Web_Service_Pencarian_Koleksi_Pustaka.pdf

WORD COUNT	2424	TIME SUBMITTED	27-SEP-2021 02:59PM
		PAPER ID	77074046

WEB SERVICE PENCARIAN KOLEKSI REPOSITORY PERPUSTAKAAN PADA APLIKASI EPRINTS

Meizano Ardhi Muhammad[1*], Mardiana[2], Yessi Mulyani[3], Dikpride Despa[4]
[1, 2,3,4] *Jurusan Teknik Elektro, Universitas Lampung*

[1*] *meizano@eng.unila.ac.id*

ABSTRAK

Institutional Repository (IR) adalah sarana penting bagi universitas untuk menyimpan bahan-bahan digital, dan termasuk upaya untuk pengorganisasian, akses dan pendistribusian yang baik untuk karya-karya akademik yang dimilikinya. Tantangan dari penggunaan IR adalah bagaimana menjamin koleksi yang dimiliki oleh universitas, khususnya perpustakaan, dapat didistribusikan dengan baik terhadap civitas akademika yang berhak mengaksesnya. API (Application Programming Interface) Web Service adalah salah satu cara yang dapat digunakan sebagai cara akses data oleh perangkat lunak tanpa terikat dengan perangkat lunak itu sendiri. Sehingga, penggunaan API Web Service dapat mengurangi waktu yang diperlukan untuk pengembangan perangkat lunak karena cara akses data seragam. Tujuan penelitian ini adalah untuk mengembangkan API web service IR yang berfungsi untuk mengoptimalkan layanan koleksi IR, khususnya untuk pencarian dokumen pada aplikasi Eprints yang tidak mendukung API Web Service. API Web Service IR dikembangkan menggunakan metode Rapid Application Development (RAD) yang memungkinkan dilakukannya iterasi revisi dengan cepat. Pembuatan API Web Service menggunakan bahasa PHP dengan dua skenario yaitu generasi data secara real-time dan generasi data periodik. Berdasarkan hasil uji skenario, diketahui bahwa skenario generasi data secara real-time membutuhkan waktu yang lebih lama dalam memperoleh data. Sedangkan, generasi data periodik memiliki performansi yang cepat walau pun jika ada data baru yang masuk, sebelum periode generasi data, menjadi tidak tersedia. Karena penggunaan di setiap universitas berbeda, kedua skenario ini harus dipertimbangkan dalam memilih model generasi data yang diperlukan. Dengan penerapan API Web Service untuk pencarian koleksi repository pada aplikasi EPrints, layanan perpustakaan Unila telah dihadirkan dalam bentuk KiosK untuk layanan bersifat mandiri.

Kata kunci: Institutional Repository, perpustakaan, API, web service, generasi data

PENDAHULUAN

Institutional Repository (IR) perguruan tinggi adalah sekumpulan layanan yang ditawarkan oleh universitas kepada komunitas anggotanya untuk manajemen dan diseminasi materi digital yang dibuat oleh institusi dan komunitas anggotanya (Lynch, 2003). IR merupakan sarana yang digunakan oleh universitas untuk menyimpan bahan-bahan digital yang dimilikinya, dan termasuk upaya untuk pengorganisasian, akses dan pendistribusian yang baik untuk karya-karya akademik yang dimilikinya. Tantangan dari penggunaan IR adalah bagaimana menjamin koleksi yang dimiliki oleh universitas, khususnya perpustakaan, dapat didistribusikan dengan baik terhadap civitas akademika yang berhak mengaksesnya.

Perkembangan teknologi informasi membuat cara akses informasi tidak lagi melalui komputer desktop saja, tetapi melalui berbagai melalui berbagai macam perangkat teknologi informasi yang berbeda. Pengembangan perangkat lunak sistem informasi IR untuk setiap perangkat yang berbeda dapat memakan waktu yang lama. Dan, walau hanya sebuah perangkat saja yang didukung, pendekatan multiplatform tetap paling efisien karena rendahnya rintangan yang harus dilalui (Heitkotter, Hanschke, & Majchrzak, 2013).

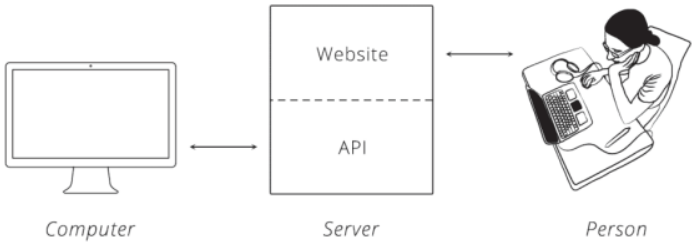
Salah faktor yang dapat dipertimbangkan untuk mempercepat pengembangan perangkat lunak sistem informasi IR adalah dengan membuat metode akses data sama. API (*Application Programming Interface*) Web Service adalah salah satu cara yang dapat digunakan sebagai cara akses data oleh perangkat lunak tanpa terikat dengan perangkat lunak itu sendiri. Layanan ini dikembangkan dengan tujuan menyederhanakan layanan kepada pengguna data (aplikasi, mesin, dsb) oleh pengembang (Massimiliano Rak, Venticinque, & Martino, 2012). Sehingga, penggunaan API Web Service dapat mengurangi waktu yang diperlukan untuk pengembangan perangkat lunak karena cara akses data seragam. Tujuan penelitian ini adalah untuk mengembangkan API web service IR yang berfungsi untuk mengoptimalkan layanan koleksi IR, khususnya untuk pencarian dokumen pada aplikasi Eprints yang tidak mendukung API Web Service.

TINJAUAN PUSTAKA

API Web Service

API (Application Programming Interface) adalah sebuah antarmuka pemrograman antar aplikasi yang memungkinkan dilakukannya pertukaran data antara dua aplikasi yang mungkin berbeda melalui metode akses yang standar (seperti melalui protokol web) sehingga interoperabilitas dapat terjadi. Ada 4 model komunikasi yang umum dipakai dalam API, yaitu: Polling, Long Polling, Webhooks, dan Subscription Webhooks. (Carson, 2015)

Salah satu bentuk API yang sering dipakai adalah web service. Web Service adalah sebuah perangkat lunak atau sistem yang menyediakan akses kepada layanannya melalui alamat yang berada di *World Wide Web*. Alamat ini dikenal sebagai URI atau URL. Intinya, web service menawarkan informasinya dalam format yang dapat dipahami atau di *parsing* oleh aplikasi-aplikasi lain. Contohnya: Flickr API dan Google Maps API.



Gambar 3. Komunikasi dengan API Server

Web service menggunakan HTTP atau HTTPS untuk bertukar informasi. Ketika aplikasi, atau bisa disebut klien, ingin berkomunikasi dengan web service, aplikasi mengirimkan HTTP *request*. Web service kemudian membalas dengan HTTP *response*. Di dalam permintaan, sebagian besar informasi yang diminta dilewatkan melalui URL, sebagai path dalam URL dan/atau parameter URL.

Institusional Repository (IR)

Institusional Repository (IR) atau Repositori Institusi merupakan koleksi digital yang menampilkan dan menyimpan karya intelektual perguruan tinggi (PT), baik dari satu institusi maupun komunitas perguruan tinggi (Crow, 2002). Repositori berkaitan dengan perpustakaan digital dan

memerlukan manajemen teknologi informasi. Repositori menjadi sarana penting bagi PT/institusi untuk menyimpan bahan-bahan digital yang dimilikinya, dan termasuk upaya untuk preservasi jangka panjang, pengorganisasian, akses dan pendistribusian yang baik untuk karya-karya akademik yang dimilikinya.

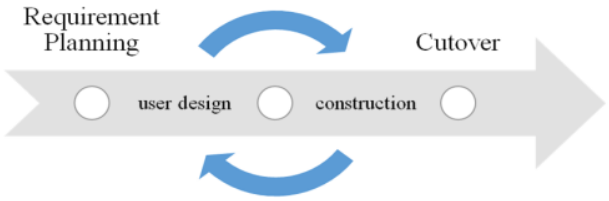
Koleksi repositori tidak merupakan seluruh koleksi perpustakaan, sehingga dibutuhkan pengetahuan terhadap konsep IR secara umum dan sejumlah kebijakan yang akan diberlakukan untuk pelaksanaan repositori tersebut. Kebijakan tersebut seperti misalnya untuk menentukan bahan-bahan apa saja yang termasuk ke dalam koleksi repositori suatu perguruan tinggi/institusi dan kebijakan akses dalam penyebaran informasi khususnya yang berkaitan dengan karya intelektual lokal PT/institusi tersebut. Tantangan berikutnya ketika repositori sudah berjalan pada institusi adalah bagaimana manajer repositori mampu mengembangkan dan menjamin sustainabilitasnya, antara lain terkait dengan indeksasi, visibilitas termasuk juga peningkatan dan *sharing* kualitas/kuantitas konten dengan institusi lain.

Aplikasi EPrints

Terdapat berbagai aplikasi untuk mengelola koleksi repositori sehingga lebih mudah diakses oleh pemustaka, baik yang berbayar maupun gratis antara lain: *Dispace, Green Stone, GDL,* dan *EPrints* (Narendra, 2014). EPrints merupakan perangkat lunak perpustakaan *digital* yang dikembangkan oleh University of Southampton, England United Kingdom. *EPrints* berbasis *opensource,* sehingga dapat dimodifikasi dan disesuaikan dengan kebutuhan lokal *(http://www.eprints.org/uk/)* (EPrints, 2017). Perpustakaan Unila menerapkan *EPrints* untuk mengelola Koleksi Repository Karya Ilmiah mahasiswa sejak tahun 2012 hingga saat ini.

METODE

API Web Service IR dikembangkan menggunakan metode Rapid Application Development (RAD) yang memungkinkan dilakukannya iterasi revisi dengan cepat antara proses user design dan construction. Pembuatan API Web Service menggunakan bahasa PHP dengan dua skenario yaitu generasi data secara real-time dan generasi data periodik. Kedua skenario tersebut dikembangkan untuk mengetahui performansi berdasarkan kebutuhan di lapangan.



Gambar 2. Rapid Application Development

Phase 1: User Requirements

Pustakawan, pemustaka, staf TI Perpustakaan, dan pengembang membahas dan menyepakati kebutuhan bisnis, lingkup proyek, kendala, dan persyaratan sistem. Ini berakhir ketika tim setuju pada isu-isu kunci dan memperoleh otorisasi manajemen untuk melanjutkan.

Phase 2: user design

Pustakawan, pemustaka, staf TI Perpustakaan, dan pengembang mengembangkan model dan prototipe yang mewakili proses semua sistem, input, dan output. CASE Tool digunakan untuk memudahkan penerjemahan kebutuhan pengguna ke dalam model kerja. Desain web service dilakukan dengan interaktif yang berkesinambungan yang memungkinkan pengguna untuk memahami, memodifikasi, dan akhirnya menyetujui model kerja dari sistem. Format dari web service pada tahap ini ditentukan.

Phase 3: construction

Pengembangan API Web Service dilakukan dan pengguna terus berpartisipasi dan masih dapat menyarankan perubahan atau perbaikan dari API yang dikembangkan. Terutama keterkaitan teknologi yang dapat digunakan untuk implementasi dan pemanfaatan API Web Service. Kegiatan yang dilakukan adalah adalah pemrograman dan pengembangan aplikasi, coding, integrasi unit dan pengujian.

Phase 4: cutover

Cutover dilakukan ketika implementasi dianggap cukup atau sudah tenggat waktu. Kegiatan konversi data, pengujian, dan pembuatan petunjuk petunjung API Web Service dilakukan. Dibandingkan dengan metode tradisional, seluruh proses dikompresi. Akibatnya, sistem baru dibangun, disampaikan, dan ditempatkan dalam operasi lebih cepat.

HASIL DAN PEMBAHASAN

Desain API Web Service

Berdasarkan kebutuhan layanan data, API Web Service menyediakan himpunan data yang yang memuat informasi mengenai pustaka yang disajikan. Layanan data diberikan memiliki informasi sebagai berikut:

- Nama
- NPM
- Judul
- Tahun
- Fakultas
- Universitas
- URL

Untuk kebutuhan performansi, format data dipilih dalam bentuk JSON karena ukurannya yang lebih kecil dari XML. Himpunan data disediakan sekaligus sehingga dapat dimanfaatkan untuk melakukan pencarian offline jika diperlukan. Format data dalam bentuk JSON memiliki format sebagai berikut:

Gambar 4.Format JSON API Web Service

Data yang dihasilkan, disajikan dalam bentuk array JSON sebanyak data yang ada dan dapat diakses melalui indeks angka. Untuk mengakses secara spesifik atribut yang ada pada data, dapat dilakukan dengan menggunakan atribut key dan value yang tersemat pada data. Walau hanya tersedia satu data, tetap parsing data terhadap format JSON dilakukan melalui pembacaan sebagai array JSON.

Format Request dan Response API Web Service

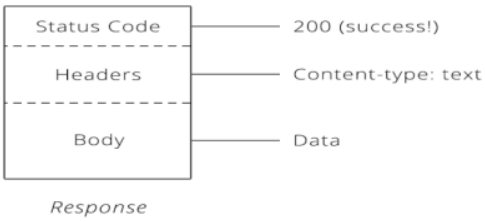
Bentuk data yang dikirimkan melalui HTTP *request* memiliki empat komponen, yaitu: URL, Method, Header, dan Body. Pengiriman permintaan API Web Service menggunakan ketentuan khusus:

- URL : http://example.com/**api/**
- Method: POST

Gambar 5. Struktur HTTP Request

Sedangkan, bentuk data yang dikirimkan kembali memiliki tiga komponen, yaitu: kode status (umumnya 200 untuk sukses), Header, dan Body. Balasan dari API Web Service menggunakan ketentuan:

- Status Code: 200
- Headers: Content-type: JSON
- Body: Data dalam format JSON



Gambar 6. Struktur HTTP Response

Pembuatan Web Service dengan PHP

Real-Time

Berkas PHP digunakan untuk membuat response HTTP atas permintaan layanan API Web Service pada saat permintaan diajukan.

```
<?php
$dataJSON = json_encode($data);

// Menghasilkan string data JSON
echo $dataJSON;
?>
```

Gambar 7. Skrip PHP untuk Real-Time Service

Generasi Data

Berkas PHP digunakan untuk melakukan generasi berkas JSON yang dapat diakses untuk kebutuhan layanan data.

```
<?php
$dataJSON = json_encode($data);
```

```
// Generasi berkas ws.json
$berkasJSON = fopen("ws.json", "w") or die("Tidak bisa dibuka.");
fwrite($berkasJSON, $dataJSON);
fclose($berkasJSON);
?>
```

Gambar 8. Skrip PHP untuk Generasi Data dalam berkas ws.json

Pengujian

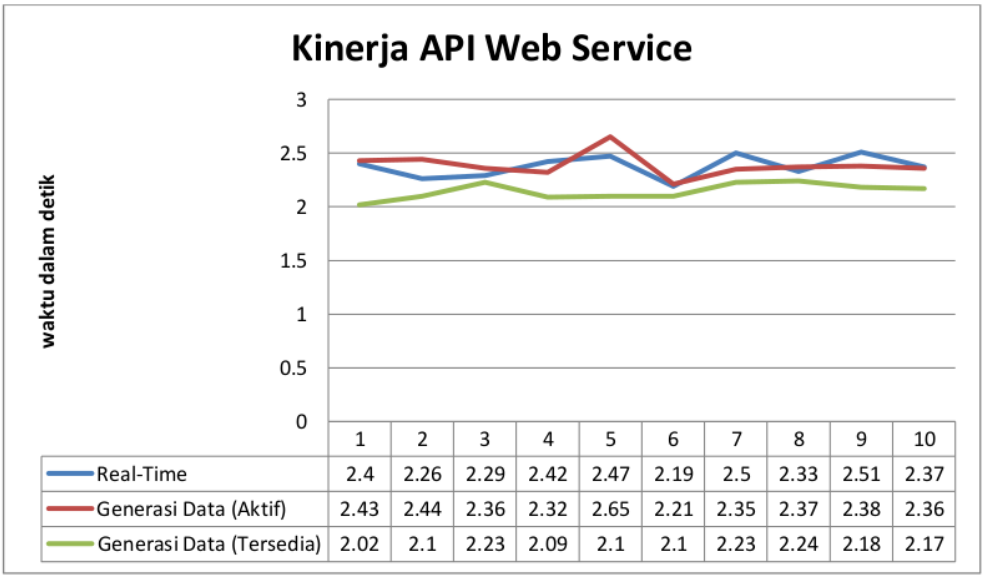
Layanan data yang dihasilkan sesuai dengan format JSON yang telah dirancang dan pengujian dilakukan terhadap perbedaan kinerja dari skenario generasi data dan real-time. Dilakukan pengukuran 10x terhadap pengaksesan layanan API Web Service sampai dengan kondisi DOMContentLoaded dengan ukuran data yang diterima sebesar 1.1MB.

Tabel 1. Kinerja API Web Service: Generasi data vs Real-Time (dalam detik)

Skenario	1	2	3	4	5	6	7	8	9	10	Rata-rata
Real-Time	2,4	2,26	2,29	2,42	2,47	2,19	2,5	2,33	2,51	2,37	2,374
Generasi Data (Aktif)	2,43	2,44	2,36	2,32	2,65	2,21	2,35	2,37	2,38	2,36	2,387
Generasi Data (Tersedia)	2,02	2,1	2,23	2,09	2,1	2,1	2,23	2,24	2,18	2,17	2,146

Berdasarkan hasil uji skenario, diketahui bahwa skenario generasi data secara real-time membutuhkan waktu yang lebih lama dalam memperoleh data dibandingkan generasi data saat data sudah tersedia. Sedangkan, generasi data periodik memiliki performansi yang cepat jika sedang tidak melakukan generasi data, tetapi cenderung lebih lambat dibanding Real-Time, hal ini disebabkan karena skrip harus mengakses perangkat penyimpanan untuk menulis berkas. Walau generasi data pada saat data sudah tersedia paling cepat dibanding yang lain, karena data tidak disajikan berdasarkan input data terbaru, jika ada data baru yang masuk sebelum periode generasi data, data baru menjadi tidak tersedia. Gambar 8 memperlihatkan perbandingan kinerja API web service.

Dalam implementasi nyata, tetap perlu diperhatikan bagaimana kebijakan penyediaan data terbaru dari perpustakaan. Misalnya, data baru tersedia pada Web Service setelah satu jam dimasukkan ke dalam basis data tidak mengganggu kegiatan operasional, generasi data merupakan pilihan yang tepat. Namun, apabila kebutuhan data harus langsung tersedia saat dimasukkan ke dalam basis data, seperti pada kondisi peminjaman buku yang baru dimasukkan ke basis data, disarankan untuk menggunakan layanan web service real-time.

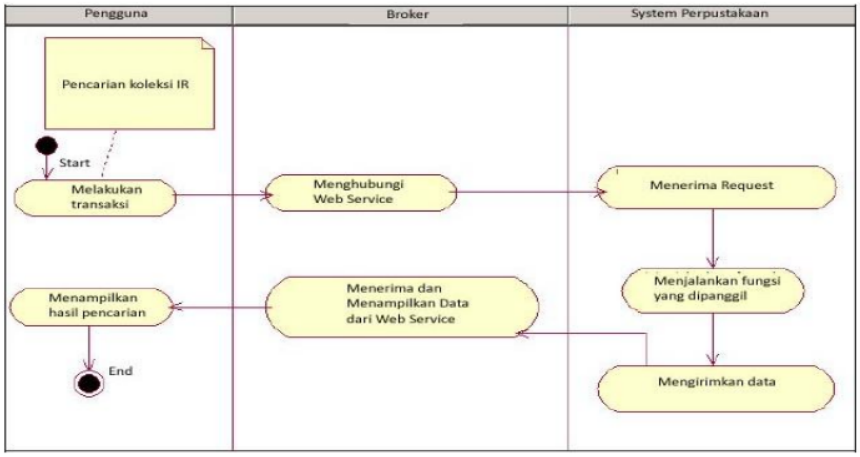


	1	2	3	4	5	6	7	8	9	10
Real-Time	2.4	2.26	2.29	2.42	2.47	2.19	2.5	2.33	2.51	2.37
Generasi Data (Aktif)	2.43	2.44	2.36	2.32	2.65	2.21	2.35	2.37	2.38	2.36
Generasi Data (Tersedia)	2.02	2.1	2.23	2.09	2.1	2.1	2.23	2.24	2.18	2.17

Gambar 9. Kinerja API Web Service

Implementasi Web Service pada aplikasi EPrints Perpustakaan

Activity Diagram Sistem Pencarian diperlihatkan pada gambar 9. berikut. Layanan Pencarian Koleksi Repository pada Aplikasi EPrints pada UPT Perpustakaan Unila diimplementasikan dalam bentuk KIOS-K, sehingga pengguna dapat mencari koleksi secara mandiri. Hasil pencarian pun dapat dibaca atau langsung dicetak secara mandiri pula. Gambar 10. memperlihatkan implementasi pada KIOS-K.



Gambar 9. Activity Diagram Pencarian Koleksi

Gambar 10. Implementasi Eprints pada KIOS-K

Pencarian Koleksi

Pencarian koleksi dilakukan melalui kotak input pencarian yang dapat dimasukkan tulisan dengan virtual keyboard, seperti pada gambar 11. Koleksi yang dicontohkan dalam hal ini adalah koleksi repository berupa artikel jurnal.

Gambar 11. Pencarian Koleksi

Hasil pencarian akan ditampilkan dalam bentuk tabel seperti pada gambar 12. Tabel memberikan informasi mengenai nomor ISBN/ISSN, Judul, Pengarang, Tahun, Penerbit, dan Tempat Terbit. Tombol yang bisa diakses adalah Detil, Abstrak, dan Cetak.

Gambar 12. Hasil Pencarian Koleksi Jurnal

KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa:

1. Layanan data real-time membutuhkan waktu yang lebih lama dalam memperoleh data dibanding generasi data pada saat data telah tersedia.
2. Generasi data pada saat data telah tersedia memiliki performansi yang cepat walau pun jika ada data baru yang masuk, sebelum periode generasi data aktif, menjadi tidak tersedia.
3. Dengan penerapan *Web Service* untuk pencarian koleksi repository pada aplikasi EPrints, layanan perpustakaan dalam hal pencarian koleksi IR bisa dihadirkan dalam bentuk KiosK untuk layanan bersifat mandiri.

DAFTAR PUSTAKA

Carson , R. (2015). *An Introduction to APIs*, An Introd. to EDCI 554, hal. 1–29

Crow, R. (2002). *The case for institutional repositories: A SPARC position paper*. Washington, D.C: The Scholarly Publishing & Academic Resources Coalition.

EPrints. (2017). *Building Repositories*. Diambil kembali dari EPrints: ://www.eprints.org/uk/

Heitkotter, H., Hanschke, S., & Majchrzak, T. A. (2013). Evaluating Cross-Platform Development Approaches for Mobile Applications. *Lecture Notes in Business Information Processing (LNBIP)*, 120-138.

Lynch, C. A. (2003, February). Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age. *ARL no. 226*, hal. 1-7. Diambil kembali dari http://www.arl.org/resources/pubs/br/br226/br226ir.shtml

Massimiliano Rak, R. A., Venticinque, S., & Martino, B. D. (2012). User Centric Service Level Management in mOSAIC Applications. Dalam M. Alexander, P. D'Ambra, A. Belloum, G. Bosilca, M. Cannataro, M. Danelutto, . . . J. Weidendorfer, *Euro-Par 2011 Workshops* (hal. 106-115). Berlin: Springer.

Narendra, A. P. (2014, Februari). Perpustakaan Digital dan Repositori Institusi Universitas. *Info Persadha: Media Informasi Perpustakaan Universitas Sanata Dharma. Vol. 12, No. 1.*

UPT Perpustakaan Universitas Lampung
Jl. S. Brojonegoro No. 1 Gedungmeneng, Rajabasa Bandar Lampung
http://sniper.unila.ac.id ; sniperunila@gmail.com ; library@kpa.unila.ac.id

9 772614 468002

SNIPer 2017
Semiloka Nasional Inovasi Perpustakaan

Web_Service_Pencarian_Koleksi_Pustaka.pdf

ORIGINALITY REPORT

19%
SIMILARITY INDEX

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

★text-id.123dok.com 5%
Internet

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		