

Penerapan RESTFul Web Service Untuk Optimalisasi Kecepatan Akses Pada Aplikasi Berbasis Android

by Lailin Nur Asiyah

Submission date: 16-Jan-2020 08:39AM (UTC+0700)

Submission ID: 1242441471

File name: Kecepatan_Akses_Pada_Aplikasi_Berbasis_Android_160403010021.docx (575.38K)

Word count: 3030

Character count: 18529

PENERAPAN ¹RESTFUL WEB SERVICE UNTUK OPTIMALISASI KECEPATAN AKSES PADA APLIKASI BERBASIS ANDROID

Lailin Nur Asiyah¹, Muhammad Priyono Tri. S², Abdul Aziz³

¹²³Teknik Informatika, Fakultas Sains & Teknologi, Universitas Kanjuruhan Malang

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

Abstract

In development, android applications must pay attention to web service access speed that optimal. Two web service architectures that commonly used are SOAP and REST. Systems that use the principles of REST are called "RESTful". The application of SOAP is still not optimal, because it uses of wasteful throughput. Throughput affects the speed of web service access. So that in this study discussed optimizing the speed of access to web services using RESTful. The RESTful test shows an average value of minimum throughput on Android Pie = 1736.8596 bps. While on SOAP the maximum average value obtained on Android Lollipop = 11640.4432 bps. Besides on Android, on CPU speed specification. RESTful shows the average value of minimum throughput on a combination of Octa-core CPU speed specifications = 2489.6253 bps. While in SOAP the maximum average value is obtained in a combination of quad-core CPU speed specifications = 12570 bps. So, RESTful is much faster than SOAP.

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* boros. *Throughput* berpengaruh pada kecepatan akses *web service*. Sehingga pada penelitian ini dibahas mengenai optimalisasi kecepatan akses *web service* menggunakan RESTful. Pada uji RESTful menunjukkan nilai rata - rata minimum *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 spesifikasi kecepatan CPU, RESTful menunjukkan nilai rata - rata minimum *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. Dengan demikian, RESTful jauh lebih cepat daripada SOAP.

Kata kunci: RESTful, SOAP, Web Service, Android

© 2020 Jurnal JOINTECS

1. Pendahuluan

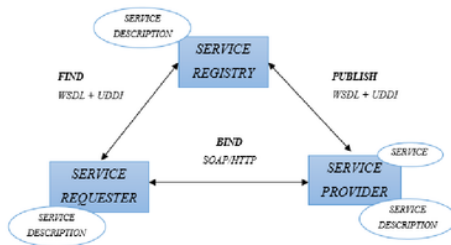
Perkembangan teknologi informasi dan komunikasi terus berkembang, memberikan pengaruh besar terhadap organisasi maupun individu. Yang bertujuan untuk memenuhi kebutuhan informasi maupun komunikasi bagi pengguna. Kemudahan akses informasi antar perangkat dan media adalah suatu fungsi penting dalam pengembangan aplikasi untuk menjadi lebih besar, maka ¹⁹erlukan integrasi sistem yang saling terhubung dalam berbagai macam platform serta perangkat yang ada 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]. Dua arsitektur pada web service yang secara umum

²⁰sering digunakan yaitu Simple Object Access Protocol (SOAP) dan Representational State Transfer (REST). REST merupakan architectural design untuk mengidentifikasi resource, REST menggunakan Uniform Resource Identifier (URI) dan HTTP, mendukung format data seperti JSON dan XML [4]. 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. Prinsip-prinsip ini membuat RESTful menjadi hemat throughput dan mempunyai kecepatan yang optimal. Model arsitektur RESTful web service dapat digambarkan seperti pada Gambar 1.



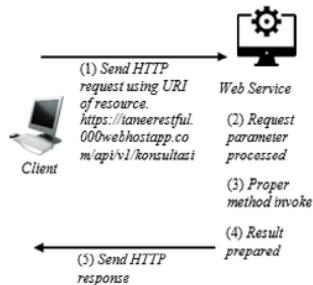
Gambar 1. Model arsitektur RESTful web service

1 SOAP merupakan web service yang menggunakan protocol berbasis Extensible Markup Language (XML) 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 sisi server. Hal ini membuat SOAP menjadi boros dalam pemakaian throughput. Karena, client tidak dapat mengidentifikasi pesan informasi secara langsung dari URI dan HTTP. Model arsitektur SOAP web service dapat digambarkan seperti pada Gambar 2.

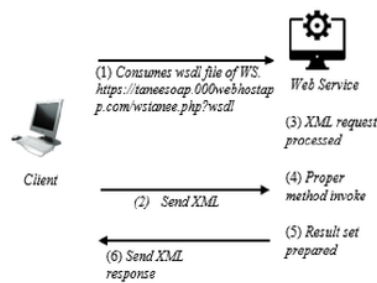


Gambar 2. Model arsitektur SOAP web service

Sedangkan, perbandingan langkah-langkah routing request dan response client ke server pada web service berbasis RESTful dan SOAP ditunjukkan pada Gambar 3. dan Gambar 4.



Gambar 3. Langkah – langkah routing pada RESTful web service



Gambar 4. Langkah – langkah routing pada SOAP web service

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 2 menjadi lebih optimal dibandingkan dengan SOAP” [5]. 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]. Hasil dari penelitian tersebut, dapat disimpulkan bahwa throughput berpengaruh pada kecepatan akses web service erat kaitannya dengan masalah efektivitas dan biaya koneksi internet. Penggunaan throughput yang boros menjadi persoalan dalam proses request dan response data dari web service [6]. Sehingga, semakin optimal kecepatan akses web service akan diikuti dengan meningkatnya kepuasan penggunaannya. Banyak pengguna menggunakan koneksi internet pada smartphone untuk mengakses teknologi 1 informasi maupun komunikasi. Android atau smartphone merupakan perangkat mobile yang paling banyak 5 digunakan untuk mengakses internet sebesar 72, 9%, berdasarkan survey yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) [7]. Dan survei pada tanggal 17 Juni 2019 yang menyatakan bahwa android adalah platform yang paling populer bagi para developer, yaitu sebanyak 83% developer aplikasi bergerak [8].

2. Metode Penelitian

2.1. Rancangan Penelitian

4 Penggunaan metode dalam 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*) [9]. Adapun skema model penelitian disajikan pada tabel dibawah:

Tabel 1. Skema Tabel Penelitian

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

Tujuan 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 akhirnya 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 dibawah ini.

Tabel 2. Layanan aplikasi pelayanan pertanian "TANEE"

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

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 16e atas. Dan spesifikasi kecepatan CPU minimum *Quad-core 1.0 GHz Cortex-A53*. Pada bahan 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

3. Hasil dan Pembahasan

3.1. Deskripsi Data

Desain pada penelitian merupakan penelitian eksperimen dengan desain *posttest only control design*. 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. Sedangkan, Gambar 6 menunjukkan *throughput* menggunakan SOAP dengan *time* 0.910644 seconds, *size* 1752 bytes dan *throughput* 1925 bps.

```
#RESTFUL GET TENAGA KERJA
#SIZE
Size Header      : 143 bytes
Size Download    : 358 bytes
Size Request     : 180 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      : 143 bytes
Size Download    : 1752 bytes
Size Request     : 180 bytes
Size Upload      : 0 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

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.



Gambar 7. Tampilan layanan dari aplikasi pelayanan pertanian “TANEe”

3.2. Pengujian Dan Perbandingan Deskriptif Pada Android

3.2.2. Pengujian Pada Android

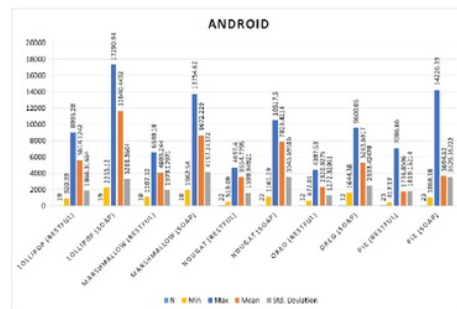
Pengujian android dibagi menjadi 5 kelompok yaitu Lollipop, Marshmallow, Nougat, Oreo dan Pie. Untuk mengetahui perbandingan antara *RESTful* dan *SOAP* dilakukan pengujian pada penggunaan *throughput*. Hasil pengujian *RESTful* dan *SOAP* pada Android dapat dicermati pada Tabel 3. Tabel 3 adalah nilai *minimum*, *maximum*, *mean* dan *std.deviation* pada proses pengujian delapan puluh enam kali menggunakan *RESTful* dan *SOAP* pada penggunaan *throughput*.

Tabel 3. Hasil Pengujian pada android.

N o	Android	Kelas	N	Min (bps)	Max (bps)	Mean (bps)	Std. Deviation
1	Lollipop	RESTful	19	920.39	8995.28	5614.1242	1868.31634
		SOAP	19	225.12	1729.094	11640.4432	3288.36040
2	Marshmallow	RESTful	10	110.75	6549.18	4083.2440	1973.17971
		SOAP	10	196.25	1375.462	8672.2290	4157.24372
3	Nougat	RESTful	22	513.09	4697.40	3554.7795	1599.94921
		SOAP	22	116.119	1051.750	7829.4214	3545.69516
4	Oreo	RESTful	12	672.01	4397.53	2320.9275	1272.32051
		SOAP	12	164.438	9600.05	5263.6917	2533.42478
5	Pie	RESTful	23	417.17	7086.66	1736.8596	1810.15214
		SOAP	23	106.818	1422.639	3694.6200	3529.35722

3.2.2. Perbandingan Kecepatan *RESTful* dan *SOAP* Pada Android

Berdasarkan hasil pengujian menyatakan penggunaan *throughput RESTful* dan *SOAP*. Pada *RESTful* diperoleh nilai rata – rata *minimum throughput* pada Pie = 1736.8596 *bps*. Sedangkan *SOAP* diperoleh nilai rata – rata *maximum* pada Lollipop = 11640.4432 *bps*. Berdasarkan hasil pengujian *RESTful* dan *SOAP* pada android secara rata – rata dapat disimpulkan bahwa *RESTful* memiliki *throughput* lebih sedikit daripada *SOAP* yang menyebabkan kecepatan akses *RESTful* lebih cepat daripada kecepatan akses *SOAP*. Hasil perbandingan pengujian pada android diperoleh nilai *minimum*, *maximum*, *mean* dan *std.deviation* dapat dicermati pada Gambar 8.



Gambar 8. Grafik perbandingan pada android

3.3. Pengujian Dan Perbandingan Deskriptif Pada Spesifikasi Kecepatan CPU

3.3.1. Pengujian Pada Spesifikasi Kecepatan CPU

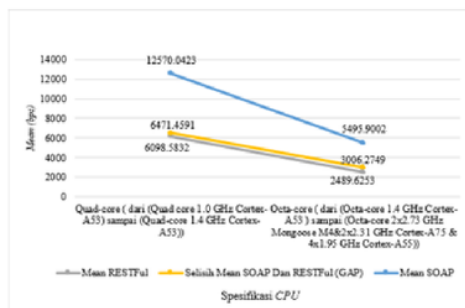
Pengujian spesifikasi kecepatan *CPU* dibagi menjadi 2 kelompok, yaitu kombinasi spesifikasi kecepatan *CPU Quad-core* dan *Octa-core*. Untuk mengetahui perbandingan antara *RESTful* dan *SOAP* dilakukan pengujian pada penggunaan *throughput*. Hasil pengujian pada spesifikasi kecepatan *CPU* dapat dicermati pada Tabel 4. Tabel 4 adalah nilai *minimum*, *maximum*, *mean* dan *std.deviation* pada proses pengujian delapan puluh enam kali menggunakan *RESTful* dan *SOAP* pada penggunaan *throughput*.

Tabel 4. Hasil pengujian pada spesifikasi CPU

N o	Spesifikasi CPU	Kelas	N	Min (bps)	Max (bps)	Mean (bps)	Std. Deviation
1	Kombinasi Quad-core	RESTful	22	4331.97	8995.28	6098.5832	1258.80472
		SOAP	22	9540.96	1729.094	12570.0423	2033.59873
2	Kombinasi Octa-core	RESTful	64	417.17	4697.40	2489.6253	1647.13719

Karena nilai *Sig.* dari *RESTful* dan *SOAP* = .000 < α (0.05) maka, hal ini menunjukkan ada perbedaan yang signifikan antara *RESTful* dan *SOAP*. Dapat disimpulkan bahwa kecepatan akses pada penggunaan *RESTful* lebih optimal dari pada penggunaan *SOAP*.

Dapat diamati rata – rata kecepatan antara *RESTful* dan *SOAP* pada setiap spesifikasi kecepatan *CPU* selalu menanjak. Spesifikasi kecepatan *CPU* juga mendukung kecepatan akses, yang ditandai pada saat menggunakan kombinasi *Quad-core* (rendah) terlihat *gap* semakin lebar, sedangkan pada saat menggunakan kombinasi *Octa-core* (tinggi) terlihat *gap* semakin sempit yang menunjukkan, semakin tingginya spesifikasi kecepatan *CPU* 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*. *Mean gap throughput RESTful dan SOAP* dapat dicermati pada Gambar 11.



Gambar 11. Grafik *mean gap* pada Spesifikasi *CPU*

Pada hasil analisis di atas perbedaan perbedaan pada android dan spesifikasi kecepatan *CPU* dapat mendukung kecepatan akses. Penelitian ini juga membuktikan penelitian sebelumnya dimana kecepatan *RESTful* dua kali lipat lebih cepat daripada *SOAP* [5]. 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. Beberapa arsitektur pada *web service* yaitu *SOAP*, *WSDL* dan *UDDI* yang secara umum sering digunakan yaitu *Simple Object Access Protocol (SOAP)* dan *Representational State Transfer (REST)* [10]. Menurut [11] kinerja *RESTful web service* lebih cepat dibandingkan dengan *SOAP web service*.

4. Kesimpulan Dan Saran

4.1. Kesimpulan

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.

4.2. Saran

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* yang dapat menerapkan konsep *Single Sign On (SSO)* [12]. Maupun *JSON Web Token (JWT) HMAC SHA-512* [13].

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] 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.
- [5] 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.
- [6] 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.
- [7] Global Stashot, "Asosiasi Penyelenggara Jasa Internet Indonesia (APJII)," [Online] Available

- at : <http://dailysocial.id> [Diakses 30 Maret 2019], 2017.
- [8] D. Ramel, "Survey Shows Most Mobile Development Still Done with Native Tools," [Online] Available at : <http://adtmag.com/articles/2019/06/17/jetbrains-survey.aspx> [Diakses 23 Desember 2019]], 2019. .
- [9] Sugiyono, ¹³ *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2015.
- [10] M. Ali, M. F. Zolkipli, J. M. Zain, and S. Anwar, "Mobile Cloud ²² Computing with SOAP and REST Web Services," *J. Phys. Conf. Ser.*, vol. 1018, no. 1, 2018.
- [11] M. S. Warvante, "A Survey on ¹⁸ Developing Web Services with SOAP and REST," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 6, no. 10, pp. 118–124, 2017.
- [12] Z. Musliyana, T. Y. Arif, and R. Munadi, "Peningkatan Sistem Keamanan Autentikasi Single Sign On (SSO) Menggunakan Algoritma AES dan One-Time Password Studi Kasus: SSO Universitas Ubudiyah Indonesia," *J. Rekayasa Elektr.*, vol. 12, no. 1, p. 21, 2016.
- [13] ²¹ A. Rahmatulloh, H. Sulastri, and R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 2, 2018.

Penerapan RESTFul Web Service Untuk Optimalisasi Kecepatan Akses Pada Aplikasi Berbasis Android

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

2%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

j-ptiik.ub.ac.id

Internet Source

1%

2

www.scribd.com

Internet Source

1%

3

Submitted to University of Queensland

Student Paper

<1%

4

journal.institutpendidikan.ac.id

Internet Source

<1%

5

repository.fe.unj.ac.id

Internet Source

<1%

6

Submitted to UIN Sunan Gunung Djati Bandung

Student Paper

<1%

7

ojs.amikom.ac.id

Internet Source

<1%

8

www.ingenio-web.it

Internet Source

<1%

9

Submitted to Fakultas Ekonomi Universitas

Indonesia

Student Paper

<1 %

10

e-jurnal.pelitanusantara.ac.id

Internet Source

<1 %

11

jurnal.umk.ac.id

Internet Source

<1 %

12

klik.ulm.ac.id

Internet Source

<1 %

13

etheses.iainponorogo.ac.id

Internet Source

<1 %

14

eprints.walisongo.ac.id

Internet Source

<1 %

15

docobook.com

Internet Source

<1 %

16

detikponsel.com

Internet Source

<1 %

17

mafiadoc.com

Internet Source

<1 %

18

Submitted to iGroup

Student Paper

<1 %

19

garuda.ristekdikti.go.id

Internet Source

<1 %

20

M. Miftakul Amin. "Interoperabilitas perangkat

<1 %

lunak menggunakan RESTful web service",
Register: Jurnal Ilmiah Teknologi Sistem
Informasi, 2018

Publication

21

ejournal.st3telkom.ac.id

Internet Source

<1 %

22

Submitted to Universitas Siliwangi

Student Paper

<1 %

23

Submitted to Sriwijaya University

Student Paper

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On