuk mendukung proses uji kinerja sistem. Data-data yang diambil berdasarkan skenario testing yang sudah dibuat.

Adapun data-data yang perlu digenerate sebagai berikut:

1. Data Inquiry

Data *inquiry* digunakan untuk mengirimkan data mahasiswa ketika mengakses service *inquiry*. Dari service ini akan menghasilkan data berapa tagihan masing-masing mahasiswa. Data *inquiry* menggunakan format sebagai berikut:

virtual account, channel pembayaran, nomor referensi, tanggal transaksi

contoh:

7575010015013475, 1, 15013475, 20170807091525

2. Data Payment

Data *payment* digunakan untuk mengirimkan data pembayaran masing-masing mahasiswa ketika mengakses service *payment*. Data *payment* menggunakan format sebagai berikut:

virtual account, tanggal transaksi, jenis mata uang, channel pembayaran, nama mahasiswa, nomor referensi, tagihan, pembayaran

contoh:

7575010015013475, 20170807091525, 360, 1, ANIS HANIFAH, 15013475, 1050000, 1050000

3. Data Reversal

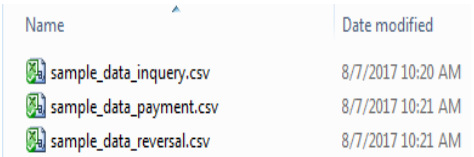
Data *reversal* digunakan untuk mengirimkan data pembayaran masing-masing mahasiswa yang dibatalkan ketika mengakses service *reversal*. Data *reversal* menggunakan format sebagai berikut:

virtual account, tanggal pembayaran, tanggal transaksi, channel pembayaran, nomor referensi, pembayaran

contoh:

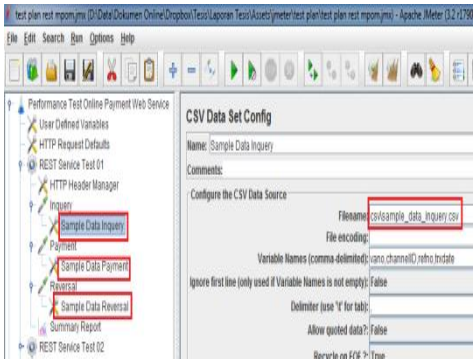
7575010015013475, 20170807000000, 20170807091525, 1, 15013475, 1050000

Semua hasil generate data disimpan dalam bentuk file dengan format csv (*comma-separated values*) seperti pada Gambar 2.

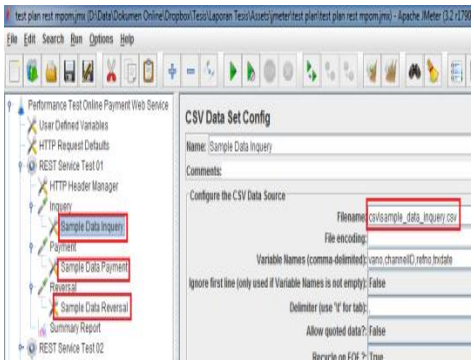


Gambar 2. File Hasil Generate Data

Semua sample data ini akan digunakan JMeter untuk mensimulasikan jumlah mahasiswa yang akan di tes seperti pada



Gambar 3 di bawah ini.



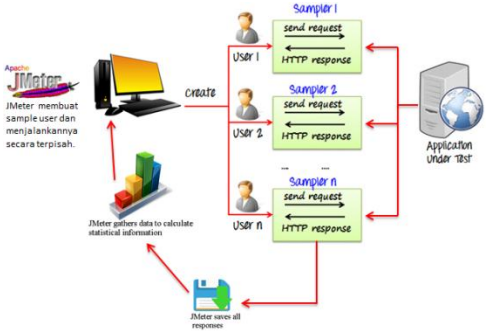
Gambar 3. Konfigurasi sample data Jmeter

4.5. Uji Performance Menggunakan Apache JMeter

Untuk Keperluan uji coba kinerja prototipe sistem baru yang dibangun dan sistem lama menggunakan software *Apache Jmeter*.

Apache Jmeter adalah proyek *open source* yang dibuat menggunakan bahasa pemrograman Java, yang bisa digunakan sebagai *load and performance testing* tool. Selain bisa digunakan untuk melakukan uji kinerja aplikasi *web/web service*, *Apache Jmeter* juga bisa digunakan untuk aplikasi lain yang menggunakan server/protocol *FTP*, *SMTP*, *LDAP*, *Java Database Connectivity (JDBC)*, dan generic *TCP* connections [7].

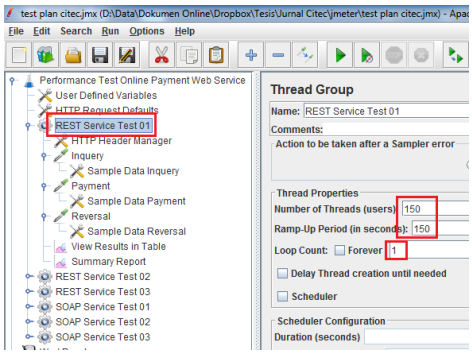
Dalam prosesnya *Apache JMeter* menggunakan *workflow* seperti pada Gambar 4.



Gambar 4. Apache JMeter Workflow

JMeter akan membuat beberapa simulasi user yang akan mengakses server. Jumlah user dan jeda *request* antar user tergantung konfigurasi yang dilakukan pada JMeter. Masing-masing user yang mengakses server akan mendapat *response* dari server, yang kemudian data-data *response* inilah yang disimpan untuk ditampilkan dalam bentuk statistik.

Berdasarkan data skenario testing (tabel XI dan tabel XII) kemudian diinputkan kedalam JMeter seperti yang terlihat pada gambar 5.

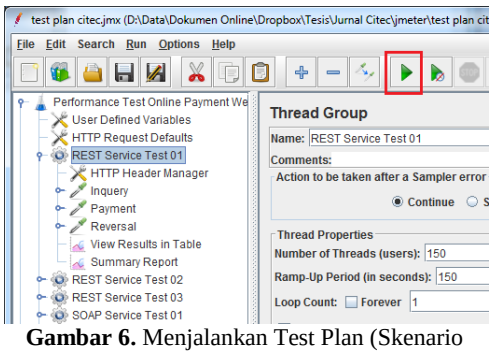


Gambar 5. Konfigurasi Skenario Testing

Contoh pada Gambar 5 di atas, kita melakukan skenario testing sebagai berikut:

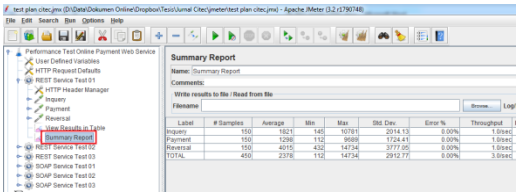
1) Jumlah user sebanyak 150 orang
2) JMeter membutuhkan waktu 150 detik untuk membuat dan menjalankan semua thread(user). Setiap thread akan dimulai 1 (150/150) detik setelah thread sebelumnya dimulai. Atau dengan kata lain setiap 1 detik JMeter akan mengirimkan request ke server.
3) Total jumlah sample = 450 (150 x 3)

Setelah melakukan konfigurasi skenario testing, langkah berikutnya adalah menjalankan tes menggunakan *Apache JMeter* dengan mengklik/ tombol *Start* pada toolbar seperti pada Gambar 6.



Gambar 6. Menjalankan Test Plan (Skenario Testing)

Setelah uji kinerja selesai, JMeter akan menghasilkan hasil test plan (skenario testing) seperti pada Gambar 7 di bawah ini.



Gambar 7. Hasil Test Plan menggunakan format Summary Report

4.6. Analisis Hasil

Analisis hasil pada penelitian ini dilakukan berdasarkan hasil *Summary Report* yang dihasilkan oleh JMeter berdasarkan beberapa skenario testing.

1) Skenario Pertama

a) Jumlah mahasiswa : 150
b) Jumlah service : 3
c) Total jumlah *request* : 450 (150 x 3)
d) *Ramp-up periode* : 150 (detik)
e) *Delay in second* : 1 (Ramp-up periode / jumlah user)

Ramp-up periode adalah waktu yang dibutuhkan untuk membuat dan menjalankan semua *thread*. Jadi berdasarkan skenario pertama, JMeter membutuhkan waktu sekitar 150 detik untuk membuat dan menjalankan semua thread yang ada, di mana jeda antar *thread* akan dimulai 1 detik setelah *thread* sebelumnya dimulai.

Hasil yang didapat pada skenario pertama ini terlihat pada Tabel XIII.

TABEL XIII.
SUMMARY REPORT TESTING SKENARIO PERTAMA

Service	Response Time		Throughput		KB/sec	
	REST	SOAP	REST	SOAP	REST	SOAP
Inquiry	0.12	0.08	1.01	1.01	0.77	1.28
Payment	0.08	0.05	1.01	1.01	0.77	1.4
Reversal	0.33	0.33	1	1	0.73	1.36

Throughput adalah jumlah *request* yang diproses oleh server dalam satuan waktu (detik, menit atau jam). *Response Time* adalah waktu yang dibutuhkan dimulai dari pada saat data dikirim ke server sampai menerima balasan dari server. Sedangkan *Received/Sent* adalah jumlah data yang dikirim atau diterima dari server selama uji kinerja.

Untuk *Throughput* semakin tinggi nilainya semakin bagus kinerjanya sedangkan untuk *Response Time* dan *Received/Sent* semakin kecil nilainya semakin baik.

2) Skenario Kedua

a) Jumlah mahasiswa : 150
b) Jumlah service : 3 (inquiry, payment dan reversal)
c) Total jumlah *request* : 450 (150 x 3)
d) *Ramp-up periode* : 75 (detik)
e) *Delay in second* : 0.5 (Ramp-up periode / jumlah user)

Ramp-up periode adalah waktu yang dibutuhkan untuk membuat dan menjalankan semua *thread*. Jadi berdasarkan skenario kedua, JMeter membutuhkan waktu sekitar 75 detik untuk membuat dan menjalankan semua thread yang ada, di mana jeda antar *thread* akan dimulai 0.5 detik setelah *thread* sebelumnya dimulai.

Hasil yang didapat pada skenario kedua ini terlihat pada Tabel XIV.

TABEL XIV.
SUMMARY REPORT TESTING SKENARIO KEDUA

Service	Response Time		Throughput		KB/sec	
	REST	SOAP	REST	SOAP	REST	SOAP
Inquiry	6.63	0.08	1.86	2.01	1.42	2.56
Payment	4	0.05	1.81	2.01	1.4	2.81
Reversal	8.18	0.32	1.67	2	1.22	2.71

3) Skenario Ketiga

a) Jumlah mahasiswa : 150
b) Jumlah service : 3
c) Total jumlah *request* : 450 (150 x 3)
d) *Ramp-up periode* : 30 (detik)
e) *Delay in second* : 0.2 (Ramp-up periode / jumlah user)

Ramp-up periode adalah waktu yang dibutuhkan untuk membuat dan menjalankan semua *thread*. Jadi berdasarkan skenario ketiga, JMeter membutuhkan waktu sekitar 60 detik untuk membuat dan menjalankan semua thread yang ada, di mana jeda antar *thread* akan dimulai 0.4 detik setelah *thread* sebelumnya dimulai.

Hasil yang didapat pada skenario ketiga ini terlihat pada Tabel XV.

TABEL XV.
SUMMARY REPORT TESTING SKENARIO KETIGA

Service	Response Time		Throughput		KB/sec	
	REST	SOAP	REST	SOAP	REST	SOAP
Inquiry	11.05	8.72	2.86	2.79	2.19	3.57
Payment	12.23	13.13	2.58	2.17	1.99	3.02
Reversal	14.89	14.3	1.97	1.93	1.44	2.61

Kemudian dari keseluruhan hasil uji kinerja sistem, peniliti rangkum hasilnya seperti yang terlihat pada Tabel XVI, Tabel XVII, Tabel XVIII dan Tabel XIX.

TABEL XVI.
RATA-RATA SUMMARY REPORT

Service	Response Time		Throughput		KB/sec	
	REST	SOAP	REST	SOAP	REST	SOAP
Inquiry	5.93	2.96	1.91	1.94	1.46	2.47
Payment	5.44	4.41	1.8	1.73	1.39	2.41
Reversal	7.8	4.98	1.55	1.65	1.13	2.23

TABEL XVII.
SELISIH NILAI PARAMETER *RESPONSE TIME*

Service	Service Lama	Prototipe Service Baru	Selisih (%)
	SOAP	REST	
Inquiry	✓		50.08
Payment	✓		18.93
Reversal	✓		36.15

Untuk hasil *Response Time* semakin kecil nilainya semakin baik. Sistem lama unggul di semua service dengan selisih nilai masing-masing service yaitu: *inquiry* (50.08%), *payment* (18.93%) dan *reversal* (36.15%).

TABEL XVIII.
SELISIH NILAI PARAMETER *THROUGHPUT*

Service	Service Lama	Prototipe Service Baru	Selisih (%)
	SOAP	REST	
Inquiry	✓		1.55
Payment		✓	3.89
Reversal	✓		6.06

Sedangkan untuk hasil *Throughput* semakin besar nilainya semakin baik [8]. Prototipe sistem baru unggul di service *payment* (3.89%) dan sistem lama unggul di service *inquiry* (1.55%) dan *reversal* (6.06%).

TABEL XIX.
SELISIH NILAI PARAMETER *RECEIVED/SENT (KB/SEC)*

Service	Service Lama	Prototipe Service Baru	Selisih (%)
	SOAP	REST	
Inquiry		✓	40.89
Payment		✓	42.32
Reversal		✓	49.33

Terakhir untuk variabel *Received/Sent (kb/sec)*, prototipe sistem baru unggul di semua service yaitu: *inquiry* (40.89%), *payment* (42.32%) dan *reversal* (49.33%).

V. Kesimpulan dan Saran

Uji performance (kinerja) sistem yang sudah dilakukan, baik terhadap sistem lama (*SOAP*) dan prototipe sistem baru (*REST*), menghasilkan tiga variabel yang dianalisis yaitu: *Response Time*, *Throughput*, dan *Received/Sent* data.

Untuk hasil *Response Time*, sistem lama unggul di semua service dengan selisih nilai masing-masing service yaitu: *inquiry* (50.08%), *payment* (18.93%) dan *reversal* (36.15%). Sedangkan untuk hasil *Throughput*, prototipe sistem baru unggul di service payment (3.89%) dan sistem lama unggul di service *inquiry* (1.55%) dan *reversal* (6.06%). Terakhir untuk *Received/Sent* (kb/sec) data, prototipe sistem baru unggul di semua service yaitu: *inquiry* (40.89%), *payment* (42.32%) dan *reversal* (49.33%).

Sebagai saran untuk penelitian lebih lanjut, penambahan parameter latency time, min, dan max sebagai parameter pengujian perlu dilakukan untuk menghasilkan analisis kinerja yang lebih mendalam. Selain itu solusi untuk isu keamanan mungkin juga perlu dimasukkan pada saat pengembangan sistem atau prototipe sistem.

REFERENSI

[1] E. Sutanta and K. Mustofa, "Strategi Pengembangan Web Service Untuk Integrasi Inter Sistem E-Government," *Sisfotenika*, pp. 1–5, 2012.

[2] A. Adi and Riyanto, "Pemanfaatan Web Service Sebagai Integrasi Data Farmasi di RSU Banyumas," *Juita*, vol. II, pp. 231–238, 2013.

[3] Yogiswara, Wijono, and H. S. Dahlan, "Kinerja Web Service pada Proses Integrasi Data," vol. 1, no. 1, pp. 73–78, 2014.

[4] A. Alsa, *Pendekatan Kuantitatif Kualitatif dalam Penelitian Psikologi*, 6th ed. Yogyakarta: Pustaka Pelajar, 2011.

[5] Sukardi, *Metodologi Penelitian Pendidikan: Kompetensi dan Praktiknya*. Jakarta: PT Bumi Aksara, 2011.

[6] B. Erinle, *Performance Testing with JMeter 2.9*. 2013.

[7] T. A. S. Foundation, "Apache JMeter.," *The Apache Software Foundation*. [Online]. Available: http://jmeter.apache.org/. [Accessed: 15-Mar-2016].

[8] P. A. Ochang and P. Irving, "Performance Analysis of Wireless Network Throughput and Security Protocol Integration," vol. 9, no. 1, pp. 71–78, 2016.