

## Terakreditasi SINTA Peringkat 4

Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 28/E/KPT/2019  
masa berlaku mulai Vol.3 No. 1 tahun 2018 s.d Vol. 7 No. 1 tahun 2022

Terbit online pada laman web jurnal:  
<http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>



Vol. 5 No. 2 (2020) 129 - 136

# JOINTECS

**(Journal of Information Technology  
and Computer Science)**

e-ISSN:2541-6448

p-ISSN:2541-3619

## Penerapan *Restful Web Service* Untuk Optimalisasi Kecepatan Akses Pada Aplikasi Berbasis Android

Lailin Nur Asiyah<sup>1</sup>, Muhammad Priyono Tri Sulistyanto<sup>2</sup>, Abdul Aziz<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Kanjuruhan Malang

<sup>1</sup>lailinofficial@gmail.com, <sup>2</sup>m.priyono.ts@unikama.ac.id, <sup>3</sup>abdul.aziz@unikama.ac.id

### Abstract

*In developing Android applications, we must pay attention to the optimal speed of access to web services. Two web service architectures that are commonly used are SOAP and REST. Systems that use the principles of REST are called "RESTful". SOAP Implementation is still not optimal, because the use of throughput is relatively wasteful. Throughput largely determines the speed of access to web services. Related to this research will discuss optimizing access speed on web services using RESTful. The RESTful test shows the minimum average throughput usage on Android Pie = 1736.8596 bps, while on SOAP the average value obtained on android Lollipop = 11640.4432 bps. In addition to Android, in testing CPU speed specifications, RESTful determines the average value of minimum throughput usage on a combination of Octa-core CPU specifications = 2489.6253 bps, while on SOAP an average value is obtained on a combination of quad-core CPU speed specifications = 12570 bps. RESTful has less throughput than SOAP, which causes RESTful access speeds to be much faster than two times faster than SOAP access speeds. Besides RESTful web services on android and CPU speed specifications can support access speeds.*

*Keywords: RESTful; SOAP; web service; android.*

### Abstrak

Dalam pengembangan aplikasi android harus memperhatikan kecepatan akses *web service* yang optimal. Dua arsitektur *web service* yang secara umum digunakan yaitu *SOAP* dan *REST*. Sistem yang menggunakan prinsip - prinsip dari *REST* disebut dengan "*RESTful*". Penerapan *SOAP* dirasa masih belum optimal, karena penggunaan *throughput* yang relatif boros. *Throughput* sangat berpengaruh pada kecepatan akses *web service*. Sehingga pada penelitian ini akan dibahas mengenai optimalisasi kecepatan akses pada *web service* menggunakan *RESTful*. Pada uji *RESTful* menunjukkan nilai rata – rata minimum pemakaian *throughput* pada android Pie = 1736.8596 bps, sedangkan pada *SOAP* diperoleh nilai rata – rata maksimum pada android Lollipop = 11640.4432 bps. Selain pada android, pada pengujian spesifikasi kecepatan *CPU*, *RESTful* menunjukkan nilai rata – rata minimum pemakaian *throughput* pada kombinasi spesifikasi kecepatan *CPU Octa-core* = 2489.6253 bps, sedangkan pada *SOAP* diperoleh nilai rata – rata maksimum pada kombinasi spesifikasi kecepatan *CPU Quad-core* = 12570 bps. *RESTful* memiliki *throughput* yang lebih sedikit daripada *SOAP*, yang menyebabkan kecepatan akses *RESTful* jauh lebih cepat yaitu dua kali lebih cepat daripada kecepatan akses *SOAP*. Selain *RESTful web service* pada android dan spesifikasi kecepatan *CPU* dapat mendukung kecepatan akses.

Kata kunci: *RESTful; SOAP; web service; android.*

© 2020 Jurnal JOINTECS

### 1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi terus berkembang, memberikan pengaruh besar

terhadap organisasi maupun individu. Yang bertujuan untuk memenuhi kebutuhan informasi maupun komunikasi bagi pengguna. Kemudahan akses informasi antar perangkat dan media merupakan suatu

Diterima Redaksi : 19-01-2020 | Selesai Revisi : 12-02-2020 | Diterbitkan Online : 30-05-2020

fungsi yang sangat penting dalam pengembangan aplikasi untuk menjadi lebih besar, maka diperlukan integrasi sistem yang saling terhubung dalam berbagai macam platform serta perangkat yang ada pada saat ini [1]. Penyajian informasi yang efektif dan tepat guna akan menjadikan performa sebuah perusahaan maupun organisasi semakin berkualitas, untuk menghasilkan informasi tersebut maka dibutuhkan sebuah sistem informasi yang mendukung dalam pengolahan datanya [2].

Web Service merupakan teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi yang dapat digunakan untuk mendukung pertukaran data dan dapat diakses oleh aplikasi lain [3]. Beberapa arsitektur pada web service diantaranya SOAP, WSDL, UDDI dan REST, yang secara umum sering digunakan yaitu Simple Object Access Protocol (SOAP) dan Representational State Transfer (REST) [4].

Menurut Warvante, kinerja RESTful web service lebih cepat dibandingkan dengan SOAP web service [5]. REST merupakan architectural design untuk mengidentifikasi resource, REST menggunakan Uniform Resource Identifier (URI) dan HTTP [6], mendukung format data seperti JSON dan XML [7].

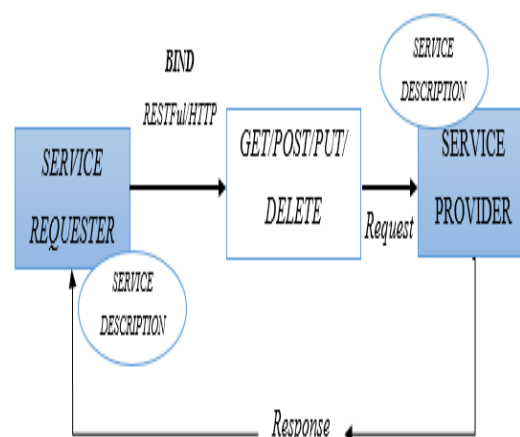
Menurut (R. T. Fielding, 2000), sistem yang menggunakan prinsip – prinsip dari arsitektur REST disebut dengan “RESTful”. Cara kerja RESTful web service yaitu bermula dari client mengirimkan sebuah data atau request melalui protokol HTTP request, kemudian server merespon melalui HTTP response [8].

Prinsip-prinsip ini membuat RESTful menjadi hemat throughput dan mempunyai kecepatan yang optimal. Model arsitektur RESTful web service dapat digambarkan seperti pada Gambar 1. SOAP merupakan web service yang menggunakan protocol berbasis Extensible Markup Language (XML) [6] yang memungkinkan bertukar informasi menggunakan HTTP yang terdiri dari sebuah envelope, yang mendefinisikan apa yang harus dimasukkan dalam pesan dan bagaimana pesan ini harus diproses, seperangkat aturan penyandian, dan konvensi untuk mewakili panggilan prosedur dan tanggapan [3]. Sebagian besar informasi dalam SOAP dan WSDL berlebihan, yang meningkatkan volume komunikasi jaringan dan muatan pada sisi server [9]. Hal ini membuat SOAP menjadi boros dalam pemakaian throughput. Karena, client tidak dapat mengidentifikasi pesan informasi secara langsung dari URI dan HTTP. Throughput adalah jumlah paket data yang diterima ( $\sum bits$ ) yang berhasil diamati pada kurun waktu tertentu dibagi oleh durasi waktu tersebut (s) [10]. Model pada arsitektur SOAP web service dapat digambarkan seperti pada Gambar 2. Sedangkan, perbandingan langkah-langkah routing request dan

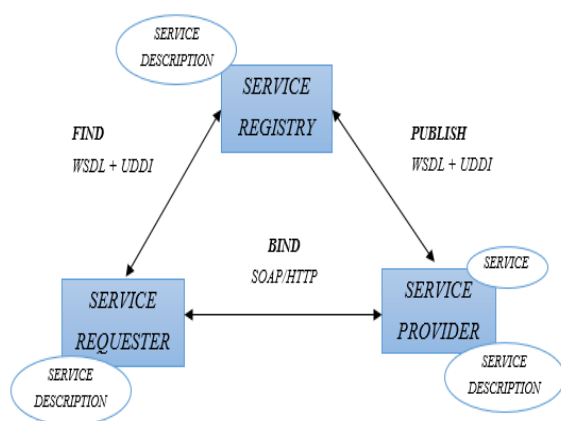
response client ke server pada web service berbasis RESTful dan SOAP ditunjukkan pada Gambar 3 dan 4.

Pada request data dan response time RESTful lebih kecil dibandingkan dengan SOAP yang dibuktikan melalui penelitian yang dilakukan oleh “Dudhe & Sheekar, yang menyatakan bahwa pada implementasi response time di localhost Apache Tomcat (HTTP GET request) tercatat rata - rata request data dan response time pada SOAP yaitu sebesar 425 Byte dan 528 ms. Sedangkan pada RESTful hanya sebesar 49 Byte dan 115 ms. Karena, request size datanya lebih sedikit menyebabkan transmission time menjadi lebih singkat, sehingga konsumsi daya menjadi lebih rendah maka, kinerja kecepatan akses pada web service menjadi lebih optimal dibandingkan dengan SOAP” [11]. Hal ini juga diperkuat oleh penelitian lain yang dilakukan oleh “Bora dan Bezboruah, yang menyatakan bahwa RESTful mempunyai kecepatan Response Time (RT) dengan rata – rata 2.115 s sedangkan SOAP 10.978 s” [3]. Penelitian ini juga membuktikan penelitian sebelumnya dimana kecepatan RESTful adalah dua kali lipat lebih cepat daripada SOAP [11].

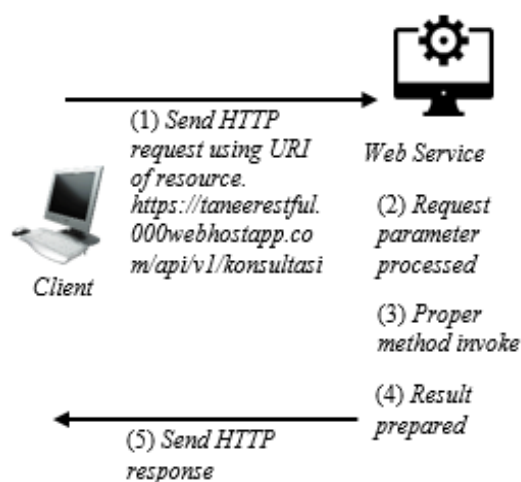
Pada hasil dari penelitian tersebut, dapat disimpulkan bahwa throughput berpengaruh pada kecepatan akses web service dan erat kaitannya dengan masalah efektivitas dan penggunaan biaya koneksi internet. Penggunaan pada throughput yang boros menjadi persoalan dalam proses request dan response data dari web service [12]. Sehingga, semakin optimal kecepatan akses web service akan diikuti dengan meningkatnya kepuasan penggunaannya [13]. Banyak pengguna menggunakan koneksi internet pada smartphone untuk mengakses teknologi informasi maupun komunikasi. Android atau smartphone merupakan perangkat mobile yang paling banyak digunakan untuk mengakses internet sebesar 72, 9%, berdasarkan pada survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) [14]. Dan survei pada tanggal 17 Juni 2019 yang menyatakan bahwa android merupakan platform yang paling populer bagi para developer, yaitu sebanyak 83% developer aplikasi bergerak [15].



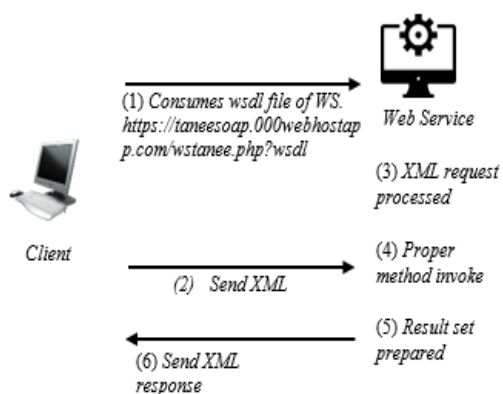
Gambar 1. Model Arsitektur Restful Web Service



Gambar 2. Model Arsitektur SOAP Web Service



Gambar 3. Langkah – Langkah Routing pada Restful Web Service



Gambar 4. Langkah – Langkah Routing pada SOAP Web Service

## 2. Metode Penelitian

### 2.1. Rancangan Penelitian

Penggunaan metode pada penelitian ini adalah metode penelitian eksperimen dengan pendekatan kuantitatif. Bentuk yang digunakan dalam penelitian ini adalah “Nonequivalent Control Group Design” yang merupakan bentuk dari metode penelitian semu (*quasi eksperimen*). Adapun skema model penelitian disajikan pada Tabel 1.

Tujuan pada penelitian ini adalah untuk membuktikan ada tidaknya perbedaan yang signifikan pada kelas eksperimen dan menyediakan kelas kontrol dengan cara memberikan perlakuan tertentu. *Web service* pada kelas eksperimen memperoleh perlakuan dengan menggunakan *RESTful*, sedangkan pada kelas kontrol memperoleh perlakuan menggunakan *SOAP*. Pada hasilnya nanti akan terlihat manakah yang lebih optimal antara kelompok eksperimen dan kelompok kontrol.

### 2.2. Populasi dan Sampel

Populasi dalam penelitian ini adalah seluruh hasil pengukuran *throughput* pada *web service* aplikasi pelayanan pertanian “TANEE” yang meliputi layanan ahli, irigasi, pembibit, *store*, tenaga, traktor dan *transport* pada *request POST*. Jumlah data layanan pertanian “TANEE” dapat dicermati pada Tabel 2. Sampelnya adalah semua populasi dari seluruh hasil proses pengukuran *throughput* pada *web service* aplikasi pelayanan pertanian “TANEE” yang meliputi layanan ahli, irigasi, pembibit, *store*, tenaga, traktor dan *transport* yang berjumlah 86 data.

### 2.3. Alat dan Bahan Penelitian

Alat yang digunakan pada penelitian ini adalah *mobile device* dengan sistem operasi android minimum *sdk version 5.0* atau Lollipop ke atas. Dan dengan spesifikasi kecepatan *CPU minimum Quad-core 1.0 GHz Cortex-A53*. Dan bahan dari penelitiannya adalah data index penggunaan *throughput* pada pemakaian *SOAP* dan *RESTful*.

### 2.4. Analisis Data

Teknik analisis data yang digunakan yaitu analisis data kuantitatif dengan menggunakan uji deskriptif dan uji *One Way ANOVA* dengan taraf nilai signifikansi pada  $\alpha = 0.05$ . Untuk mengukur *throughput* yang lebih optimal pada *web service* dilakukan dengan menerapkan *RESTful* dan *SOAP*.

Tabel 1. Skema Tabel Penelitian

Kelompok	Perlakuan	Post-Test
Eksperimen	<i>RESTful</i>	Hasil <i>throughput</i>
Kontrol	<i>SOAP</i>	Hasil <i>throughput</i>

Tabel 2. Layanan Aplikasi Pelayanan Pertanian “TANEE”

Jenis Layanan	Jumlah Data Layanan
Ahli	5
Irigasi	12
Lahan	5
Pembibit	9
<i>Store</i>	25
Tenaga	5
Traktor	10
<i>Transport</i>	15
Jumlah	86

```
#RESTFUL GET TENAGA KERJA
#SIZE
Size Header      : 143 bytes
Size Download    : 358 bytes
Size Request     : 128 bytes
Size Upload      : 0 bytes

#TIME
Time Appconnect  : 0.000000 seconds
Time Connect     : 0.276462 seconds
Time Namelookup  : 0.029051 seconds
Time Pretransfer : 0.277061 seconds
Time Redirect    : 0.000000 seconds
Time Starttransfer : 0.000000 seconds
Time Total       : 0.567194 seconds

#SPEED
Speed Download   : 631.000 bps
Speed Upload     : 0.000 bps
```

Gambar 5. Hasil Pengujian Proses *POST Restful*

```
#SOAP GET TENAGA KERJA
#SIZE
Size Header      : 283 bytes
Size Download    : 1752 bytes
Size Request     : 151 bytes
Size Upload      : 180 bytes

#TIME
Time Appconnect  : 0.625873 seconds
Time Connect     : 0.283907 seconds
Time Namelookup  : 0.034314 seconds
Time Pretransfer : 0.626754 seconds
Time Redirect    : 0.000000 seconds
Time Starttransfer : 0.000000 seconds
Time Total       : 0.910644 seconds

#SPEED
Speed Download   : 1925.000 bps
Speed Upload     : 197.000 bps
```

Gambar 6. Hasil Pengujian Proses Proses *POST SOAP*



Gambar 7. Tampilan Layanan dari Aplikasi Pelayanan Pertanian “TANEE”

### 3. Hasil dan Pembahasan

#### 3.1. Deskripsi Data

Analisa data untuk mengukur *throughput* pada aplikasi pelayanan pertanian “TANEE” yang meliputi data layanan ahli, irigasi, lahan, pembibit, *store*, tenaga, traktor dan *transport* pada *request POST* manakah yang lebih optimal diantara menggunakan *SOAP* dan *RESTful* pada *web service*.

Hasil pengukuran *throughput* dapat diketahui dari nilai *posttest* antara *RESTful* dan *SOAP*. Pengujian *web service* dilakukan menggunakan *termux* yang memiliki fungsi sebagai *tool curl* untuk memanggil *web service* pada android. Proses *request* yang dilakukan pada *termux* yaitu *POST*.

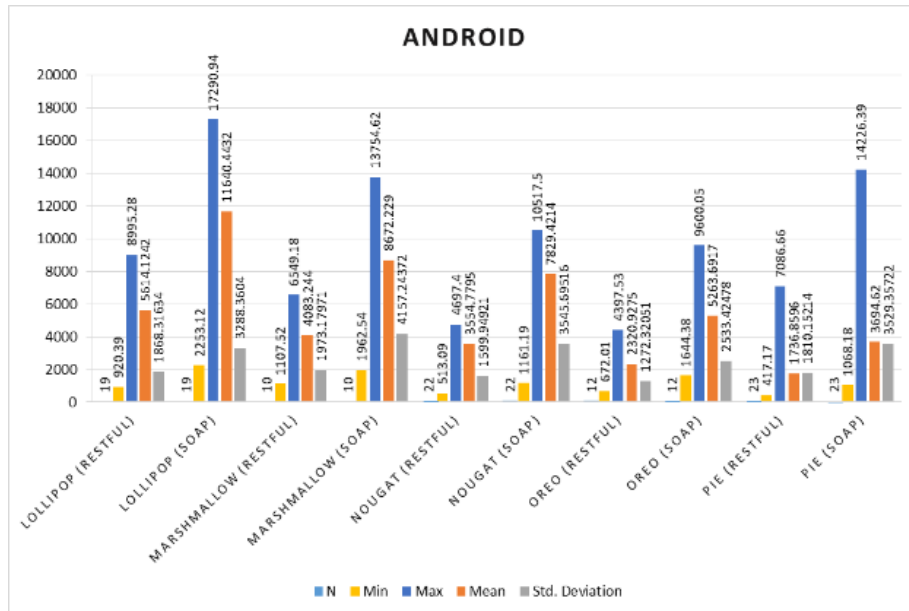
Tampilan hasil pengujian dari *web service RESTful* dan *SOAP* ditunjukkan pada Gambar 5 dan 6. Pada Gambar 5, menunjukkan *throughput* menggunakan *RESTful* yang dihasilkan dari proses *POST* dengan *time* 0.567194 *seconds*, *size* 358 *bytes* dan *throughput* 631 *bps*.

Tabel 3. Hasil Pengujian pada Android

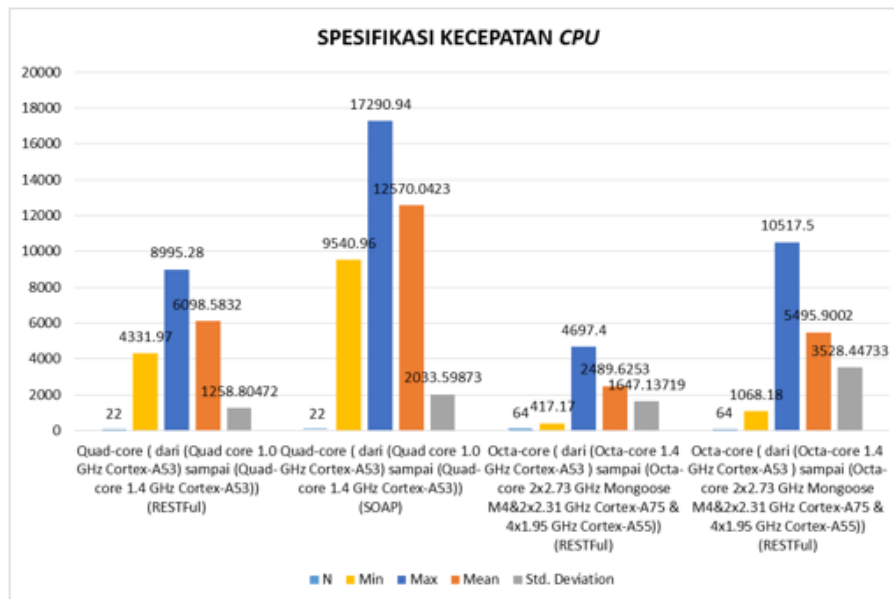
No	Android	Kelas	N	Mean (bps)
1	Lollipop	<i>RESTful</i>	19	5614.1242
		<i>SOAP</i>	19	11640.4432
2	Marshmallow	<i>RESTful</i>	10	4083.2440
		<i>SOAP</i>	10	8672.2290
3	Nougat	<i>RESTful</i>	22	3554.7795
		<i>SOAP</i>	22	7829.4214
4	Oreo	<i>RESTful</i>	12	2320.9275
		<i>SOAP</i>	12	5263.6917
5	Pie	<i>RESTful</i>	23	1736.8596
		<i>SOAP</i>	23	3694.6200

Tabel 4. Hasil Pengujian pada Spesifikasi Kecepatan CPU

No	Spesifikasi Kecepatan CPU	Kelas	N	Mean (bps)
1	Kombinasi Quad-core	<i>RESTful</i>	22	6098.5832
		<i>SOAP</i>	22	12570.0423
2	Kombinasi Octa-core	<i>RESTful</i>	64	2489.6253
		<i>SOAP</i>	64	5495.9002



Gambar 8. Grafik Perbandingan pada Android



Gambar 9. Grafik Perbandingan pada Spesifikasi Kecepatan CPU

### 3.2. Pengujian dan Perbandingan Deskriptif pada Android

#### 3.2.2. Pengujian Pada Android

Pengujian pada android dibagi menjadi 5 kelompok yang meliputi android Lollipop, Marshmallow, Nougat, Oreo dan Pie. Untuk mengetahui perbandingan antara *RESTful* dan *SOAP* maka, dilakukan pengujian pada penggunaan *throughput*.

Hasil pengujian *RESTful* dan *SOAP* pada android dapat dicermati pada Tabel 3. Tabel 3 adalah nilai *minimum*, *maximum*, *mean* dan *std.deviation* pada proses pengujian sebanyak delapan puluh enam kali pada *RESTful* dan *SOAP* pada penggunaan *throughput*.

#### 3.2.3. Perbandingan Kecepatan *RESTful* dan *SOAP* pada Android

Berdasarkan hasil pengujian yang menyatakan penggunaan *throughput* diantara *RESTful* dan *SOAP*. Pada *RESTful* diperoleh nilai rata – rata *minimum* penggunaan *throughput* pada Pie = 1736.8596 *bps*. Sedangkan, *SOAP* diperoleh nilai rata – rata *maximum* penggunaan *throughput* pada Lollipop = 11640.4432 *bps*. Sedangkan, Gambar 6 menunjukkan *throughput* menggunakan *SOAP* dengan *time* 0.910644 *seconds*, *size* 1752 *bytes* dan *throughput* 1925 *bps*. *POST* untuk mengirimkan *parameter* yang berupa data layanan ahli, irigasi, lahan, pembibit, *store*, tenaga, traktor dan *transport* yang berjumlah 86 data. Jenis layanan dari

aplikasi pelayanan pertanian “TANEE” yang diujikan dapat dicermati pada Gambar 7.

Berdasarkan hasil pengujian antara *RESTful* dan *SOAP* pada android secara rata – rata dapat disimpulkan bahwa *RESTful* memiliki penggunaan *throughput* yang lebih sedikit daripada *SOAP* yang menyebabkan kecepatan akses *RESTful* lebih cepat daripada kecepatan akses *SOAP*. Grafik hasil perbandingan pengujian pada penggunaan android dapat dicermati pada Gambar 8.

### 3.3. Pengujian dan Perbandingan Deskriptif pada Spesifikasi Kecepatan CPU

#### 3.3.1. Pengujian Pada Spesifikasi Kecepatan CPU

Pengujian spesifikasi kecepatan CPU dibagi menjadi 2 kelompok, yaitu kombinasi spesifikasi kecepatan CPU *Quad-core* (mulai dari (*Quad core 1.0 GHz Cortex-A53*) sampai (*Quad-core 1.4 GHz Cortex-A53*)) dan *Octa-core* (mulai dari (*Octa-core 1.4 GHz Cortex-A53*) sampai (*Octa-core 2x2.73 GHz Cortex-A55*)). Untuk mengetahui perbandingan antara *RESTful* dan *SOAP* dilakukan pengujian pada penggunaan *throughput*. Hasil pengujian pada spesifikasi kecepatan CPU dapat dicermati pada Tabel 4. Tabel 4 adalah nilai *mean* pada proses pengujian delapan puluh enam kali menggunakan *RESTful* dan *SOAP* pada penggunaan *throughput*.

#### 3.3.2. Perbandingan Pada Spesifikasi Kecepatan CPU

Berdasarkan hasil pengujian menyatakan penggunaan *throughput RESTful* dan *SOAP*. Pada *RESTful* diperoleh nilai rata – rata *minimum* penggunaan *throughput* pada kombinasi spesifikasi kecepatan CPU *Octa-core* = 2489.6253 *bps*. Sedangkan pada *SOAP* didapatkan nilai rata – rata *maximum* penggunaan *throughput* pada kombinasi spesifikasi kecepatan CPU *Quad-core* = 12570.0423 *bps*. Berdasarkan hasil pengujian *RESTful* dan *SOAP* pada spesifikasi kecepatan CPU secara rata – rata dapat disimpulkan bahwa *RESTful* memiliki penggunaan *throughput* lebih sedikit daripada *SOAP* yang dapat menyebabkan kecepatan akses pada *RESTful* lebih cepat daripada kecepatan akses *SOAP*. Grafik hasil pengujian pada penggunaan spesifikasi kecepatan CPU dapat dicermati pada Gambar 9.

### 3.4 Pengujian Hipotesis

#### 3.4.1. Pengujian Hipotesis Pada Android

Pengujian hipotesis dilakukan untuk menunjukkan ada tidaknya perbedaan yang signifikan pada penggunaan *throughput* diantara *RESTful* dan *SOAP* pada *web service*. Dengan taraf nilai pada  $\alpha$  yaitu,  $\alpha < 0.05$ . Hasil pengujian hipotesis pada android dapat dicermati pada Tabel 5. Tabel 5 adalah nilai *mean* dan *Sig* dengan hasil nilai pada *Sig*. yaitu 0.00. Pada proses pengujian menggunakan analisis *compare means One Way ANOVA* pada *RESTful* dan *SOAP*.

Karena nilai *Sig*. dari *RESTful* dan *SOAP* = .000 <  $\alpha$  (0.05) (lebih kecil dari nilai  $\alpha$ ) maka, hal ini menunjukkan ada perbedaan yang signifikan pada penggunaan *throughput* antara *RESTful* dan *SOAP*. Dapat disimpulkan bahwa kecepatan akses pada penggunaan *RESTful* lebih optimal dari pada penggunaan *SOAP*. Dapat diamati *mean* penggunaan *throughput* antara *RESTful* dan *SOAP* pada setiap android selalu menanjak. Penggunaan android juga mendukung kecepatan akses, yang ditandai saat menggunakan lollipop (versi lama) terlihat *gap* semakin lebar. *Mean gap throughput RESTful* dan *SOAP* dapat dicermati pada Gambar 10. Karena nilai *Sig*. dari *RESTful* dan *SOAP* = .000 <  $\alpha$  (0.05) (lebih kecil dari

Tabel 5. Hasil Uji Hipotesis pada Android

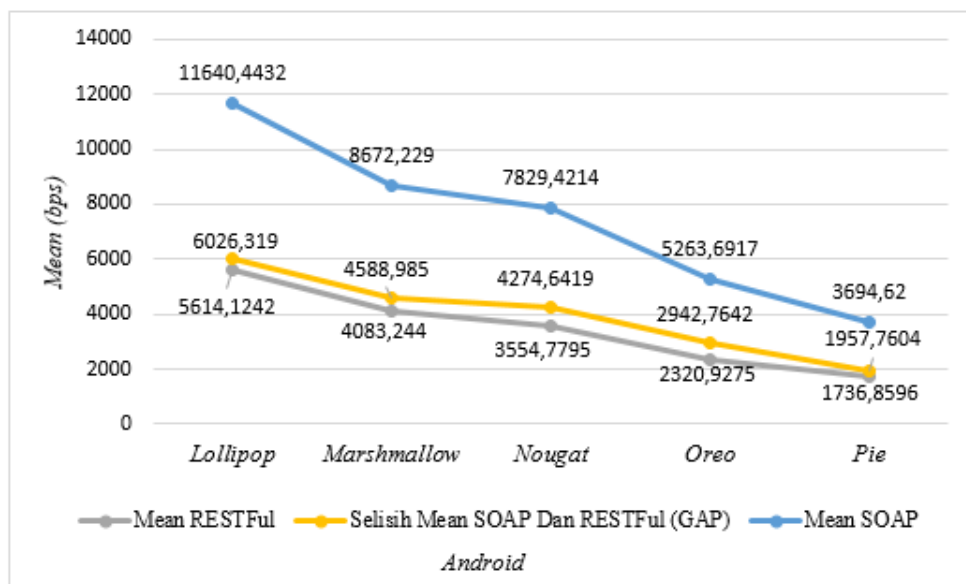
No	Android	Kelas	Mean (bps)	Sig.
1	Lollipop	<i>RESTful</i>	5614.1242	0.00
		<i>SOAP</i>	11640.4432	0.00
2	Marshmallow	<i>RESTful</i>	4083.2440	0.00
		<i>SOAP</i>	8672.2290	0.00
3	Nougat	<i>RESTful</i>	3554.7795	0.00
		<i>SOAP</i>	7829.4214	0.00
4	Oreo	<i>RESTful</i>	2320.9275	0.00
		<i>SOAP</i>	5263.6917	0.00
5	Pie	<i>RESTful</i>	1736.8596	0.00
		<i>SOAP</i>	3694.6200	0.00

Tabel 6. Hasil Uji Hipotesis pada Spesifikasi Kecepatan CPU

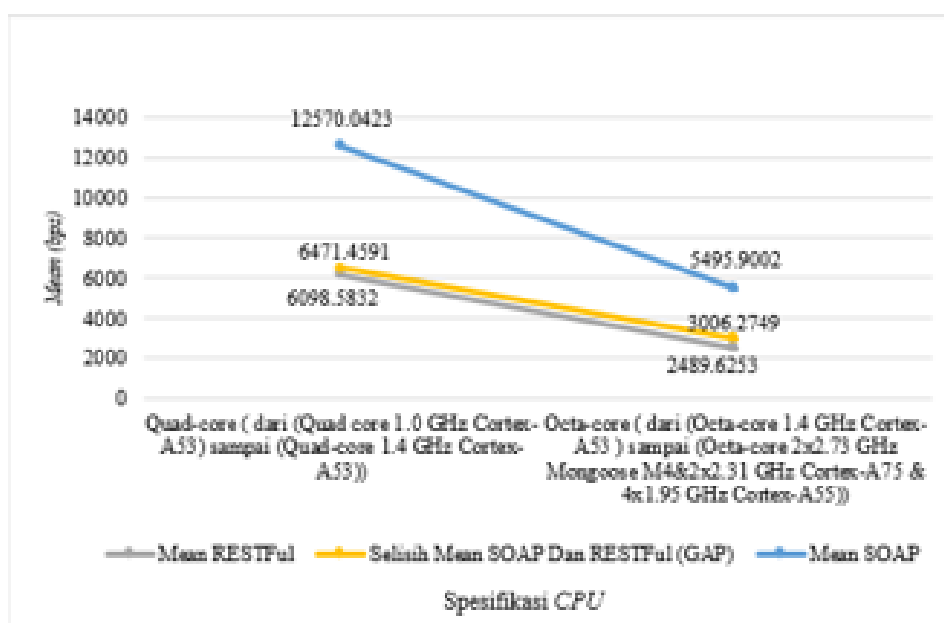
No	Spesifikasi Kecepatan CPU	Kelas	Mean (bps)	Sig.
1	Kombinasi <i>Quad-core</i>	<i>RESTful</i>	6098.5832	0.00
		<i>SOAP</i>	12570.0423	0.00
2	Kombinasi <i>Octa-core</i>	<i>RESTful</i>	2489.6253	0.00
		<i>SOAP</i>	5495.9002	0.00

#### 3.4.2. Pengujian Hipotesis Pada Spesifikasi Kecepatan CPU

Pengujian hipotesis dilakukan untuk menunjukkan ada tidaknya perbedaan yang signifikan pada penggunaan *throughput* diantara *RESTful* dan *SOAP* pada *web service*. Dengan taraf nilai pada  $\alpha$  yaitu,  $\alpha < 0.05$ . Hasil pengujian hipotesis pada spesifikasi kecepatan CPU dapat dicermati pada Tabel 6. Tabel 6 adalah nilai *mean* dan *Sig*. dengan hasil nilai pada *Sig*. yaitu 0.00. Pada proses pengujian menggunakan analisis *compare means One Way ANOVA* pada *RESTful* dan *SOAP*. Sedangkan pada saat menggunakan kombinasi spesifikasi kecepatan CPU *Octa-core* (spesifikasi tinggi) terlihat *gap* semakin sempit yang menunjukkan, semakin tingginya spesifikasi kecepatan CPU memiliki kecepatan akses yang lebih cepat. Dapat disimpulkan bahwa kecepatan akses pada penggunaan *RESTful* lebih optimal dari pada penggunaan *SOAP*.



Gambar 10. Grafik Mean Gap pada Android



Gambar 11. Grafik Mean Gap pada Spesifikasi Kecepatan CPU

nilai  $\alpha$ ) maka, hal ini menunjukkan ada perbedaan yang signifikan diantara *RESTful* dan *SOAP*.

Sedangkan pada saat menggunakan pie (versi baru) terlihat *gap* semakin sempit yang menunjukkan, dan semakin barunya android memiliki kecepatan akses yang lebih cepat. Jadi, apabila *RESTful* lambat maka *SOAP* jauh lebih lambat. Oleh karena itu, kita dapat menyimpulkan bahwa *RESTful* mempunyai kecepatan akses lebih optimal daripada *SOAP*.

Dapat diamati *mean* penggunaan *throughput* antara *RESTful* dan *SOAP* pada setiap spesifikasi kecepatan *CPU* selalu menanjak. Spesifikasi kecepatan *CPU* juga dapat mendukung kecepatan akses, yang ditandai pada saat menggunakan kombinasi spesifikasi kecepatan

*CPU Quad-core* (spesifikasi rendah) terlihat *gap* semakin lebar.

Jadi, apabila *RESTful* lambat maka *SOAP* jauh lebih lambat. Oleh karena itu, kita dapat menyimpulkan bahwa *RESTful* mempunyai kecepatan akses lebih optimal daripada *SOAP*. *Mean gap throughput RESTful* dan *SOAP* dapat dicermati pada Gambar 11.

Pada hasil analisis di atas perbedaan perbedaan pada android dan spesifikasi kecepatan *CPU* dapat mendukung kecepatan akses. Selain itu ada faktor lain yang mendukung kecepatan akses yaitu koneksi internet dari pengguna. Dengan demikian dikatakan bahwa penggunaan *RESTful* lebih optimal dalam kecepatan akses.

#### 4. Kesimpulan

Kesimpulan yang dapat diambil berdasarkan penelitian yang telah dilakukan pada penerapan *RESTful* dan *SOAP web service* dengan jumlah data layanan 86 data. Dapat disimpulkan bahwa *web service* berbasis *RESTful* memiliki *throughput* yang lebih sedikit daripada *SOAP*, yang menyebabkan kecepatan akses *RESTful* jauh lebih cepat yaitu dua kali lebih cepat daripada kecepatan akses *SOAP*. Selain *RESTful web service* pada android dan spesifikasi kecepatan *CPU* dapat mendukung kecepatan akses.

Penelitian ini hanya menggunakan *request POST* dan terfokus pada kecepatan akses *web service*. Belum ada diskusi lebih mendalam mengenai *request GET, PUT, DELETE* maupun *OPTIONS* dan diskusi mengenai keamanan data dan *otentikasi* antar sistem, sehingga perlu ada peneliti kembali mengenai keamanan dan *otentikasi* pada *web service*.

#### Daftar Pustaka

- [1] D. Wijonarko And B. W. R. Mulya, "Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode Restful Pada Sistem Informasi Akademik Politeknik Kota Malang," *Smatika J.*, Vol. 8, No. 02, Pp. 63–66, 2018.
- [2] N. K. Ariasih And I. M. Gede Sri Artha, "Rancang Bangun Stiki Class Facilities E-Complaint," *Lontar Komput. J. Ilm. Teknol. Inf.*, Vol. 8, No. 2, Pp. 101–111, 2017.
- [3] A. Bora And T. Bezboruah, "A Comparative Investigation On Implementation Of Restful Versus Soap Based Web Services," *Int. J. Database Theory Appl.*, Vol. 8, No. 3, Pp. 297–312, 2016.
- [4] M. Ali, M. F. Zolkipli, J. M. Zain, And S. Anwar, "Mobile Cloud Computing With Soap And Rest Web Services," *J. Phys. Conf. Ser.*, Vol. 1018, No. 1, Pp. 1–9, 2018.
- [5] M. S. Warvante, "A Survey On Developing Web Services With Soap And Rest," *Int. J. Adv. Res. Comput. Commun. Eng.*, Vol. 6, No. 10, Pp. 118–124, 2017.
- [6] M. G. L. Putra and M. I. A. Putera, "Analisis Perbandingan Metode SOAP Dan REST Yang Digunakan Pada Framework Flask Untuk Membangun Web Service," vol. XIV, no. 2, pp. 1–7, 2019.
- [7] D. Rathod, "Performance Evaluation Of Restful Web Services And Soap / WsdL Web Services," *Int. J. Adv. Res. Comput. Sci.*, Vol. 8, No. 7, Pp. 415–420, 2017.
- [8] A. Rahmatulloh, H. Sulastri, And R. Nugroho, "Keamanan Restful Web Service Menggunakan
- [9] J. Nas. Tek. Elektro Dan Teknol. Inf., Vol. 7, No. 2, Pp. 131–137, 2018.
- [10] A. Memeti, F. Imeri, And B. Çiço, "Rest Vs. Soap: Choosing The Best Web Service While Developing In-House Web Applications," *J. Nat. Sci. Math. Ut.*, Vol. 2, No. 3, Pp. 63–68, 2017.
- [11] P. Wulandari, S. Soim, And M. Rose, "Monitoring Dan Analisis Qos (Quality Of Service) Jaringan Internet Pada Gedung Kpa Politeknik Negeri Sriwijaya Dengan Metode Drive Test," *Politek. Negeri Sriwij.*, Vol. 4, No. 1, Pp. 153–160, 2017.
- [12] A. Dudhe And S. Sherekar, "Critical Analysis Of Performance Optimization Of Mobile Web Services In Cloud Environment," *Int. Conf. Commun. Electron. Syst.*, Pp. 355–360, 2018.
- [13] A. L. Azhari And R.- Tanone, "Analisis Penerapan Single Page Application Menggunakan Teknologi Ajax Dan Rest Api (Studi Kasus :Sistem Informasi Reservasi Wisma Tamu Uksw)," *J. Tek. Inform. Dan Sist. Inf.*, Vol. 3, No. 1, Pp. 116–126, 2018.
- [14] A. Widodo, H. R. D. Putranti, And Nurchayati, "Pengaruh Kualitas Sistem Aplikasi Dan Kualitas Informasi Terhadap Kepuasan Pengguna Sistem Aplikasi Rts (Rail Ticketing System)," *Jurnal Media Ekon. Dan Manaj.*, Vol. 31, No. 2, Pp. 160–181, 2016.
- [15] Global Stashot, "Asosiasi Penyelenggara Jasa Internet Indonesia (Apjii)," [Online] Available At: [Http://Dailysocial.Id](http://Dailysocial.Id) [Diakses 30 Maret 2019], 2017.
- [15] M. Lachgar and A. Abdelmounaim, "Survey of Mobile Development Approaches," 2015, no. November.