

PEMANFAATAN TEKNOLOGI *WEB SERVICES* PADA PERTUKARAN DATA KATALOG ANTAR PERPUSTAKAAN

Endah Kristiani

Fakultas Teknik dan Ilmu Komputer Jurusan Teknik Informatika
Universitas Kristen Krida Wacana
endah.kristiani@ukrida.ac.id

Abstrak

Peran perpustakaan universitas di era kemajuan Teknologi Informasi perlu dikembangkan menuju ke arah integrasi data. Pembentukan jaringan perpustakaan dapat menjadi sarana pembelajaran komunitas yang efektif. Permasalahan yang dihadapi adalah *interoperability* atau kemampuan untuk mengintegrasikan aplikasi yang berbeda di antara perpustakaan dalam jaringan kerjasama. Untuk mengatasi permasalahan tersebut diperlukan teknologi *Web Service*, yaitu teknologi yang dapat mengintegrasikan aplikasi, bahasa pemrograman, dan *platform* yang berbeda melalui *http*, sehingga aplikasi-aplikasi yang berbeda tersebut dapat saling berkomunikasi dan bertukar data. Pada *paper* ini, dirancang suatu simulasi integrasi data katalog antara Perpustakaan A dan Perpustakaan B. *Web Services* yang menjembatani jaringan perpustakaan dibuat menggunakan PHP dengan menggunakan *NuSOAP class*, sedangkan *database* yang digunakan dalam simulasi ini adalah *Database MySQL* pada Perpustakaan A dan *Database Visual Foxpro* dengan Konektor *ODBC* pada Perpustakaan B.

Kata kunci: *Web service, NuSOAP, online catalogue* terintegrasi, interoperabilitas

Abstract

The role of university libraries in the era of information technology advances need to be developed towards the integration of data. Establishment of library networks can be an effective means of learning communities. Problems faced is the interoperability or the ability to integrate different applications in co-operation between libraries in the network. To overcome these problems required Web Service technology, ie technology that can integrate applications, programming languages, and different platforms via http, so that different applications can communicate and exchange data. In this paper, a simulation designed a catalog of data integration between the Library A and Library B. Web Services that bridges the library network is created using PHP using NuSOAP class, while the database used in this simulation is at the Library A MySQL Database, and Visual Foxpro Database with ODBC connector on the Library B.

Keyword: *Web service, NuSOAP, integrated online catalogue, interoperability*

1. PENDAHULUAN

1.1 Latar Belakang

Pembentukan jaringan perpustakaan dapat menjadi sarana pembelajaran komunitas yang efektif. Melalui jaringan tersebut, perpustakaan mempunyai kesempatan untuk memperluas layanannya. Untuk memenuhi hal tersebut diperlukan kesepakatan kerja sama dan teknologi integrasi informasi di mana data-data katalog pada perpustakaan

dalam jaringan kerjasama menjadi terintegrasi, sehingga memudahkan penggunaannya untuk mengakses informasi yang dibutuhkannya.

1.2 Perumusan Masalah

Permasalahan yang dihadapi adalah *interoperability* atau kemampuan untuk mengintegrasikan aplikasi yang berbeda di antara perpustakaan dalam jaringan kerjasama. Untuk mengatasi permasalahan tersebut diperlukan teknologi *Web Service*, yaitu teknologi yang dapat mengintegrasikan aplikasi, bahasa pemrograman, dan *platform* yang berbeda melalui *http*, sehingga aplikasi-aplikasi yang berbeda tersebut dapat saling berkomunikasi dan bertukar data.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk melakukan analisis, perancangan, implementasi, dan pengujian pertukaran data katalog antar perpustakaan dalam jaringan kerja sama dalam suatu model aplikasi, dengan memperhatikan beberapa hal yang diperlukan seperti arsitektur dan skenario implementasinya.

1.4 Metodologi Penelitian

Penelitian yang dilakukan merupakan studi kasus untuk mengintegrasikan dua perpustakaan, yaitu Perpustakaan A dan Perpustakaan B. Katalog Perpustakaan A disimulasikan dengan *database* MySQL dan Perpustakaan B dengan *database* Visual Foxpro dan Konektor ODBC. Tahapan pelaksanaan penelitian ini terdiri dari:

1) Analisis Kebutuhan Sistem

Mempelajari struktur *database* perpustakaan B dan mendapatkan data buku perpustakaan B. *Database* yang digunakan adalah *Database Visual Foxpro* dengan konektor ODBC.

Selanjutnya dilakukan pembelajaran terhadap katalog perpustakaan A pada *website* perpustakaan A. Selanjutnya dirancang suatu simulasi *database* dari perpustakaan A dengan MySQL.

2) Perancangan

Tahap perancangan dilakukan dengan langkah-langkah seperti berikut ini:

- a) Melakukan perancangan arsitektur pertukaran data dan arsitektur peminjaman antarperpustakaan dengan penjelasan tentang deskripsi skenario pertukaran data antar perpustakaan.
- b) Membuat *use case diagram*, *class diagram*, *sequence diagram*, *activity diagram*, dan *deployment diagram web service* pertukaran data katalog antarperpustakaan.
- c) Membuat *form* pengujian *input* dan *output* pemanggilan *web service*.

3) Implementasi

Implementasi dilakukan dengan membuat program aplikasi pencarian katalog buku terintegrasi yang membaca data katalog antara perpustakaan A dan B. Pemrograman dilakukan dengan menggunakan PHP dan NuSOAP untuk membuat *file* WSDL.

Simulasi dilakukan dengan menggunakan tiga perangkat komputer dan *file* WSDL sebagai program penghubung di lokasi alamat <http://192.168.1.2/perpusB/> untuk data katalog perpustakaan B dan <http://192.16.1.3:8888/perpusA/> untuk data katalog Perpustakaan A, serta aplikasi *client* yang memakai kedua *web service* perpustakaan di alamat <http://192.168.1.2/perpusws/>.

4) Pengujian

Pengujian dilakukan dengan menggunakan teknik:

a) *Black Box*

Pengujian dilakukan dengan menggunakan program aplikasi pencarian katalog buku terintegrasi untuk menguji kebenaran *input* dan *output*.

b) *Gray Box*

Pengujian dilakukan dengan menggunakan *automation tool* JMeter dan Jmeterv yang mempunyai kemampuan untuk menguji dengan teknik *Gray Box* melalui file WSDL. Pengujian dilakukan berdasarkan pada dua parameter berikut:

- Fungsionalitas
Melakukan pengujian parameter dan fungsi *request* serta *response*-nya menggunakan *tool* JMeter
- Performansi
Melakukan pengujian skalabilitas dan ketahanan *web service* dan menentukan karakteristik *performance* serta daya tahan operasi-operasi pada file WSDL dengan menggunakan *tool* JMeter. *Tool* akan menentukan *response time*, *latency* melalui profil *web service*. Untuk meningkatkan profil performansi, *tool* seharusnya dijalankan dalam durasi tertentu untuk mengukur daya tahan dan untuk menentukan skalabilitas dengan melakukan *bombarding target Web Services* menggunakan pesan SOAP yang bervariasi secara bersamaan.

2. KONSEP DASAR

2.1 *Web Service*

Web services menghadirkan paradigma arsitektur baru dalam aplikasi. *Web services* menerapkan layanan yang ada ke aplikasi lain (atau *web service* lain) melalui standar jaringan industri dengan *interface* dan protokol aplikasi. Suatu aplikasi dapat menggunakan layanan *web service* cukup dengan melakukan *request* melalui jaringan. *Web service* menampilkan *reusable software building block* melalui alamat URL.

Kemampuan yang disediakan *web service* terdiri dari beberapa kategori, termasuk:

- Fungsi, contohnya adalah suatu rumus untuk menghitung akar pangkat sebuah angka.
- Data, contohnya data katalog buku Amazon.
- Proses bisnis, contohnya penerimaan *order*, pengiriman *invoice*, dan lain-lain.

Teknologi *web service* menawarkan cara untuk menjembatani aplikasi-aplikasi yang ingin berkomunikasi dan bertukar data [1]. *Web Service* menyediakan standar untuk berkomunikasi antara berbagai aplikasi yang berbeda dan dapat berjalan pada *platform* maupun *framework* yang berbeda. Interoperabilitas *web service* menggunakan sekumpulan spesifikasi *open standard* yang terdiri dari HTTP, XML, SOAP, WSDL, dan UDDI.

Untuk dapat menjalankan fungsi tersebut, suatu *web service* memerlukan agen. Agen adalah potongan *software* atau *hardware* yang mengirim dan menerima pesan (*message*). Agen dapat ditulis dengan berbagai jenis bahasa pemrograman. Komunikasi memerlukan pembicara dan lawan bicara, dalam *web service* dikenal dengan istilah *providers* dan *requesters*. *Provider entity* adalah seseorang atau organisasi yang menyediakan agen untuk layanan fungsi tertentu sedangkan *requester entity* adalah seseorang atau organisasi yang menggunakan layanan yang disediakan *provider*. Agar dapat berkomunikasi dengan lancar, kedua entitas ini harus sepakat untuk menggunakan mekanisme yang sama dalam pertukaran pesan (*message*).

Mekanisme pertukaran pesan didokumentasikan di *Web Service Description* (WSD). WSD adalah spesifikasi mesin yang dapat mendefinisikan format pesan, tipe data, *protocol transport*, format serialisasi yang digunakan antara agen di *requester* dan agen di *provider* dan juga menentukan lokasi di mana *service* ditempatkan.

Secara umum langkah-langkah berikut ini diperlukan:

- 1) Entitas *requester* dan entitas *provider* saling mengetahui satu sama lain, atau setidaknya salah satu mengetahui yang lain.
- 2) Entitas *requester* dan entitas *provider* sama-sama setuju deskripsi layanan yang akan mengatur interaksi agen peminta dan penyedia.
- 3) Deskripsi layanan berdasarkan agen peminta dan penyedia
- 4) Agen peminta dan penyedia bertukar pesan.

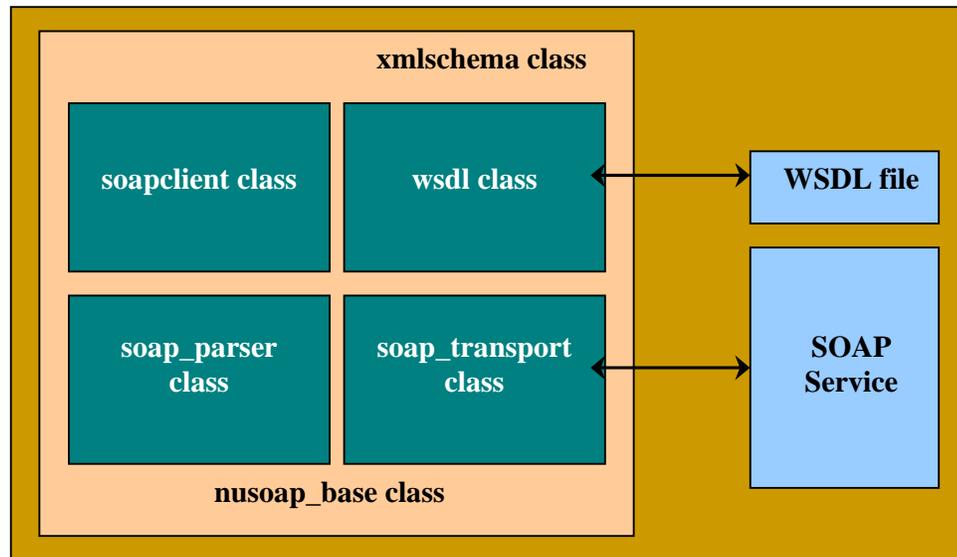
2.2 PHP NuSOAP

NuSOAP adalah sekumpulan kelas-kelas PHP yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP melalui HTTP [2]. Sebelumnya NuSOAP dikenal sebagai SOAPx4 yang didistribusikan oleh *NuSphere Corporation* (<http://www.nusphere.com>). SOAPx4 merupakan *open source*, berlisensi di bawah GNU LGPL. SOAPx4 telah digunakan sebagai standar beberapa *toolkits web services* untuk PHP, termasuk PEAR-SOAP dan *Active State software* proyek API *web service* sederhana. Salah satu keuntungan dari NuSOAP adalah bahwa bukan merupakan ekstensi PHP, tetapi program murni PHP. Hal ini berarti semua pengembang PHP, tanpa dibatasi oleh *web server* dan izin dapat menggunakan NuSOAP.

NuSOAP adalah komponen yang berbasis *toolkit web services*. NuSOAP digunakan sebagai kelas dasar yang menyediakan *utility methods*, seperti variabel dan *envelope serialization*, sama seperti memberi nama informasi dan memetakan tipe-tipe yang berbeda ke nama-nama yang berbeda. Interaksi *web service* dicapai melalui sebuah kelas *high-level client* yang disebut *soapclient*. Kelas *high-level* ini mengizinkan *user* menspesifikasikan *option*, seperti HTTP *authorization*, HTTP *proxy information*, sama seperti mengelola pengiriman dan penerimaan pesan SOAP itu sendiri [3].

Operasi SOAP dapat dieksekusi dengan menggunakan nama operasi yang ingin dieksekusi ke dalam *call() method*. Jika *service* yang dikonsumsi menyediakan suatu *file WSDL*, kelas *soapclient* mengambil URL dari *file WSDL* sebagai argumen *constructor*-nya. Kelas WSDL digunakan untuk menguraikan *file WSDL* dan mengekstrak semua data. Kelas WSDL mempunyai metode untuk mengekstraksi data berdasarkan per operasi atau per *binding*.

Soapclient menggunakan data ini dari *file WSDL* untuk melakukan *encode* parameter-parameter dan membuat SOAP *envelope* ketika *user* mengeksekusi sebuah panggilan ke *service* tersebut. Pada saat pemanggilan fungsi dilakukan, kelas *soapclient* memakai kelas *soap_transport_http* untuk mengirim pesan keluar dan menerima pesan yang masuk. Pesan yang masuk diuraikan menggunakan kelas *soap_parser*. Gambar 1 menunjukkan proses konsumsi SOAP *web service* memakai NuSOAP [4], [5].



Gambar 1. Proses konsumsi SOAP *web service* menggunakan NuSOAP

3. HASIL PENELITIAN DAN PEMBAHASAN

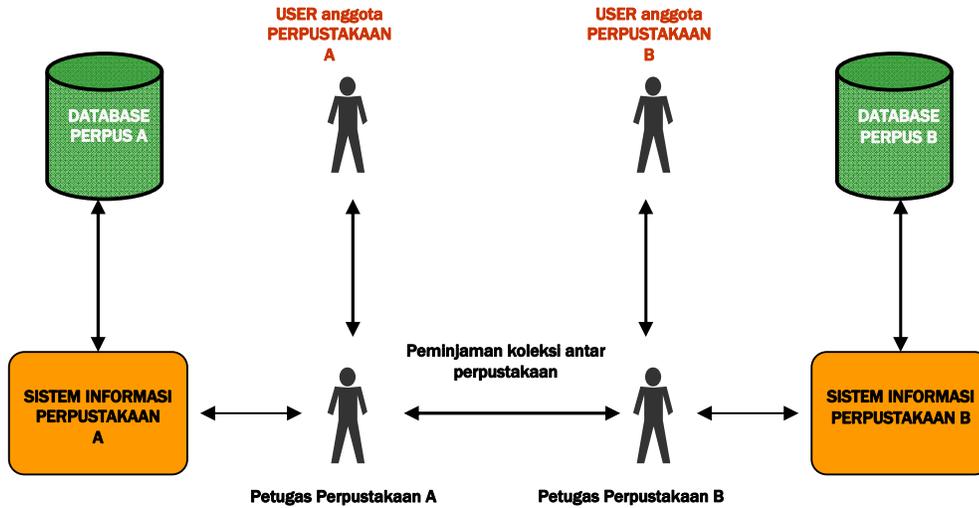
3.1 Hasil Penelitian

3.1.1 Proses Bisnis

Katalog bersama bertujuan untuk memudahkan pengguna mencari informasi buku pada Perpustakaan dalam jaringan kerja sama. Terdapat beberapa kesepakatan yang diperlukan untuk membangun aplikasi *web* katalog bersama dalam jaringan perpustakaan. Berikut ini adalah mekanisme yang diperlukan:

- **Mekanisme Tingkat kerjasama dan kesepakatan antar perpustakaan**
Diperlukan kerjasama dan kesepakatan berupa nota kesepakatan antarperpustakaan untuk membuat katalog bersama dan transaksi peminjaman antarperpustakaan.
- **Mekanisme Proses Transaksi Peminjaman dan Pengembalian**
Peminjaman dan Pengembalian buku dapat dilakukan dengan dua cara, tergantung dari nota kesepakatan:
 - *Inter library loan* dimana pengguna menghubungi perpustakaan masing-masing untuk meminjam buku dari perpustakaan lain dan petugas perpustakaan setempat meminjamkan buku ke perpustakaan yang dimaksud pengguna.
 - Pengguna menjadi anggota tamu pada perpustakaan lain dengan rekomendasi dari organisasi di mana pengguna berada.

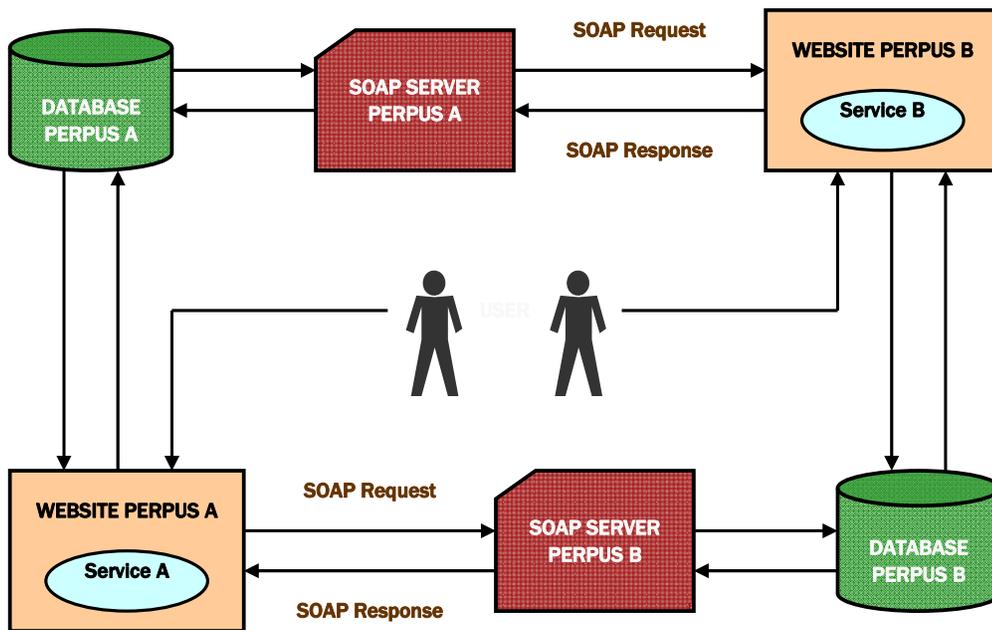
Skenario peminjaman dan pengembalian buku antar perpustakaan dapat digambarkan pada Gambar 2.



Gambar 2. Skenario peminjaman dan pengembalian antar perpustakaan

- **Mekanisme Proses Pertukaran Data**

Arsitektur proses pertukaran data menggunakan *web service* dilakukan dengan membuat aplikasi SOAP pada *Server* masing-masing sebagai program integrasi yang dapat dibaca *website* perpustakaan lain seperti digambarkan pada Gambar 3.



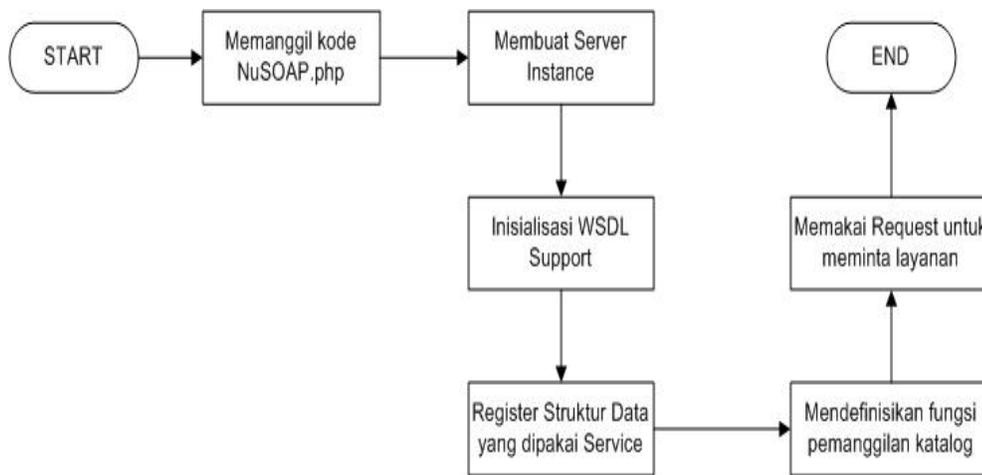
Gambar 3. Arsitektur katalog *web service*

Pengguna mengakses katalog B dari *website* perpustakaan A, aplikasi *web* A akan meminta (*request*) layanan SOAP Perpustakaan B. Layanan SOAP Perpustakaan B akan menjembatani pengambilan data dari *database*-nya kemudian me-*response* permintaan tersebut dari layanan SOAP-nya ke *website* Perpustakaan A, demikian juga sebaliknya. Terdapat beberapa variabel yang harus ada dan poin yang distandarkan untuk memudahkan implementasi pertukaran data, yaitu:

- Poin pencarian katalog berupa variabel judul, pengarang, dan penerbit dengan sub pencarian *Dewey Decimal Classification* yang merupakan kategori pencarian berdasarkan klasifikasi ilmu.
- Data ISBN buku sebagai identitas unik dari suatu buku yang berfungsi untuk mengintegrasikan buku yang sama di antara perpustakaan.

3.1.2 Algoritma Pemrograman

Algoritma pemrograman WSDL katalog dapat dilihat pada Gambar 4.

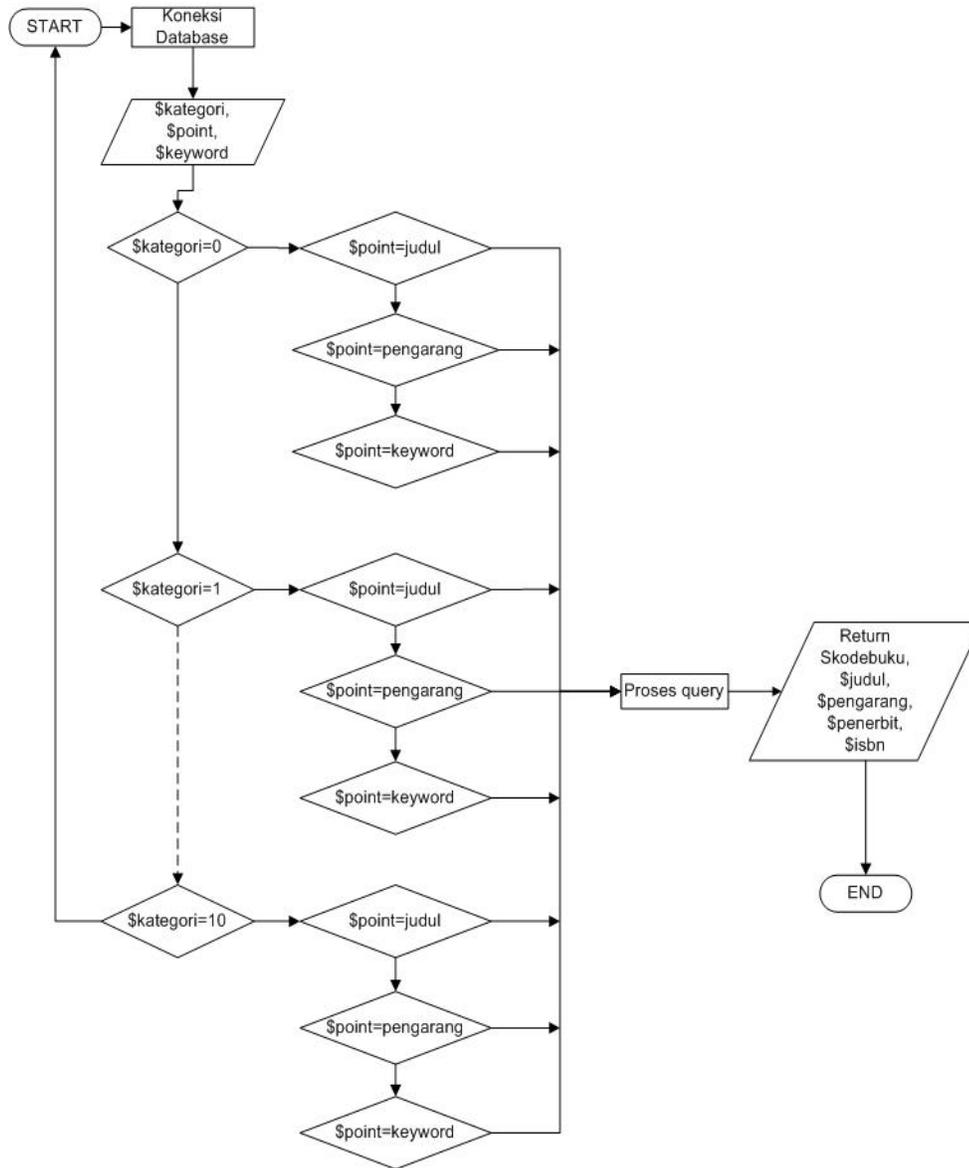


Gambar 4. Algoritma pemrograman WSDL katalog

Langkah-langkah pemrograman WSDL di *server* adalah:

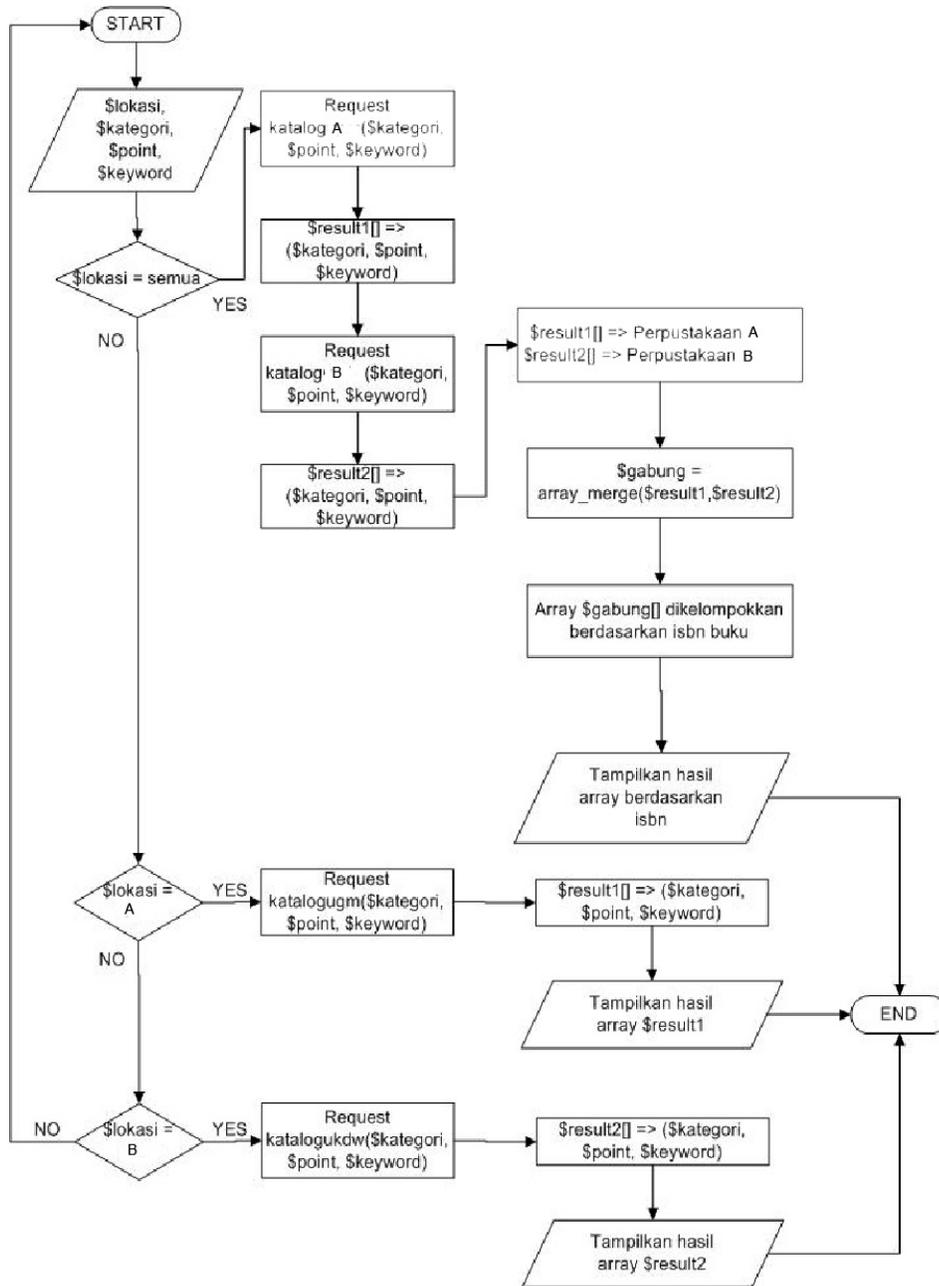
- Langkah pertama pemrograman WSDL di *server* adalah pemanggilan NuSOAP *code* yang merupakan *class* PHP untuk mengirim dan menerima pesan SOAP melalui HTTP dengan menggunakan `require_once('nusoap.php');`
- Selanjutnya membuat *server instance*: `$server = new soap_server();`
- Kemudian dilakukan inisialisasi WSDL *support* :
`$server -> configureWSDL('Aserver', 'urn:Aserver');`
- Langkah berikutnya adalah melakukan *register* struktur data yang digunakan oleh Service: `$server->wsdl->addComplexType`
`$server-> register('katalogA', array('param' => 'tns:TypeDataInput'), array('return'=>'xsd:string'))`
- Langkah terakhir adalah mendefinisikan fungsi pemanggilan katalog:
`function katalogA($param) dan menggunakan request untuk meminta layanan Service: $server->service($HTTP_RAW_POST_DATA);`

Gambar 5 merupakan algoritma dari fungsi pemanggilan katalog sedangkan Gambar 6 menjelaskan tentang algoritma pemrograman *client web service* katalog yang dirancang pada penelitian ini.



Gambar 5. Algoritma pemrograman fungsi pemanggilan katalog di *server*

Dalam fungsi pemanggilan katalog di *server*, langkah pertama adalah melakukan koneksi ke *database* katalog, selanjutnya menentukan parameter-parameter yang akan digunakan untuk pencarian katalog pengguna, ditentukan parameter *keyword*, *point* dan judul untuk melakukan *query* yang dikehendaki. Berdasarkan nilai parameter-parameter yang diterima dibuat *query* dimana selanjutnya hasil *query* tersebut mengembalikan variabel-variabel kode buku, judul, pengarang, penerbit, dan ISBN.



Gambar 6. Algoritma pemrograman *client web service* katalog

Penjelasan tentang algoritma pemrograman *client web service* katalog pada Gambar 6, dapat dijelaskan sebagai berikut:

- Aplikasi *client* mengirimkan variabel lokasi, *keyword*, *point*, dan kategori.
- Jika lokasi pencarian di semua perpustakaan, maka program akan mengirimkan *request* variabel *keyword*, *point*, dan kategori ke *web service* katalog A. Dari *web service* katalog A mendapat *response* berupa variabel *array* yang berisi data katalog sesuai dengan parameter yang diminta, kemudian dilanjutkan ke langkah berikut:

- Program mengirimkan *request* variabel *keyword*, *point*, dan kategori ke *web service* katalog B. Dari *web service* katalog B mendapat *response* berupa variabel *array* yang berisi data katalog sesuai dengan parameter yang diminta.
- Kedua *array* dari Perpustakaan A dan B kemudian digabung menjadi satu dan dikelompokkan berdasarkan ISBN untuk selanjutnya dimasukkan ke dalam *array* baru.
- *Array* baru yang terbentuk dari pengelompokan judul buku berdasarkan ISBN tersebut kemudian ditampilkan ke pengguna.
- Jika lokasi pencarian di perpustakaan A, maka program akan mengirimkan *request* variabel *keyword*, *point*, dan kategori ke *web service* katalog A, dari *web service* katalog A mendapat *response* berupa variabel *array* yang berisi data katalog sesuai dengan parameter yang diminta. Selanjutnya variabel *array* dari *web service* katalog A tersebut ditampilkan ke pengguna.
- Jika lokasi pencarian di perpustakaan B, maka program mengirimkan *request* variabel *keyword*, *point*, dan kategori ke *web service* katalog B, dari *web service* katalog B mendapat *response* berupa variabel *array* yang berisi data katalog sesuai dengan parameter yang diminta. Selanjutnya variabel *array* dari *web service* katalog B tersebut ditampilkan ke pengguna.

3.2 Pembahasan

Pembahasan pada subbab ini lebih mengarah pada pengujian hasil penelitian yang telah diuraikan pada sub bab sebelumnya. Hasil pengujian diperoleh dari pertukaran data katalog antarperpustakaan.

3.2.1 Pengujian *Black Box*

Untuk menguji kebenaran *input* dan *output* dibuat suatu form pencarian terintegrasi seperti Gambar 7 berikut.

Search
Katalog Perpustakaan

Semua Perpustakaan ▾		
Searching Point	Kategori	Keyword
Judul ▾	Semua Kategori ▾	

Gambar 7. Form pencarian katalog buku terintegrasi

Pengujian dilakukan dengan memasukkan *input* kata kunci *web service* dengan pilihan Lokasi di Semua Perpustakaan dan *point* pencarian berdasarkan Judul. Selanjutnya hasil eksekusi pencarian menghasilkan *output* berupa daftar buku yang berhubungan dengan Judul yang mengandung frase *web service* di Perpustakaan A dan B seperti Tabel 1 berikut ini.

Tabel 1. Hasil *query* di semua perpustakaan

Katalog A dan B				
No	Judul	Pengarang	Penerbit	Lokasi
1	Building web services with Java : making sense of XML, SOAP, WSOL,...	Graham, Steve, et al	SAMS	Perpustakaan A Perpustakaan B
2	Building web services with java : making sense ofXML, SOSP, WSDL, and	Graham, Steve, et al	SAMS	Perpustakaan A Perpustakaan B
3	Soap : cross platform web service development using XML	Seely, Scott	Prentice Hall PTR	Perpustakaan A Perpustakaan B
4	Real world XML web services : for VB and VB .Net developers.	Shohoud, Yasser	Addison-Wesley	Perpustakaan A Perpustakaan B
5	Visual basic, NET XML web services developer's guide.	Jennings, Roger	McGraw-Hill	Perpustakaan A Perpustakaan B
6	Real world XML web services : for VB and VB .Net developers.	Shohoud, Yasser	Addison-Wesley	Perpustakaan A Perpustakaan B
7	Visual basic. net XML Web services developer's guide.	Jennings, Roger	McGraw-Hill	Perpustakaan A

Hasil *query* di atas menunjukkan bahwa dari *input keyword* dan *point* pencarian yang dimasukkan menghasilkan *output* yang sesuai dengan pencarian berdasarkan *input* dan *point* pencariannya.

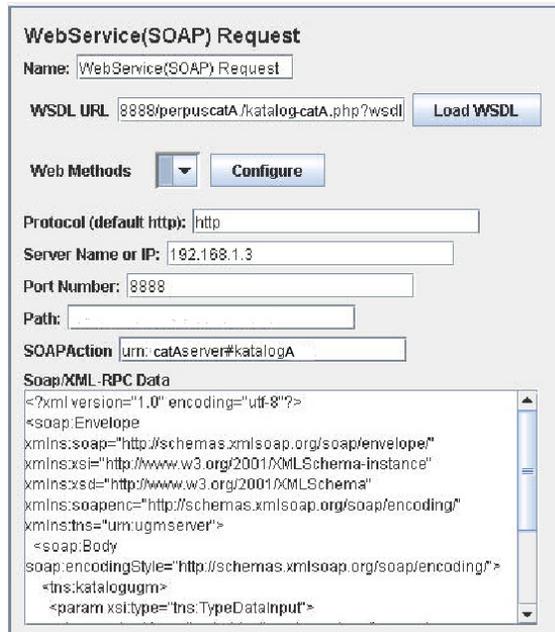
3.2.2 Pengujian *Gray Box*

Untuk pengujian *gray box* dilakukan uji fungsionalitas dan uji performansi yaitu sebagai berikut:

- **Test Fungsionalitas**

Untuk menguji fungsionalitas *web service* digunakan *tool* JMeter yang melakukan pengujian berdasarkan *file WSDL*. *Test Plan* dibuat dengan menambahkan *Thread Group and a Loop Controller* seperti pada Gambar 8. *Thread group* tersebut kemudian di-*set* jumlah *thread*-nya. Pada *test plan* ini digunakan *thread* 20, 50, dan 100 berturut-turut dengan *Ramp-Up Period* 3 detik dan *Loop count* 3 kali.

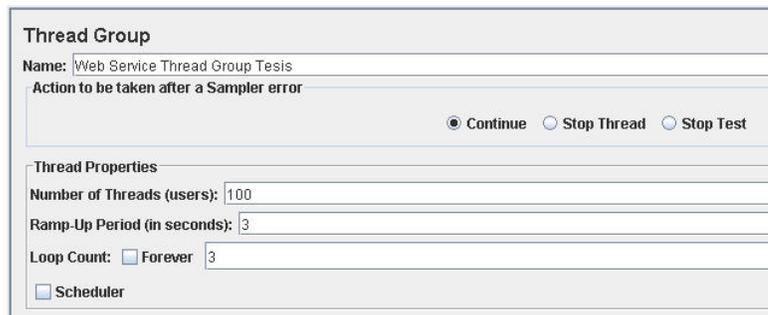
JMeter mendeteksi bahwa terdapat parameter-parameter *keyword*, *point*, dan kategori dari *web service*. Selanjutnya parameter tersebut diuji apakah berfungsi dengan baik, dengan cara mengirimkan *request* ke *server* berdasarkan parameter yang ada tersebut. Dari *request* yang diminta tersebut diperoleh *response* dari *server*. Gambar 8 membuktikan bahwa fungsionalitasnya berjalan dengan baik.



Gambar 8. *WebService(SOAP) request dialog* perpustakaan A

- **Test Performansi**

Hasil pengujian performansi *web service* dengan tool JMeter ditunjukkan dengan Gambar 9 dan Tabel 2 berikut ini.



Gambar 9. *Thread group dan a loop controller setting*

Tabel 2. Perbandingan hasil *request* dari *test* performansi perpustakaan A

<i>Number of threads</i>	<i>Throughput (responses/minute)</i>	<i>Average (ms)</i>	<i>Median (ms)</i>
20	155	4758	4697
50	272	7534	6860
100	458	12084	11006

Tabel 2 menunjukkan bahwa peningkatan jumlah *client* mempunyai *nonlinear impact* terhadap waktu *response server*. Dapat dikatakan bahwa meningkatkan jumlah 2 atau 3 kali tidak mempunyai pengaruh yang signifikan.

4. KESIMPULAN

Dari simulasi implementasi *web service* katalog antar perpustakaan dapat disimpulkan sebagai berikut:

- Teknologi *web service* dapat menjembatani pertukaran data antarperpustakaan yang berbeda sistem dan *Database*-nya.
- Simulasi program data katalog perpustakaan A dengan *database* MySQL dan data katalog perpustakaan B dengan *database* Visual Foxpro serta Konektor ODBC membuktikan bahwa katalog kedua perpustakaan ini dapat diintegrasikan.
- Hasil tes fungsionalitas menggunakan *tool* JMeter menunjukkan bahwa JMeter dapat mendeteksi adanya fungsi layanan dan parameter-parameter *keyword*, *point*, dan kategori yang terdapat dalam *web service* katalogA. *Request* dan *response* pada layanan parameter-parameter tersebut membuktikan bahwa fungsionalitasnya berfungsi dengan baik.
- Hasil tes performansi menggunakan *tool* Jmeter menunjukkan bahwa peningkatan jumlah *client* mempunyai *nonlinear impact* pada waktu *response server* sehingga dapat dikatakan bahwa meningkatkan jumlah *client* 2 atau 3 kali tidak mempunyai pengaruh yang signifikan terhadap *response server*.

REFERENSI

- [1]. Siswoutomo, W, “*Membangun Web Service Open Source menggunakan PHP*”, Elex Media Komputindo, Jakarta, 2004.
- [2]. Ayala, D., Browne, C., Chopra, V., dkk., “*Professional Open Source Web Services*”, Wiley Publishing, Inc., Indianapolis, 2000.
- [3]. “*Black, White & Gray Box SOA Testing Tools*”, http://www.crosschecknet.com/soa_testing_black_white_gray_box.php, diakses tanggal 3 Maret 2007.
- [4]. “*Essentials of Web Services Testing*”, <http://www.crosschecknet.com/> , diakses tanggal 4 Februari 2007.
- [5]. Nevedrov, Dmitri, “*Using JMeter to Performance Test Web Services*”, 2006.