

Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android

Achmad Fauzi Harismawan¹, Agi Putra Kharisma², Tri Afirianto³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹achmadfauziharismawan@gmail.com, ²agi@ub.ac.id, ³tri.afirianto@ub.ac.id

Abstrak

Dalam pengembangan aplikasi perangkat bergerak harus memperhatikan efisiensi dalam hal penggunaan sumber daya, selain itu aplikasi perangkat bergerak sangatlah sensitif terhadap *delay*. Salah satu faktor yang dapat menyebabkan *delay* adalah proses yang dilakukan pada *web service*. Kecepatan proses pada *web service* bergantung pada bahasa pemrograman yang digunakan, bahasa pemrograman juga dapat mempengaruhi penggunaan CPU (*Central Processing Unit*) dan *memory* pada *server*. Oleh karena itu, perlu dilakukan analisis performa *web service* untuk mengetahui perbedaan penggunaan CPU (*Central Processing Unit*), *memory*, dan kecepatan eksekusi bahasa pemrograman yang digunakan pada *web service*. Pada penelitian ini, bahasa pemrograman yang digunakan adalah Python, PHP dan Perl dengan menggunakan CGI (*Common Gateway Interface*) pada Apache *web server* diakses melalui perangkat bergerak dengan platform Android. Setelah dilakukan penelitian didapatkan hasil bahwa bahasa pemrograman Perl memiliki kecepatan eksekusi paling cepat sedangkan bahasa pemrograman Python memiliki penggunaan CPU (*Central Processing Unit*) dan *memory* paling sedikit.

Kata kunci: Web Service, Android, Python, PHP, Perl

Abstract

In the application development of mobile devices must pay attention to the efficiency in terms of resource usage, in addition mobile device applications are very sensitive to delay. One of the factors that can cause delay is the process done on the web service. The speed of the process in the web service depends on the programming language used, the programming language can also affect the use of CPU (Central Processing Unit) and memory on the server. Therefore, it is necessary to analyze the performance of web service to know the difference of CPU usage (Central Processing Unit), memory, and speed of execution of programming language used in web service. In this research, programming languages used are Python, PHP and Perl using CGI (Common Gateway Interface) on Apache web server accessed via mobile device with Android platform. After the experiment, it was found that Perl programming language has the fastest execution speed while the Python programming language has CPU usage (Central Processing Unit) and the least memory..

Keywords: Web Service, Android, Python, PHP, Perl

1. PENDAHULUAN

Dalam pengembangan perangkat lunak harus memperhatikan efisiensi dalam hal penggunaan sumber daya sistem tersebut, terutama pada pengembangan aplikasi perangkat bergerak. Dikarenakan keterbatasan sumberdaya yang ada pada perangkat bergerak, *programmer* harus handal dalam mengelola sumberdaya yang ada.

Selain itu aplikasi perangkat bergerak juga tidak bisa lepas dari koneksi data melalui

jaringan *internet*. Koneksi data dibutuhkan pada saat aplikasi perangkat bergerak akan berkomunikasi dengan *web service* pada saat melakukan sebuah operasi. Pada tahap ini sangat rawan terjadi *delay*. Sedangkan aplikasi perangkat bergerak itu sendiri sangat sensitif terhadap *delay*. *Delay* pada sebuah operasi akan mengakibatkan gangguan pada *user-experience* seorang *user* yang menggunakan aplikasi perangkat bergerak tersebut.

Terdapat beberapa parameter *web response time* yang dapat menyebabkan sebuah operasi mengalami *delay*, salah satunya adalah *server*

processing time yaitu waktu yang dibutuhkan oleh *server* untuk memproses *request* yang dikirimkan oleh *client*. Hal ini dapat bervariasi secara drastis berdasarkan proses yang dilakukan oleh *server* (Savoia, 2001). Proses yang dilakukan oleh *server* dapat berupa akses *database*, komputasi, verifikasi, dan autentifikasi. Untuk melakukan proses-proses tersebut, perlu dilakukan pengembangan *web service* yang efisien agar dapat meminimalisir *delay* yang terjadi pada *server processing*.

Performa pada *web service* dapat berbeda berdasarkan bahasa pemrograman yang digunakan (Sagayaraj, 2013). Oleh karena itu, dalam melakukan pengembangan *web service* kita perlu memilih bahasa pemrograman yang tepat. Bahasa pemrograman sendiri adalah bahasa buatan yang didesain untuk mengekspresikan komputasi yang dapat dilakukan oleh mesin, umumnya adalah komputer (Oguntunde, 2012). Bahasa pemrograman dapat digunakan untuk mengatur perilaku mesin, untuk mengekspresikan algoritme dengan benar, atau hanya sebagai alat bagi manusia untuk berkomunikasi dengan mesin.

Pada umumnya, komputer tidak dapat mengerti bahasa pemrograman *high-level* secara langsung. Agar komputer dapat mengerti program yang ditulis dengan bahasa pemrograman *high-level* dibutuhkan penerjemah yang digunakan untuk menerjemahkan program ke dalam bahasa mesin. Penerjemah tersebut dinamakan *language processors*. *Language processors* sendiri memiliki tiga kelas, yaitu *assembler*, *interpreter*, dan *compiler* (Oguntunde, 2012). Masing-masing *language processors* yang ada pada bahasa pemrograman memiliki perbedaan arsitektur yang digunakan dalam memproses suatu kode program sampai dapat dieksekusi oleh komputer. Hal ini menyebabkan perbedaan penggunaan sumber daya yang ada pada perangkat komputer, seperti jumlah *memory* dan waktu komputasi CPU (*Central Processing Unit*) yang dibutuhkan oleh bahasa pemrograman (Oguntunde, 2012). Pada *web service* hal tersebut dapat mempengaruhi *server response time* secara signifikan.

Dari permasalahan tersebut, maka perlu dilakukan penelitian untuk menentukan bahasa pemrograman mana yang lebih efisien dan memiliki performa lebih baik untuk digunakan pada *web service*. Hal tersebut dapat dilihat dari perbedaan penggunaan sumberdaya pada komputer, seperti *memory usage*, *CPU usage*,

dan *execution time* (Sagayaraj, 2013). Setelah pengembangan *web service* akan diperlukan aplikasi *client* untuk mengakses *web service* tersebut. Aplikasi *client* akan dikembangkan pada sistem operasi berbasis Android. Penulis memilih sistem operasi berbasis Android karena dengan menggunakan sistem operasi berbasis Android akan lebih mudah melakukan kostumisasi dan analisis terhadap paket data yang diterima maupun dikirim melalui aplikasi *client*.

2. LANDASAN KEPUSTAKAAN

2.1 Python

Python adalah bahasa pemrograman yang bersifat *open source*. Bahasa pemrograman ini dioptimalisasikan untuk *software quality*, *developer productivity*, *program portability*, dan *component integration* (Lutz, 2010). Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti *internet scripting*, *systems programming*, *user interfaces*, *product customization*, *numeric programming* dll. Python saat ini telah menduduki posisi 4 atau 5 bahasa pemrograman paling sering digunakan di seluruh dunia (Lutz, 2010).

Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman Python (Lutz, 2010):

1. *Multi Paradigm Design*
2. *Open Source*
3. *Simplicity*
4. *Library Support*
5. *Portability*
6. *Extendable*
7. *Scalability*

2.2 PHP

PHP adalah bahasa pemrograman yang biasa digunakan untuk *server-side scripting*. PHP merupakan bahasa pemrograman yang simpel namun *powerful* dan tepat untuk digunakan pada *web server* (Tatro, 2013). PHP dapat berjalan pada dapat berjalan pada semua sistem operasi yang sering digunakan, seperti UNIX, Windows, dan Mac OS.

Bahasa pemrograman PHP memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah

beberapa fitur yang ada pada bahasa pemrograman PHP:

1. *Multi Paradigm Design*
2. *Open Source*
3. *Simplicity*
4. *Framework Support*
5. *Library Support*
6. *Extendable*
7. *Portability*
8. *Familiarity*

2.3 Perl

Bahasa pemrograman Perl pada awalnya dikembangkan untuk manipulasi teks, namun seiring dengan berjalannya waktu Perl mulai digunakan untuk mengembangkan beberapa macam sistem, seperti sistem administrasi, pengembangan *web*, pemrograman jaringan, pengembangan GUI (*Graphical User Interface*), dll (Siever, 1998). Saat ini Perl merupakan salah satu bahasa pemrograman yang sering digunakan untuk mengembangkan perangkat lunak *web server*.

Bahasa pemrograman Perl memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman Perl:

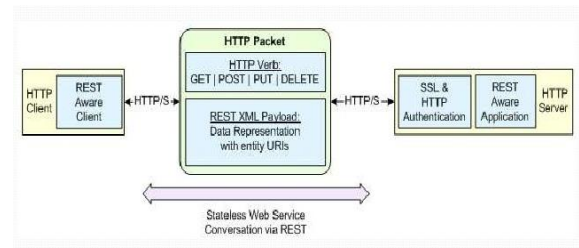
1. *Multi Paradigm Design*
2. *Open Source*
3. *Simplicity*
4. *Framework Support*
5. *Library Support*
6. *Extendable*
7. *Portability*

2.4 Web Service

Web service adalah sistem perangkat lunak yang didesain untuk membantu interaksi antara mesin dengan mesin pada sebuah jaringan (W3C, 2004). *Web service* pada awalnya menggunakan sebuah *interface* mesin untuk berkomunikasi dengan mesin yang lain (seperti WSDL). Namun seiring berjalannya waktu *web service* kini lebih banyak menggunakan SOAP dan REST *messages*, biasanya pesan akan dikonversi menggunakan HTTP dengan format JSON atau XML sesuai dengan standar *web* yang ada.

Pada penelitian ini akan menggunakan arsitektur *web service* berbasis REST. REST sendiri adalah arsitektur *web service* yang dimodelkan berdasarkan data yang disediakan, diakses, dan dimodifikasi pada *web* (Hamad,

2010). Pada *web service* dengan arsitektur REST, data dan fungsional dianggap sebagai *resources*, dan *resources* tersebut diakses menggunakan *Uniform Resources Identifiers* (URIs), biasanya berupa *link* pada *web*. Diagram arsitektur REST *web service* dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur REST Web Service
Sumber: Hamad (2010)

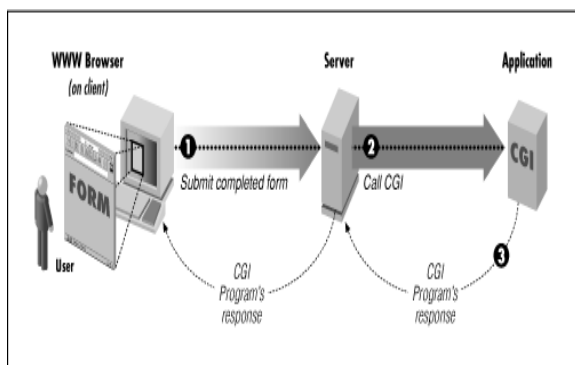
1. **HTTP Client.** Merupakan *client* yang akan mengakses *web service*.
2. **HTTP Packet.** HTTP *packet request* dapat berupa GET, POST, PUT, atau DELETE. Sedangkan, HTTP *packet response* dapat berupa XML maupun JSON *payload*.
3. **HTTP Server.** Merupakan *server* tempat *web service* diimplementasikan.

Terdapat beberapa HTTP *method* yang dapat diimplementasikan pada REST *web service*, diantaranya:

1. **GET.** *Method* yang digunakan untuk menerima *resources*.
2. **POST.** *Method* yang digunakan untuk membuat *resources*.
3. **PUT.** *Method* yang digunakan untuk meng-update *resources*.
4. **DELETE.** *Method* yang digunakan untuk menghapus *resources*.

2.5 Common Gateway Interface (CGI)

Common Gateway Interface (CGI) merupakan bagian dari *web server* yang dapat berkomunikasi dengan program lain yang berjalan pada *server* (Kathuria, 2014). Dengan menggunakan CGI, *web server* dapat memanggil program yang ada maupun yang sedang berjalan pada *server*. *Web server* akan mengirimkan *user data* ke program menggunakan CGI, kemudian program akan memproses data dan memberikan respon kembali ke *web server*.



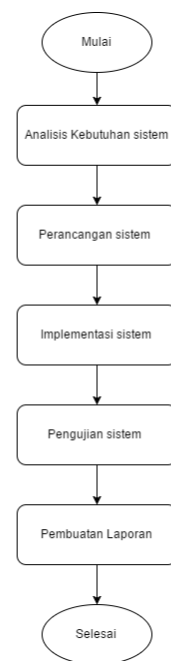
Gambar 2. Model Sederhana CGI
Sumber: Kathuria (2014)

Gambar 2 menjelaskan model sederhana dari *Common Gateway Interface* (CGI). Pada awalnya *user* akan mengirimkan *form* yang telah di *submit* ke *server*, kemudian *server* akan memanggil program menggunakan CGI, selanjutnya program akan memproses data dan mengirimkan respon kembali ke *server* dan melanjutkannya ke komputer *user*.

Common Gateway Interface (CGI) telah menjadi standar dalam pengembangan aplikasi pada *web server*, banyak programmer telah memodifikasi *script* CGI sesuai dengan kebutuhannya (Kathuria, 2014). CGI juga dapat berjalan pada berbagai *platform web server* yang ada di *world wide web* (WWW). Selain itu CGI mempunyai beberapa kekurangan, CGI mempunyai beberapa isu keamanan. Dengan menggunakan CGI akan membuat seseorang dapat menjalankan program pada *server*. Oleh karena itu program yang menggunakan CGI harus dipisahkan pada *directory* tertentu, sehingga seseorang tidak dapat menjalankan program menggunakan CGI diluar *directory* tersebut.

3. METODOLOGI

Metodologi penelitian menjelaskan metode yang digunakan dalam analisis perbandingan performa *web service* menggunakan bahasa pemrograman Python, PHP, dan Perl pada *client* berbasis Android. Tahapan metodologi penelitian dapat dilihat pada Gambar 3.



Gambar 3. Gambar Blok Metodologi Penelitian

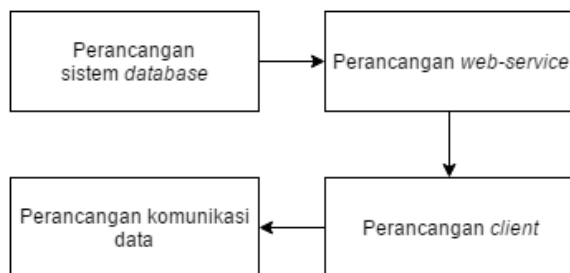
3.1 Studi Literatur

Mempelajari literatur dari berbagai bidang ilmu yang berhubungan dengan analisis performa *web service* menggunakan bahasa pemrograman Python, PHP, dan Perl pada *client* berbasis Android adalah sebagai berikut:

1. Kajian pustaka dari penelitian sebelumnya
2. Pengembangan *web service*
3. Pengembangan aplikasi perangkat bergerak berbasis Android
4. Bahasa pemrograman Java
5. Bahasa pemrograman Python
6. Bahasa pemrograman PHP
7. Bahasa pemrograman Perl

3.2 Perancangan Sistem

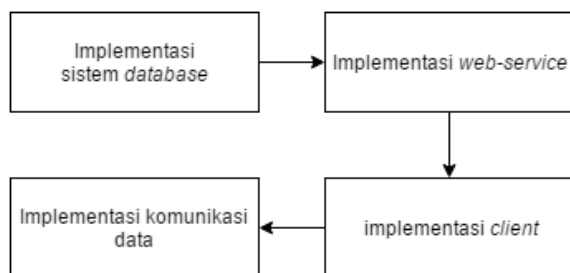
Perancangan sistem dilakukan untuk mengetahui rancangan langkah kerja dari sistem secara menyeluruh, baik dari segi model maupun dari segi arsitektur untuk mempermudah implementasi sistem. Tahap-tahap perancangan sistem dapat dilihat pada Gambar 4.



Gambar 4. Tahap-tahap Perancangan Sistem

3.3 Implementasi Sistem

Implementasi sistem adalah tahap membangun sistem berdasarkan pada perancangan sistem dan menerapkan hal yang telah didapat pada tahap studi literatur. Tahap-tahap implementasi sistem dapat dilihat Gambar 5.



Gambar 5. Tahap-tahap Implementasi Sistem

3.4 Pengujian Sistem

Pengujian performa *web service* terbagi menjadi dua elemen yaitu kuantitatif dan kualitatif. Kuantitatif meliputi penggunaan CPU (*Central Processing Unit*), Penggunaan RAM (*Random Access Memory*), kecepatan eksekusi perintah, perbandingan SOAP / REST *message*, dan jumlah baris kode program. Sedangkan kualitatif meliputi metodologi, *scalability*, *security*, dan *maintainability* (Sagayaraj, 2013). Pengujian sistem pada penelitian ini akan menggunakan pengujian elemen kuantitatif dengan parameter penggunaan CPU (*Central Processing Unit*), penggunaan RAM (*Random Access Memory*), dan kecepatan eksekusi perintah.

Terdapat dua pengujian yang akan dilakukan, yaitu pengujian dari sisi *client* dan *web service*. Pengujian pada sisi *client* dilakukan dengan cara mengirimkan *request* ke *web service* dan menghitung *response time* dari *web service*. Pada sisi *web service* sendiri juga dilakukan analisis penggunaan CPU (*Central Processing Unit*) dan penggunaan RAM

(*Random Access Memory*) pada saat proses *request* sedang berlangsung. Pengujian tersebut dilakukan pada bahasa pemrograman Python 2.7, PHP 5.6.28, dan Perl 5.24.

4. PERANCANGAN

4.1 Analisis Kebutuhan Sistem

Proses analisis kebutuhan ini diawali dengan penjabaran gambaran umum sistem, identifikasi aktor, penjabaran tentang daftar kebutuhan dan dimodelkan ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengujian system.

4.2 Perancangan Sistem

Proses perancangan sistem memiliki lima tahap, yaitu perancangan arsitektur sistem, perancangan *database*, perancangan aplikasi Android, perancangan antarmuka sistem, dan perancangan *web service*.

4.3 Perancangan Pengujian Sistem

Perancangan pengujian sistem dilakukan untuk memberi gambaran pengujian sistem yang akan dilakukan sekaligus dengan perangkat-perangkat yang akan digunakan dalam pengujian. Pengujian nantinya akan dibagi menjadi dua sisi, yaitu pengujian dari sisi *client* dan pengujian dari sisi *server*. Parameter-parameter yang digunakan dalam pengujian dapat dilihat pada Tabel 1 (Sagayaraj, 2013).

Tabel 1. Paramater Pengujian Sistem

No.	Parameter	Satuan	Keterangan
1.	<i>Response time (client)</i>	<i>Milisecond (ms)</i>	Waktu respon yang dibutuhkan oleh <i>web service</i>
2.	<i>CPU usage (server)</i>	Persentase (%)	Penggunaan CPU pada <i>server</i>
3.	<i>RAM usage (server)</i>	<i>Byte (B)</i>	Penggunaan RAM pada <i>server</i>

Pengujian pada sisi *client* digunakan untuk menguji kecepatan eksekusi perintah dari masing-masing *web service* yang telah diimplementasikan menggunakan bahasa

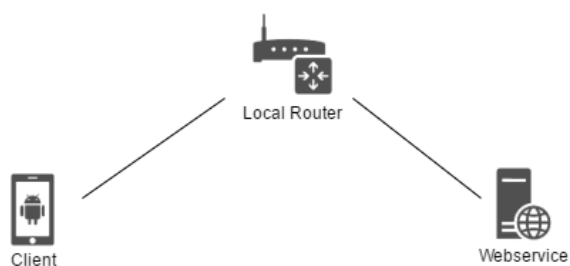
pemrograman Python, PHP, dan Perl. Aplikasi *client* diimplementasikan pada sistem operasi Android yang berfungsi untuk mencatat waktu pengiriman *request* dan penerimaan *response*, sehingga akan didapatkan total waktu proses yang dibutuhkan oleh *web service*. Sedangkan pengujian pada sisi *server* digunakan untuk menguji penggunaan CPU dan penggunaan RAM saat proses eksekusi perintah pada masing-masing bahasa pemrograman. Pengujian penggunaan CPU dan penggunaan RAM pada *server* akan dilakukan dengan menggunakan *Resource Monitor* yang ada pada sistem operasi Windows.

Pengujian nantinya juga akan dibagi menjadi beberapa tahap, masing-masing tahap terdapat perbedaan jumlah data yang akan diproses, hal ini dilakukan untuk mengetahui perbedaan performa masing-masing bahasa pemrograman dalam memproses data dengan jumlah tertentu. Tahap-tahap tersebut dapat dilihat pada Tabel 2.

Tabel 2. Tahap-tahap Pengujian Sistem

Tahap	Banyak Data	Ukuran Data	Total Ukuran Data
1.	5.000	1KB	5MB
2.	10.000	1KB	10MB
3.	15.000	1KB	15MB
4.	20.000	1KB	20MB
5.	40.000	1KB	40MB

Pengujian nantinya akan dilakukan pada jaringan nirkabel lokal, dengan hanya menggunakan dua *host* yang berperan sebagai *client* dan *server*, hal itu dilakukan untuk menghindari perbedaan beban pada jaringan maupun *router* pada saat pengujian sedang berlangsung. Rancangan lingkungan pengujian sistem dapat dilihat pada Gambar 6.



Gambar 6. Rancangan Lingkungan Pengujian

5. HASIL PENGUJIAN DAN ANALISIS

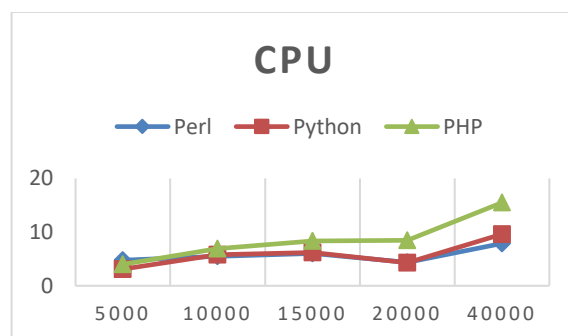
Dalam pengujian yang akan dilakukan pada sistem adalah pengujian penggunaan CPU (*Central Processing Unit*), RAM (*Random Access Memory*), dan kecepatan eksekusi perintah dari masing-masing bahasa pemrograman yang digunakan sebagai *web service* untuk aplikasi *client* berbasis Android. Kemudian dilihat hasil analisisnya menggunakan bahasa program manakah yang memiliki performansi *web service* yang tinggi untuk digunakan pada aplikasi *client* berbasis Android.

5.1 Pengujian Penggunaan CPU pada Server

Pengujian Penggunaan CPU pada *server* ini dibagi menjadi beberapa tahap, yaitu pengujian dengan 5.000, 10.000, 15.000, 20.000, dan 40.000 data pada masing-masing Bahasa pemrograman. Setiap data yang diuji berukuran 1KB sehingga total data yang diproses adalah 5MB, 10MB, 15MB, 20MB, dan 40MB. Hasil pengujian dapat dilihat pada Tabel 3 dan Gambar 7.

Tabel 3. Hasil Pengujian Penggunaan CPU pada Server

Banyak data	Perl	Python	PHP
5.000	4,76 %	3,11 %	4,01 %
10.000	5,49 %	5,76 %	6,9 %
15.000	6,01 %	6,2 %	8,37 %
20.000	4,37 %	4,3 %	8,48 %
40.000	7,89 %	9,62 %	15,49 %



Gambar 7. Grafik Penggunaan CPU pada Server

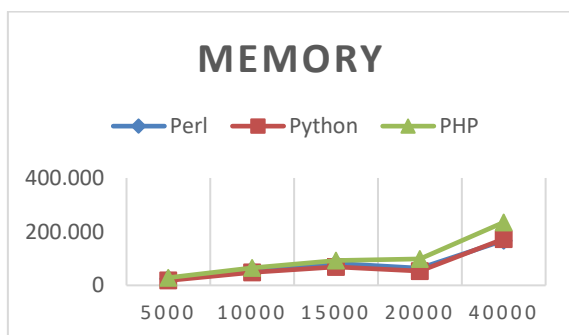
5.2 Pengujian Penggunaan Memory pada Server

Pengujian Penggunaan *Memory* pada *server* ini dibagi menjadi beberapa tahap, yaitu pengujian dengan 5.000, 10.000, 15.000, 20.000, dan

40.000 data pada masing-masing Bahasa pemrograman. Setiap data yang diuji berukuran 1KB sehingga total data yang diproses adalah 5MB, 10MB, 15MB, 20MB, dan 40MB. Hasil pengujian dapat dilihat pada Tabel 4 dan Gambar 8.

Tabel 4. Hasil Pengujian Penggunaan *Memory* pada *Server*

Banyak data	Perl	Python	PHP
5.000	24.880 KB	18.148 KB	28.600 KB
10.000	58.388 KB	48.568 KB	65.264 KB
15.000	82.028 KB	69.156 KB	92.412 KB
20.000	64.092 KB	53.584 KB	98.024 KB
40.000	167.080 KB	172.976 KB	234.824 KB



Gambar 8. Grafik Penggunaan *Memory* pada *Server*

5.3 Pengujian Kecepatan Eksekusi pada *Mobile Client*

Pengujian kecepatan eksekusi pada aplikasi *mobile client* ini dibagi menjadi beberapa tahap, yaitu pengujian dengan 5.000, 10.000, 15.000, 20.000, dan 40.000 data pada masing-masing Bahasa pemrograman. Setiap data yang diuji berukuran 1KB sehingga total data yang diproses adalah 5MB, 10MB, 15MB, 20MB, dan 40MB. Hasil pengujian dapat dilihat pada Tabel 5 dan Gambar 9.

Tabel 5. Hasil Pengujian Kecepatan Eksekusi pada Aplikasi *Mobile Client*

Banyak data	Perl	Python	PHP
5.000	1.228 ms	1.316 ms	1.312 ms
10.000	2.987 ms	3.179 ms	3.479 ms
15.000	5.062 ms	5.210 ms	5.965 ms
20.000	5.585 ms	5.808 ms	6.911 ms
40.000	9.337 ms	9.644 ms	13.966 ms



Gambar 9. Grafik Kecepatan Eksekusi pada Aplikasi *Mobile Client*

5.4 Analisis Hasil Pengujian

Berdasarkan pada pengujian yang telah dilakukan dengan melihat penggunaan CPU (*Central Processing Unit*), RAM (*Random Access Memory*) dan kecepatan eksekusi perintah masing-masing bahasa pemrograman dengan menggunakan data sebesar 5.000, 10.000, 15.000, 20.000, dan 40.000 data. Untuk mengetahui bahasa pemrograman mana yang memiliki performa lebih baik untuk digunakan pada aplikasi *client* berbasis Android. Maka perlu dilakukan analisis dari hasil pengujian yang telah dilakukan seperti Tabel 6, Tabel 7, Tabel 8, Tabel 9 dan Tabel 10.

Tabel 6. Analisis Hasil Pengujian Menggunakan 5.000 Data

Parameter	Hasil		
	Perl	Python	PHP
CPU	4,76 %	3,11 %	4,01 %
	-	✓	-
Memory	24.880 KB	18.148 KB	28.600 KB
	-	✓	-
Kecepatan Eksekusi	1.228 ms	1.316 ms	1.312 ms
	✓	-	-

Keterangan: tanda ✓ menunjukan yang terbaik.

Tabel 7. Analisis Hasil Pengujian Menggunakan 10.000 Data

Parameter	Hasil		
	Perl	Python	PHP
CPU	5,49 %	5,76 %	6,9 %
	✓	-	-
Memory	58.388 KB	48.568 KB	65.264 KB
	-	✓	-
Kecepatan Eksekusi	2.987 ms	3.179 ms	3.479 ms
	✓	-	-

Keterangan: tanda ✓ menunjukkan yang terbaik.

Tabel 8. Analisis Hasil Pengujian Menggunakan 15.000 Data

Parameter	Hasil		
	Perl	Python	PHP
CPU	6,01 %	6,2 %	8,37 %
	✓	-	-
Memory	82.028 KB	69.156 KB	92.412 KB
	-	✓	-
Kecepatan Eksekusi	5.062 ms	5.210 ms	5.965 ms
	✓	-	-

Keterangan: tanda ✓ menunjukkan yang terbaik.

Tabel 9. Analisis Hasil Pengujian Menggunakan 20.000 Data

Parameter	Hasil		
	Perl	Python	PHP
CPU	4,37 %	4,3 %	8,48 %
	-	✓	-
Memory	64.092 KB	53.584 KB	98.024 KB
	-	✓	-
Kecepatan Eksekusi	5.585 ms	5.808 ms	6.911 ms
	✓	-	-

Keterangan: tanda ✓ menunjukkan yang terbaik.

Tabel 10. Analisis Hasil Pengujian Menggunakan 40.000 Data

Parameter	Hasil		
	Perl	Python	PHP
CPU	7,89 %	9,62 %	15,49 %
	✓	-	-
Memory	167.080 KB	172.976 KB	234.824 KB
	✓	-	-
Kecepatan Eksekusi	9.337 ms	9.644 ms	13.966 ms
	✓	-	-

Keterangan: tanda ✓ menunjukkan yang terbaik.

6. KESIMPULAN

Berdasarkan pada penelitian yang telah dilakukan, didapatkan kesimpulan sebagai berikut:

1. Analisis dilakukan dengan cara menjalankan operasi *string concatenation* pada bahasa pemrograman Python, PHP, dan Perl. Operasi tersebut diimplementasikan pada *web service* di jaringan lokal dan diakses menggunakan aplikasi *client* berbasis Android. Dari operasi tersebut didapatkan hasil penggunaan CPU dan *memory* pada sisi *server*, serta kecepatan eksekusi perintah pada sisi *client*.
2. Pada pengujian dengan menggunakan 5.000 dan 20.000 data, bahasa pemrograman Perl memiliki kecepatan eksekusi paling cepat, sedangkan bahasa pemrograman Python memiliki penggunaan CPU dan *memory* paling sedikit. Pada pengujian dengan menggunakan 10.000 dan 15.000 data, bahasa pemrograman Perl memiliki kecepatan eksekusi paling cepat dan penggunaan CPU paling sedikit, sedangkan bahasa pemrograman Python memiliki penggunaan *memory* paling sedikit. Pada pengujian dengan menggunakan 40.000 data, bahasa pemrograman Perl memiliki kecepatan eksekusi paling cepat dan penggunaan CPU serta *memory* paling sedikit.
3. Bahasa pemrograman Perl mempunyai rata-rata kecepatan eksekusi paling cepat, sedangkan bahasa pemrograman Python mempunyai rata-rata penggunaan *memory*

dan CPU paling sedikit dibandingkan dengan bahasa pemrograman lainnya.

DAFTAR PUSTAKA

- Kathuria, K., Kapoor, C. & Adlakha, A., 2014. Common Gateway Interface. International Journal of Science and Research (IJSR), 3(10), pp. 1733-1735.
- Kreger, H., 2001. Web Services Conceptual Architecture.
- Lutz, M., 2010. Programming Python. Fourth Edition ed. Sebastopo: O'Reilly Media, Inc.
- Oguntunde, B. O., 2012. Comparative Analysis of Some Programming Languages. Transnational Journal of Science and Technology, Volume 2, pp. 107-118.
- Sagayaraj, S. & Kumar, M. S., 2013. Performance Evaluation of Web Services in C#, JAVA, and PHP. International Journal of Computer Science and Business Informatics, 7(1).
- Savoia, A., 2001. Web Page Response Time 101. s.l.:STQE.
- Siever, E., Spainhour, S. & Patwardhan, N., 1998. Perl in a Nutshell. First Edition ed. Sebastopol: O'Reilly & Associates, Inc.
- Tatroe, K., MacIntyre, P. & Lerdorf, R., 2013. Programming PHP. Third Edition ed. Sebastopol: O'Reilly Media, Inc..
- W3C, 2004. Web Services Architecture. W3C Working Group Note.