

Kode/Rumpun Ilmu : 461/Sistem Informasi

**LAPORAN AKHIR  
PENELITIAN HIBAH BERSAING**



**IMPLEMENTASI ARSITEKTUR PERTUKARAN DATA BERBASIS  
DATA GRID DALAM MEMBANGUN GORONTALO LIBRARY  
NETWORK**

**TIM PENELITI**

Moh. Hidayat Keniyo, S.T., M.Kom	NIDN. 0016047303
Arip Mulyanto, M.Kom.	NIDN. 0023037603
Rochmad M.T. Jassin, S.Kom., M.Eng	NIDN. 0020078303

**UNIVERSITAS NEGERI GORONTALO  
NOVEMBER, 2013**

**Kode/Rumpun Ilmu : 461/Sistem Informasi**

**LAPORAN AKHIR  
PENELITIAN HIBAH BERSAING**



**IMPLEMENTASI ARSITEKTUR PERTUKARAN DATA BERBASIS  
DATA GRID DALAM MEMBANGUN GORONTALO LIBRARY  
NETWORK**

**TIM PENELITI**

<b>Moh. Hidayat Koniyo, S.T., M.Kom</b>	<b>NIDN. 0016047303</b>
<b>Arip Mulyanto, M.Kom.</b>	<b>NIDN. 0023037603</b>
<b>Rochmad M.T. Jassin, S.Kom.,M.Eng</b>	<b>NIDN. 0020078303</b>

**UNIVERSITAS NEGERI GORONTALO  
NOVEMBER, 2013**

HALAMAN PENGESAHAN

**Judul Kegiatan** : Implementasi Arsitektur Pertukaran Data Berbasis Data Grid Dalam Membangun Gorontalo Library Network

**Peneliti / Pelaksana**  
Nama Lengkap : MOHAMMAD HIDAYAT KONIYO, ST, M.Kom  
NIDN : 0016047303  
Jabatan Fungsional : Lektor Kepala  
Program Studi : Sistem Informasi  
Nomor HP :  
Alamat surel (e-mail) : hidayat@ung.ac.id

**Anggota Peneliti (1)**  
Nama Lengkap : ARIP MULYANTO, S.Kom, M.Kom  
NIDN : 0023037603  
Perguruan Tinggi : UNIVERSITAS NEGERI GORONTALO

**Anggota Peneliti (2)**  
Nama Lengkap : ROCHMAT MOHAMMAD THOHIR YASSIN, S.Kom, M.Eng  
NIDN : 0020078303  
Perguruan Tinggi : UNIVERSITAS NEGERI GORONTALO

**Institusi Mitra (jika ada)**  
Nama institusi Mitra :  
Alamat :  
Penanggung Jawab :  
Tahun Pelaksanaan :  
Biaya Tahun Berjalan : Rp. 50.000.000,00  
Biaya Keseluruhan : Rp. 50.000.000,00



Mengetahui,  
Dekan Fakultas Teknik

Dr. Kawiyah Husnan, MT  
NIP/NIK 196404271994032901

Gorontalo, 11 - 11 - 2013,  
Ketua,

(MOHAMMAD HIDAYAT KONIYO, ST, M.Kom)  
NIP/NIK 197304162001121001



Menyetujui,  
Ketua Lembaga Penelitian

Dr. Fitriyane Lihawa, M.Si  
NIP/NIK 196912091993032001

## RINGKASAN

Meningkatnya ukuran koleksi perpustakaan dan meningkatnya kebutuhan akses ke koleksi perpustakaan daerah Gorontalo, menjadikan perpustakaan daerah belum sepenuhnya mampu memenuhi kebutuhan masyarakat. Saat ini, masyarakat kesulitan untuk mengakses perpustakaan karena lokasinya berjauhan dengan perpustakaan. Masyarakat membutuhkan sistem yang mampu menyediakan akses ke banyak koleksi dari berbagai perpustakaan. Tujuan akhir penelitian ini adalah terwujudnya Gorontalo Library Network (GLN) yang memudahkan mekanisme pertukaran data antar perpustakaan, sehingga pengguna dapat mengakses berbagai koleksi dari beberapa perpustakaan. Untuk mencapai tujuan tersebut, penelitian ini memiliki target khusus yaitu: menganalisis kebutuhan sistem, merancang arsitektur sistem, membuat prototipe sistem, membuat aplikasi GLN, implementasi sistem, serta sosialisasi dan publikasi GLN. Penelitian dilakukan dengan menggunakan metode *research and development*, yang diawali dengan kegiatan survey untuk mengidentifikasi kebutuhan sistem, kemudian merancang dan membangun sistem GLN. Penelitian menghasilkan sebuah jaringan perpustakaan Gorontalo yang terdiri dari 6 perpustakaan daerah di Provinsi Gorontalo dan 1 perpustakaan server yang berperan sebagai jembatan pertukaran data. Ketika pengguna mencari koleksi pada sebuah perpustakaan daerah, perpustakaan server akan menghubungkan permintaan pengguna ke perpustakaan daerah yang lain. Selanjutnya, hasil pencarian dari setiap perpustakaan daerah akan ditampilkan. Dengan demikian, akses ke koleksi perpustakaan dapat meningkat. Selain itu, sistem GLN dapat mengoptimalkan koleksi setiap perpustakaan. Artinya, koleksi setiap perpustakaan tidak perlu sama, karena dapat diakses dari perpustakaan lain.

**Kata kunci:** pertukaran data, perpustakaan digital, Gorontalo Library Network, peningkatan akses, optimalisasi koleksi

## DAFTAR ISI

<b>HALAMAN SAMPUL.....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>ii</b>
<b>RINGKASAN.....</b>	<b>iii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>v</b>
<b>BAB I PENDAHULUAN</b>	
<b>1.1. Latar Belakang.....</b>	<b>1</b>
<b>1.2. Urgensi Penelitian.....</b>	<b>2</b>
<b>BAB II TINJAUAN PUSTAKA</b>	
<b>2.1. Perpustakaan digital.....</b>	<b>4</b>
<b>2.2. Kerjasama antar perpustakaan.....</b>	<b>4</b>
<b>2.3. Arsitektur pertukaran data dalam perpustakaan digital.....</b>	<b>9</b>
<b>2.4. Peta Jalan (<i>Roadmap</i>) penelitian.....</b>	<b>12</b>
<b>BAB III TUJUAN DAN MANFAAT</b>	
<b>3.1. Tujuan Penelitian.....</b>	<b>14</b>
<b>3.2. Manfaat Penelitian.....</b>	<b>14</b>
<b>BAB IV METODE PENELITIAN</b>	
<b>4.1. Rancangan Penelitian.....</b>	<b>15</b>
<b>4.2. Tahapan Penelitian.....</b>	<b>15</b>
<b>BAB V HASIL DAN PEMBAHASAN</b>	
<b>5.1. Hasil.....</b>	<b>18</b>
<b>5.2. Pembahasan.....</b>	<b>43</b>
<b>BAB VI KESIMPULAN DAN SARAN</b>	
<b>6.1. Kesimpulan.....</b>	<b>44</b>
<b>6.2. Saran.....</b>	<b>44</b>
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

<b>Gambar 1. Framework GDLG.....</b>	<b>5</b>
<b>Gambar 2. Arsitektur GDLG.....</b>	<b>6</b>
<b>Gambar 3. Arsitektur Perpustakaan Digital.....</b>	<b>7</b>
<b>Gambar 4. Arsitektur DLinGrid.....</b>	<b>8</b>
<b>Gambar 5. Arsitektur Terpusat.....</b>	<b>9</b>
<b>Gambar 6. Arsitektur Terdistribusi.....</b>	<b>10</b>
<b>Gambar 7. Arsitektur Terdistribusi Bertingkat.....</b>	<b>11</b>
<b>Gambar 8. Roadmap Penelitian.....</b>	<b>13</b>
<b>Gambar 9. Tahapan Penelitian.....</b>	<b>17</b>
<b>Gambar 10. Desain Teknis Arsitektur GLN.....</b>	<b>20</b>
<b>Gambar 11. Skenario Proses Kerja Arsitektur GLN.....</b>	<b>21</b>
<b>Gambar 12. Digilib ke SLIMS.....</b>	<b>34</b>
<b>Gambar 13. SLIMS ke Server Digilib.....</b>	<b>34</b>
<b>Gambar 14. Advanced Search.....</b>	<b>35</b>
<b>Gambar 15. Hasil dari SLIMS local.....</b>	<b>35</b>
<b>Gambar 16. Server Digilib.....</b>	<b>36</b>
<b>Gambar 17. Pengujian fungsi getData.....</b>	<b>36</b>
<b>Gambar 18. Hasil pengujian getData()-1.....</b>	<b>37</b>
<b>Gambar 19. Hasil pengujian getData()-2.....</b>	<b>37</b>
<b>Gambar 20. Hasil pengujian getData()-3.....</b>	<b>38</b>
<b>Gambar 21. Pengujian server SLIMS.....</b>	<b>38</b>
<b>Gambar 22. Link get_data.....</b>	<b>39</b>
<b>Gambar 23. Link WSDL.....</b>	<b>39</b>
<b>Gambar 24. Component Diagram.....</b>	<b>40</b>
<b>Gambar 25. Membuat domain baru.....</b>	<b>41</b>
<b>Gambar 26. Mengaktifkan secure admin.....</b>	<b>41</b>
<b>Gambar 27. Halaman login glassfish server.....</b>	<b>42</b>
<b>Gambar 28. Halaman utama glassfish server.....</b>	<b>42</b>
<b>Gambar 29. Deploy applications.....</b>	<b>42</b>
<b>Gambar 30. Halaman WSDL.....</b>	<b>43</b>



# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Sebagai salah satu sumber ilmu dan pembelajaran, perpustakaan terus mengalami perkembangan dalam upaya menyediakan informasi bagi para pengguna. Saat ini, perpustakaan bukan hanya menjadi perhatian lembaga pendidikan saja, akan tetapi sudah menjadi perhatian pemerintah baik di tingkat pusat sampai daerah. Hal ini dibuktikan dengan banyaknya perpustakaan daerah yang menyediakan berbagai sumber bacaan bagi masyarakat. Dengan demikian, masyarakat memiliki banyak alternatif dalam memperoleh informasi yang dibutuhkan melalui perpustakaan.

Walaupun demikian, meningkatnya kuantitas perpustakaan belum sepenuhnya mampu memenuhi kebutuhan pengguna. Saat ini, ketika membutuhkan sumber bacaan, pengguna mencarinya di sebuah perpustakaan. Ketika sumber bacaan yang dicari tidak tersedia di perpustakaan tersebut, pengguna harus mencari perpustakaan lain. Demikian seterusnya. Hal ini akan mudah kalau jarak antar perpustakaan berdekatan. Kalau lokasinya berjauhan, maka akan menjadi kendala dalam hal waktu dan tenaga. Pengguna membutuhkan sistem perpustakaan yang mampu menyediakan akses ke banyak koleksi dari berbagai perpustakaan. Untuk itu, perlu ada pertukaran data antar perpustakaan. Untuk melakukan pertukaran data, koleksi perpustakaan harus berbentuk digital. Saat ini, sudah banyak perpustakaan yang menyediakan perpustakaan digital selain tetap menyediakan perpustakaan konvensional. Perpustakaan digital memungkinkan pertukaran data antar perpustakaan.

Namun, pertukaran data antar perpustakaan digital bukan pekerjaan mudah karena biasanya setiap perpustakaan digital mengembangkan sistemnya masing-masing. Akibatnya, sulit melakukan pertukaran data antar perpustakaan digital. Hal ini disebabkan karena perbedaan platform perangkat keras dan perangkat lunak yang digunakan, arsitektur dan media penyimpanan yang berbeda-beda, kultur kepemilikan data serta masalah keamanan data.

Penelitian ini akan menghasilkan jaringan perpustakaan yang memudahkan mekanisme pertukaran data antar perpustakaan digital. Jaringan yang bernama “Gorontalo Library Networking” (GLN) ini terdiri dari 7 perpustakaan digital yang ada di Provinsi Gorontalo, yakni 1 perpustakaan daerah provinsi dan 6 perpustakaan daerah Kabupaten/Kota. Ketika pengguna mengakses Gorontalo Library Networking, pengguna dapat mengakses koleksi dari

7 perpustakaan digital tersebut. Dengan demikian, akses pengguna terhadap koleksi perpustakaan menjadi meningkat dan lebih mudah. Kemudahan akses sangat terasa karena jarak antar perpustakaan daerah di Gorontalo berjauhan. Selain itu, GLN dapat mengoptimalkan koleksi setiap perpustakaan digital yang bekerja sama. Artinya koleksi setiap perpustakaan tidak perlu sama, karena tersedia di perpustakaan lain yang dapat diakses pengguna.

## **1.2 Urgensi Penelitian**

Saat ini, perpustakaan digital banyak dikembangkan oleh perpustakaan-perpustakaan di Indonesia. Dengan perpustakaan digital, seseorang dapat memperoleh informasi yang dibutuhkan tanpa harus datang langsung di perpustakaan sebagai pusat informasi. Secara umum perpustakaan digital merupakan koleksi informasi terstruktur dalam format digital yang telah diatur sedemikian rupa sehingga dapat diakses user dengan mudah melalui jaringan.

Meskipun perpustakaan digital sedang menjadi trend, namun masih terdapat beberapa permasalahan dalam pengembangannya. Salah satu masalah terbesar dari perpustakaan digital adalah belum adanya mekanisme berbagi sumber daya yang sistematis antar perpustakaan digital. Hal ini disebabkan beragamnya sistem yang dikembangkan masing-masing perpustakaan digital.

Untuk mengatasi masalah ini, beberapa penelitian sebelumnya mengusulkan penggunaan teknologi grid yang memiliki kemampuan mengintegrasikan platform yang heterogen. Teknologi grid menjanjikan sebuah arsitektur perpustakaan digital yang fleksibel dan lebih terbuka (Sebestyen-Pal, Banciu, Balint, Moscaiu, & Sebestyen-Pal, 2008). Dengan bantuan teknologi grid, permasalahan seperti bertambahnya konten yang heterogen, layanan yang kompleks dan ukuran metadata yang besar dapat diatasi. Grid merupakan sistem yang menjanjikan dukungan terhadap berbagi sumber daya, sumber daya yang besar dan sumber daya tersebar (Foster & Kesselman, 2004).

Penelitian Mulyanto (2010), menghasilkan sebuah arsitektur pertukaran data (DLINGRID) yang memberikan kemudahan mekanisme berbagi sumber daya antar perpustakaan digital. Dengan arsitektur ini, sebuah perpustakaan digital dapat melakukan pertukaran data dengan mudah, sehingga satu perpustakaan dapat menampilkan lebih banyak koleksi. Selain koleksi yang dimilikinya, perpustakaan ini dapat menampilkan koleksi dari perpustakaan lain yang bekerja sama. Hal tersebut diperkuat oleh Mulyanto dan Jassin (2012), yang telah berhasil



menerapkan arsitektur DLINGRID dalam kerjasama antar dua buah perpustakaan digital di Universitas Negeri Gorontalo.

Berdasarkan penjelasan di atas, perlu dilakukan penelitian untuk mengatasi permasalahan sulitnya masyarakat dalam mengakses perpustakaan daerah seperti di Gorontalo. Letak geografis Gorontalo yang berbentuk pegunungan dan lautan, menyulitkan masyarakat yang jauh dari lokasi perpustakaan dalam mengakses koleksi perpustakaan. Masyarakat membutuhkan waktu lama dan biaya yang tidak sedikit untuk mengakses koleksi perpustakaan yang masih berbentuk konvensional.

Penelitian ini bertujuan membangun perpustakaan digital berbasis data grid untuk meningkatkan akses ke koleksi perpustakaan digital dan mengoptimalkan koleksi setiap perpustakaan digital yang bekerja sama. Perpustakaan ini, mengintegrasikan berbagai perpustakaan digital menggunakan *datagrid-middleware* iRODS. Dengan iRODS, proses berbagi sumber daya menjadi lebih mudah, sehingga dapat meningkatkan akses ke koleksi perpustakaan digital. Selain itu, iRODS memungkinkan proses replikasi antar *server* sehingga dapat mengoptimalkan koleksi di masing-masing perpustakaan digital.

## **BAB 2 TINJAUAN PUSTAKA**

### **2.1 Perpustakaan Digital**

Terdapat banyak definisi mengenai perpustakaan digital. Para ahli mendefinisikan perpustakaan digital sesuai dengan kebutuhan mereka masing-masing. Menurut (Arms, 2000), perpustakaan digital merupakan kumpulan informasi beserta layanan-layanannya yang disimpan dalam format digital dan dapat diakses melalui jaringan komputer. Pendapat ini diperkuat oleh (Greenstein & Thorin, 2002), yang mengatakan bahwa perpustakaan digital adalah sebuah perpustakaan yang informasinya disimpan dalam format digital dan dapat diakses melalui komputer. Format digital disimpan secara lokal dalam sebuah komputer, tetapi dapat diakses oleh pengguna melalui jaringan komputer.

Pendapat lain dikemukakan oleh (Larson & Sanderson, 2005), yang mendefinisikan perpustakaan digital sebagai sebuah perpustakaan virtual global yang terdiri dari beberapa jaringan perpustakaan elektronik. Sedangkan (Candela, et al., 2007), mendefinisikan perpustakaan digital sebagai sebuah organisasi (bisa virtual), yang secara komprehensif mengumpulkan, mengelola dan menjaga berbagai konten objek digital untuk jangka waktu yang panjang serta menyediakan fungsi-fungsi dan layanan khusus bagi pengguna.

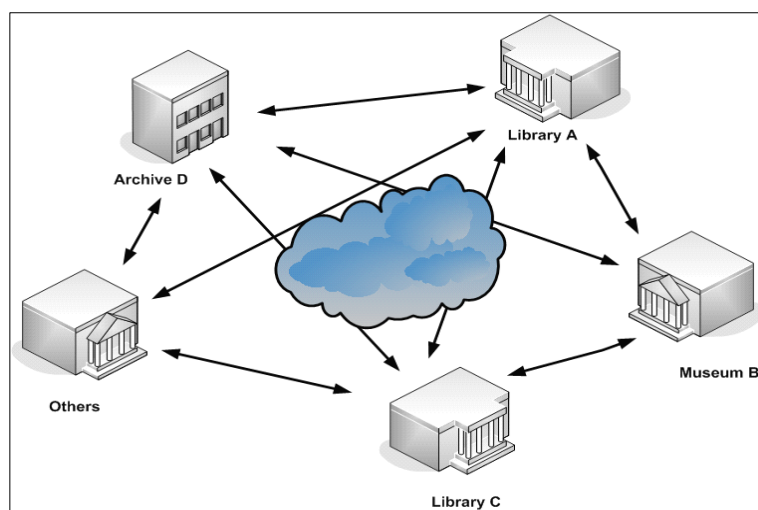
Sementara itu, (Drabenstott, 1994) berpendapat bahwa perpustakaan digital bukan merupakan sebuah entitas tunggal, melainkan terdiri dari beberapa entitas yang masing-masing memiliki sumber daya. Perpustakaan digital membutuhkan teknologi untuk menghubungkan ke berbagai sumber daya. Hubungan di antara perpustakaan digital dan layanan informasi bersifat transparan terhadap pengguna. Perpustakaan digital harus dapat memberikan layanan informasi dan dapat diakses secara universal.

### **2.2 Kerja Sama antar Perpustakaan**

Perpustakaan digital biasanya dikembangkan menggunakan teknologi, platform, protokol dan arsitektur yang berbeda-beda, akibatnya akan timbul masalah ketika beberapa perpustakaan digital melakukan kerja sama untuk berbagi sumber daya. Untuk mengatasi hal tersebut, beberapa penelitian mengusulkan penggunaan teknologi grid yang memiliki kemampuan mengintegrasikan platform yang heterogen.

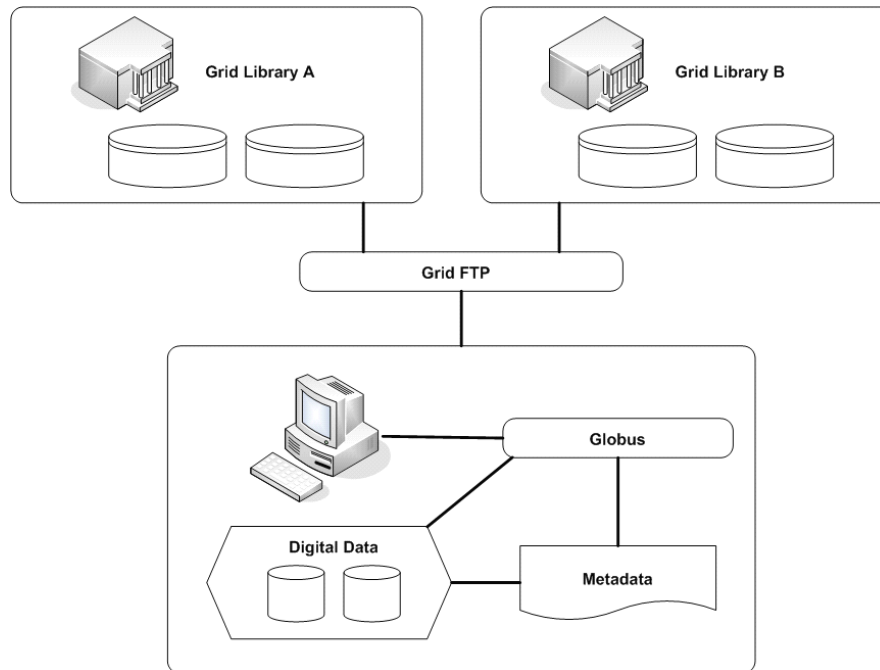
a. GDLG (Global Digital Library Grid)

(Yang & Ho, 2005), mengusulkan konsep Global Digital Library Grid (GDLG) seperti terlihat pada gambar 1. Teknologi data grid digunakan untuk menghubungkan institusi yang memiliki koleksi digital seperti perpustakaan, museum dan arsip ke dalam bentuk sebuah museum virtual besar yang disebut GDLG. Dalam mekanisme manajemen file, GDLG membolehkan pengguna untuk melakukan manajemen katalog, manajemen pengiriman replika dan akses ke file secara efisien. Dengan *replica selection*, *site* paling efisien akan ditempatkan secara otomatis dan salinannya akan terlihat di *site* lokal pengguna.



**Gambar 1. Framework GDLG** (Yang & Ho, 2005)

GDLG dibuat untuk mengatasi masalah pada perpustakaan digital. Masalah terbesar dari perpustakaan digital adalah belum adanya mekanisme berbagi sumber daya yang sistematis untuk menghubungkan berbagai perpustakaan digital. Masing-masing perpustakaan digital harus mengembangkan sistemnya sendiri. Untuk mengatasi masalah tersebut, dibuat arsitektur GDLG seperti terlihat dalam gambar 2., dengan langkah-langkah sebagai berikut:



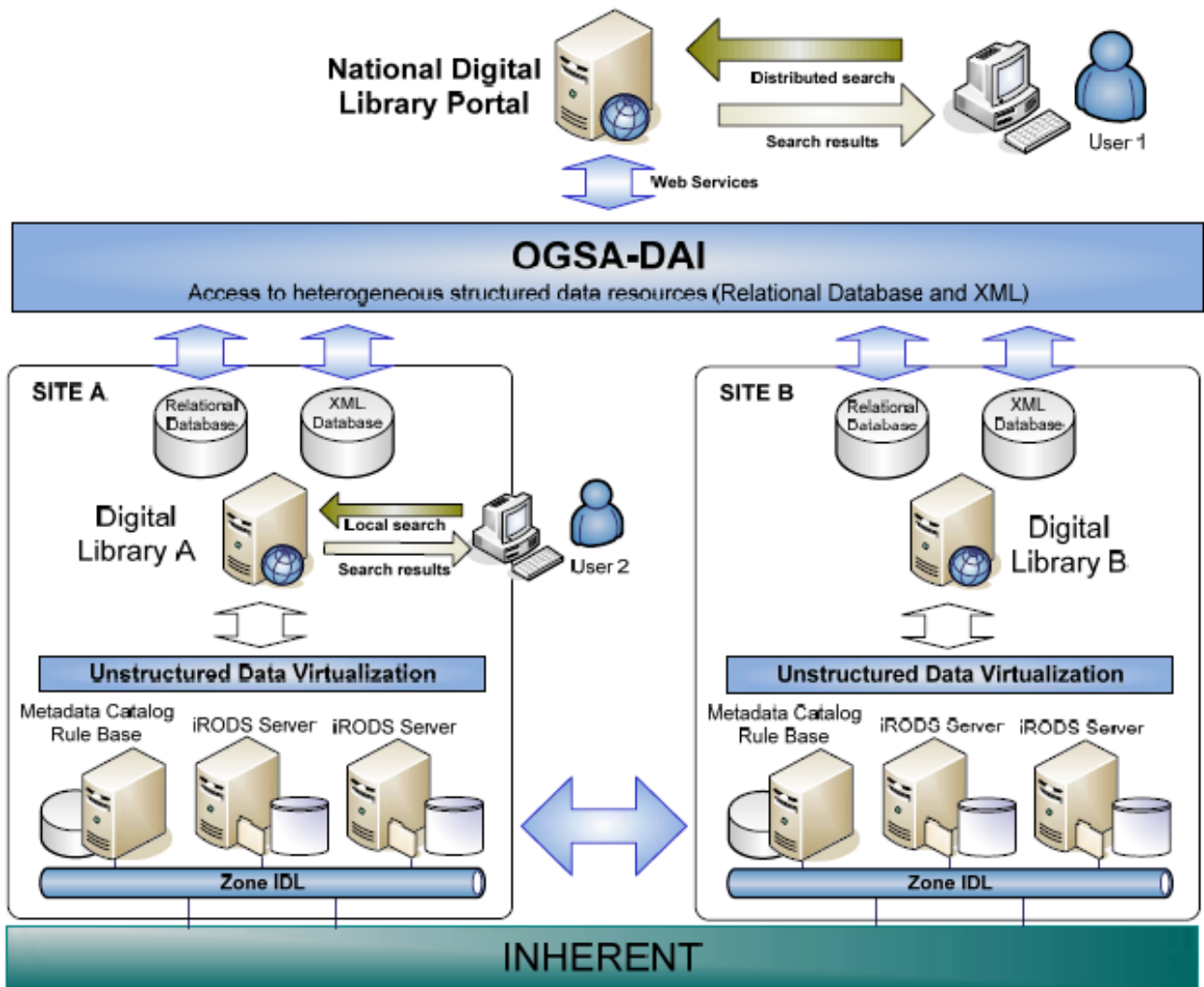
**Gambar 2. Arsitektur GDLG** (Yang & Ho, 2005)

- Step 1 : Masing-masing anggota grid (perpustakaan atau museum) harus menseting lingkungan dasar grid, seperti install Globus Toolkit dan software data grid sebelum terhubung dengan member lain.
- Step 2 : Masing-masing anggota Grid harus mendigitalkan data yang akan di *share*. Data dirubah ke bentuk format elektronik seperti file PDF atau ke bentuk multimedia seperti JPEG atau file *streaming*.
- Step 3 : Data digital disimpan dalam masing-masing sistem penyimpanan dan file metadata. Deskripsi data disimpan dalam repositori metadata, sebelum mekanisme manajemen replika dimulai.
- Step 4 : Menghubungkan Grid dengan GridFTP. GridFTP digunakan sebagai protokol untuk transfer data. GridFTP merupakan protokol standar dalam Globus Toolkit untuk transfer data ke dalam grid.
- Step 5 : Mengintegrasikan Grid Portal dan Organisasi Grid dengan Globus Toolkit.

b. Berbagi Sumber Daya Pendidikan di Indonesia

(Kurniawan & Hasibuan, 2009) mengusulkan sebuah arsitektur untuk berbagi sumber daya pembelajaran dalam lingkungan yang heterogen dengan menggunakan teknologi data grid. Dalam arsitektur ini, data sebagai salah satu sumber daya yang sering dibagi di antara berbagai perpustakaan digital, dibedakan menjadi dua kelompok, yaitu data terstruktur dan

tidak terstruktur. Selain itu, arsitektur ini menggunakan dua data grid *middleware*, yaitu iRODS dan OGSA-DAI.



**Gambar 3. Arsitektur Perpustakaan Digital**

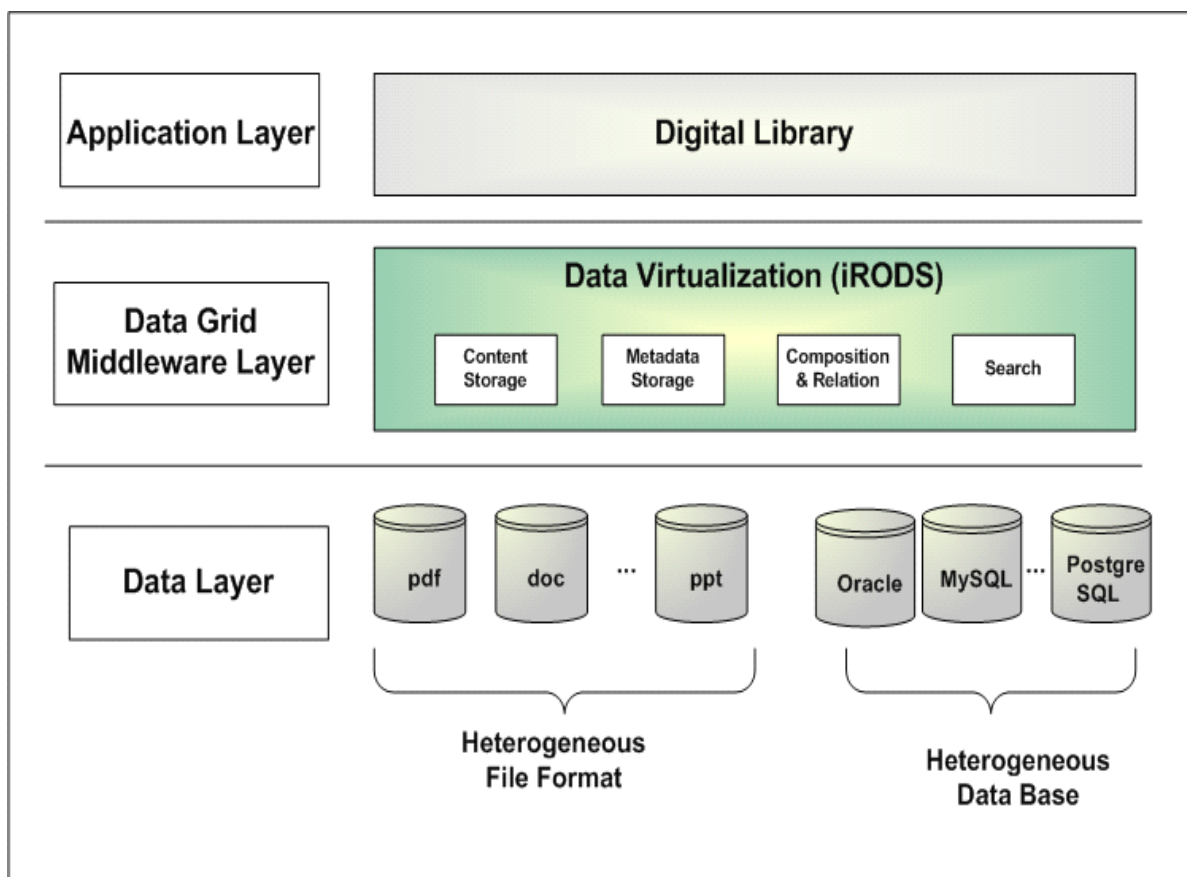
(Kurniawan & Hasibuan, 2009)

Gambar 3 memperlihatkan sebuah implementasi dari arsitektur yang diusulkan untuk perpustakaan digital. Setiap perpustakaan digital diregistrasi dalam sebuah portal *National Digital Library* yang mengelola semua sumber daya yang dimiliki oleh koleksi objek digital dan data terstruktur (data relasional dan XML). Objek digital disimpan dalam berbagai sistem penyimpanan yang dikelola oleh server iRODS. Setiap server iRODS diregistrasi dalam 1 (satu) zone yaitu zone *Indonesia Digital Library* (IDL), sehingga objek digital dapat

direplikasi di antara server-server tersebut. File dalam site A dapat direplikasi ke site B, demikian juga sebaliknya. Server katalog metadata dalam dua site tersebut akan memiliki berbagai informasi yang sama.

c. Arsitektur DLinGrid

Untuk mengatasi beberapa permasalahan dalam perpustakaan digital, Mulyanto (2010) membuat sebuah desain arsitektur perpustakaan digital berbasis data grid yang diberi nama DLinGrid (Digital Library in Data Grid), seperti terlihat dalam gambar 4.



**Gambar 4. Arsitektur DLinGrid**

Gambar 4 menunjukkan arsitektur berbagi sumber daya dalam perpustakaan digital usulan. Arsitektur ini terdiri dari tiga lapisan, yaitu: lapisan data, lapisan *data grid middleware* dan lapisan aplikasi. Lapisan data berisi berbagai jenis sumber daya data dengan format yang berbeda-beda seperti pdf, doc dan ppt. Lapisan ini juga berisi beragam basis data seperti Oracle, MySQL dan PostgreSQL. Sedangkan lapisan *data grid middleware*, berisi

metadata dari berbagai struktur data, yang menyediakan layanan dasar sebagai berikut:

- a. *Content storage service*, untuk menyimpan berbagai tipe konten objek digital dan untuk direplikasi ke berbagai sumber daya yang lain. Layanan ini tidak tergantung pada tipe konten karena hanya klien yang membutuhkan yang akan.
- b. *Metadata storage service*, untuk menyimpan metadata yang menggambarkan objek digital. Banyak metadata yang dapat dihubungkan ke satu objek tertentu. Metadata juga mencatat informasi yang berhubungan dengan replikasi.
- c. *Composition and relation service*, untuk menjelaskan berbagai relasi antar objek digital dan untuk menjelaskan berbagai kelompok objek relasi. Relasi ini dapat digunakan untuk membuat objek digital yang kompleks dan membuat koleksi objek digital.
- d. *Search service*, untuk mencari objek digital. Pencarian didasarkan pada kesesuaian kueri dengan katalog metadata.

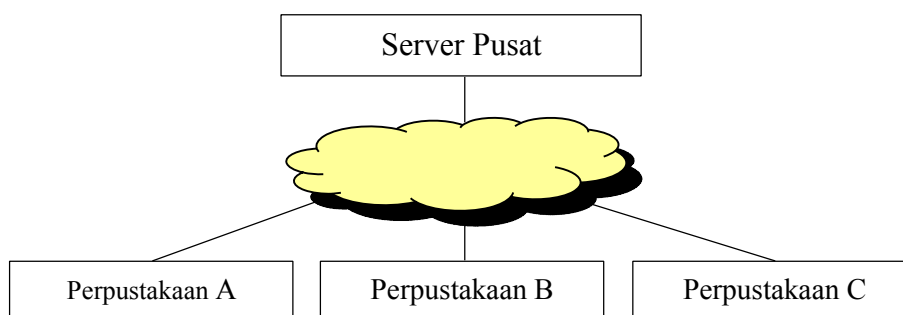
Lapisan terakhir dari arsitektur ini yaitu lapisan aplikasi, menampilkan dan membagi konten objek digital dengan menggunakan data virtualisasi.

### 2.3 Arsitektur Pertukaran Data dalam Perpustakaan Digital

Untuk mendukung kerja sama antar perpustakaan digital, diperlukan suatu arsitektur yang memungkinkan proses pertukaran data. Terdapat beberapa arsitektur pertukaran data yang dapat digunakan dalam kerja sama antar perpustakaan digital di Indonesia (Aji dan Wibowo, 2007), yaitu:

#### a. **Arsitektur Terpusat**

Pada arsitektur terpusat, semua data dari masing-masing perpustakaan dibaca oleh sebuah *server* pusat. *Server* pusat ini membaca data perpustakaan dengan menggunakan protokol-protokol pertukaran yang ada. Pengguna dapat menggunakan *server* pusat ini untuk mencari informasi di perpustakaan yang terhubung ke dalam jaringan.



**Gambar 5. Arsitektur Terpusat (Aji dan Wibowo, 2007)**



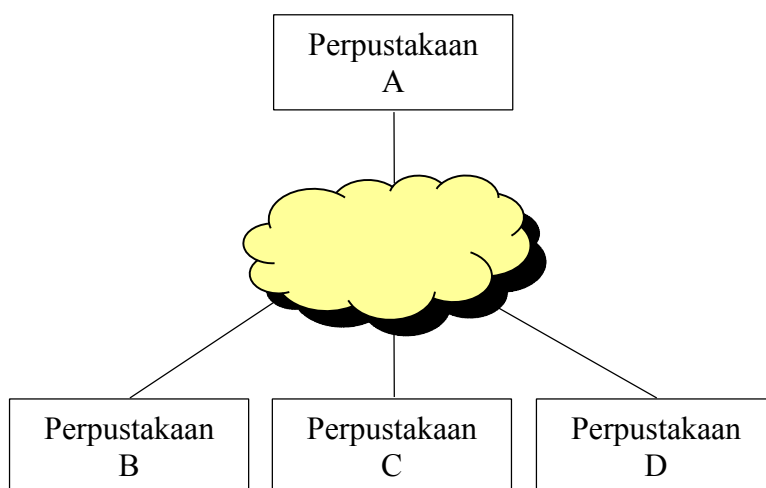
Arsitektur terpusat dapat mengatasi masalah perbedaan platform dan perbedaan format penyimpanan data, dengan menggunakan protokol dan metadata yang seragam untuk ditransfer ke *server* pusat. Namun, arsitektur ini memiliki beberapa kelemahan, yaitu:

- jika *server* pusat tidak bisa diakses, maka semua data tidak akan bisa diakses.
- perpustakaan dengan kultur kepemilikan data yang kuat, sulit bergabung karena data yang dimilikinya harus ditransfer ke *server* pusat.
- rawan terjadi *bottleneck* dan *overload* kerja pada *server* pusat.

Arsitektur terpusat dapat diaplikasikan dengan menggunakan protokol OAI-PMH, Z39.50 dan *web service*. Sedangkan untuk metadata, dapat menggunakan Dublin Core dan MARC sesuai dengan kompatibilitas protokol yang digunakan.

### b. Arsitektur Terdistribusi

Dalam arsitektur ini, setiap perpustakaan yang terhubung dalam satu jaringan kerja sama saling berkomunikasi satu sama lain. Dengan menggunakan arsitektur terdistribusi, beban pencarian kerja tidak dipusatkan pada satu perpustakaan dan dapat meminimalkan *bottleneck*. Kelemahan arsitektur ini adalah, jika salah satu perpustakaan terputus dari jaringan, maka data perpustakaan tersebut tidak dapat diakses oleh perpustakaan lain.



**Gambar 6. Arsitektur Terdistribusi (Aji dan Wibowo, 2007)**

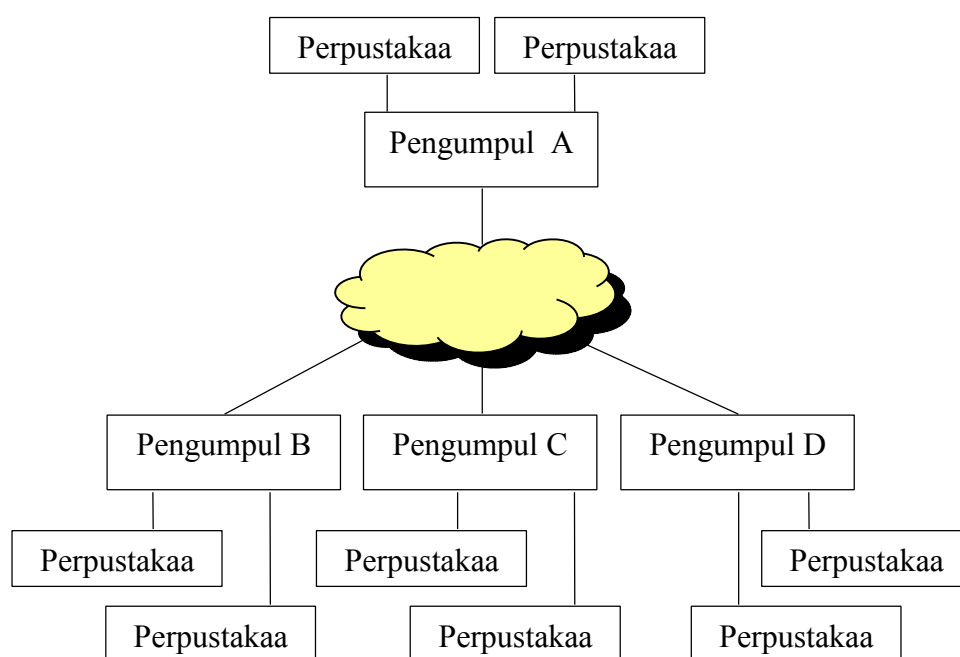
Arsitektur ini dapat mengatasi masalah perbedaan platform dan media penyimpanan serta mengatasi masalah kepemilikan data dan keamanan data perpustakaan. Setiap perpustakaan berkuasa atas data yang dimilikinya, dan berhak menentukan sendiri data apa saja yang akan

diberikan ke pihak lain. Dalam arsitektur ini, setiap perpustakaan menyediakan layanan yang dapat diakses oleh pihak lain. Layanan ini menentukan apa saja yang bisa diakses dan siapa saja yang bisa mengakses data. Untuk mempermudah proses pertukaran data, diperlukan suatu kesepakatan bersama dalam penggunaan protokol komunikasi dan metadata yang digunakan.

Arsitektur ini menggunakan protokol interaktif seperti Z39.50, *web service* dan RPC (*Remote Procedure Call*). Protokol bertipe *harvest* seperti OAI-PMH, juga dapat digunakan untuk mengimplementasikan arsitektur ini. Metadata yang digunakan, dapat disesuaikan dengan kompatibilitas protokol.

### **c.Arsitektur Terdistribusi Bertingkat**

Arsitektur terdistribusi bertingkat merupakan gabungan dari arsitektur terpusat dan terdistribusi. Dalam arsitektur ini, beberapa perpustakaan tergabung dalam satu kelompok. Setiap kelompok memiliki koordinator yang bertugas mengumpulkan data perpustakaan yang ada dalam kelompok tersebut.



**Gambar 7. Arsitektur Terdistribusi Bertingkat (Aji dan Wibowo, 2007)**

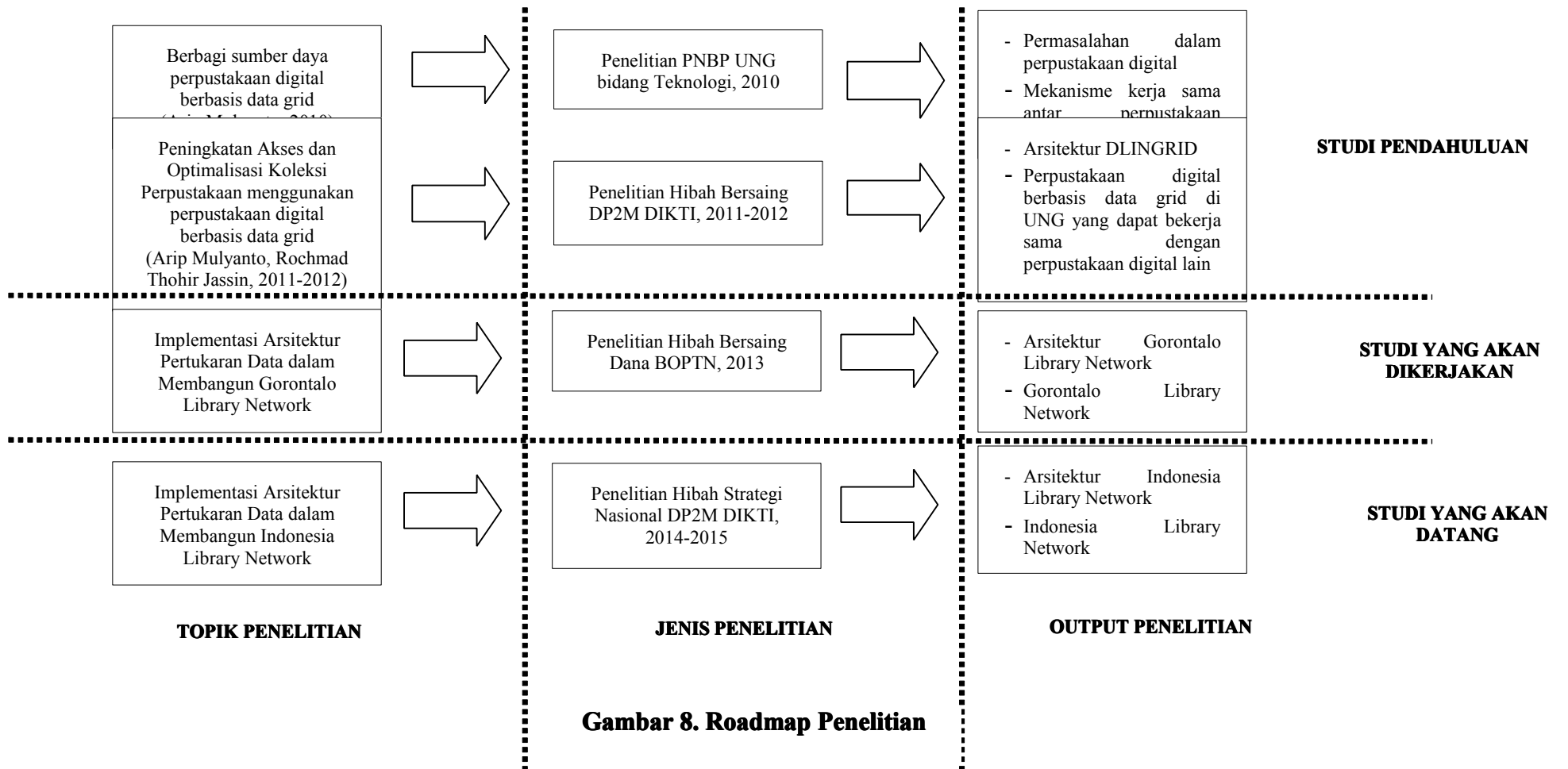
Implementasi arsitektur ini dapat menggunakan protokol yang biasa digunakan di dunia perpustakaan. Alternatif lain, pengumpul mengumpulkan data dari perpustakaan di dalam

kelompoknya dengan menggunakan protokol bertipe *harvest* seperti OAI-PMH, sedangkan komunikasi antar pengumpul data dapat menggunakan protokol tipe interaktif. Metadata yang digunakan sebaiknya disamakan untuk mempermudah proses pertukaran data.

Dari ketiga arsitektur di atas, arsitektur terdistribusi lebih cocok diterapkan di Indonesia. Hal ini disebabkan sesuai dengan kondisi perpustakaan di Indonesia yang tersebar dan memiliki kultur kepemilikan data yang kuat. Pada penelitian ini, arsitektur terdistribusi dijadikan acuan dalam proses pertukaran data antar perpustakaan digital.

#### **2.4 Peta Jalan (*Roadmap*) Penelitian**

Untuk mengatasi sulitnya mekanisme pertukaran data antar perpustakaan digital, Mulyanto (2010) telah menghasilkan sebuah arsitektur perpustakaan digital berbasis data grid (DLinGrid) yang memungkinkan beberapa perpustakaan digital yang heterogen untuk saling berbagi sumber daya. Sedangkan penelitian Mulyanto dan Jassin (2012) menghasilkan aplikasi perpustakaan digital yang memungkinkan untuk bekerja sama dengan perpustakaan digital yang lain. Dengan adanya kerja sama antara dua atau lebih perpustakaan digital, akses user ke koleksi perpustakaan akan meningkat, karena dengan satu kali permintaan user dapat memperoleh koleksi dari dua atau lebih perpustakaan, ketika koleksi yang dicari ditemukan di perpustakaan lain. Selain itu, koleksi setiap perpustakaan digital dapat dioptimalisasi. Dengan demikian, setiap perpustakaan yang bekerja sama tidak harus memiliki koleksi yang sama. Hal ini akan sangat membantu perpustakaan yang memiliki dana pengadaan koleksi terbatas.



**Gambar 8. Roadmap Penelitian**

## **BAB 3 TUJUAN DAN MANFAAT**

### **3.1 Tujuan Penelitian**

Tujuan umum penelitian ini adalah terwujudnya Gorontalo Library Network (GLN) yang memudahkan mekanisme pertukaran data antar perpustakaan, sehingga pengguna dapat mengakses berbagai koleksi dari beberapa perpustakaan.

Sedangkan tujuan khusus penelitian ini adalah:

- 1.** Mengidentifikasi permasalahan dan kebutuhan pertukaran data perpustakaan daerah pada seluruh Kabupaten/Kota di Provinsi Gorontalo
- 2.** Merancang arsitektur pertukaran data yang akan digunakan dalam GLN dengan platform yang berbeda-beda
- 3.** Membuat prototipe dan aplikasi sistem GLN yang menjembatani enam perpustakaan daerah yang dapat saling mempertukarkan data satu sama lain.

### **3.2 Manfaat Penelitian**

Manfaat dari penelitian ini adalah :

- 1.** Teridentifikasinya permasalahan dan kebutuhan pertukaran data perpustakaan daerah pada seluruh Kabupaten/Kota di Provinsi Gorontalo
- 2.** Diperolehnya arsitektur pertukaran data sistem GLN yang memungkinkan terjadinya pertukaran data antara enam perpustakaan daerah dengan platform yang berbeda-beda
- 3.** Memudahkan pengguna dalam mengakses data koleksi buku dari enam perpustakaan daerah menggunakan arsitektur pertukaran data dalam sistem GLN

## **BAB 4**

### **METODE PENELITIAN**

#### **4.1 Rancangan Penelitian**

Mengacu pada fokus dan luaran penelitian, maka penelitian ini menggunakan rancangan penelitian dan pengembangan (*research and development*). Dasar pemilihan metode tersebut adalah:

- a. pengembangan sistem GLN merupakan suatu kegiatan akademik yang memerlukan kajian teoritis dan tindakan nyata di lapangan, baik sebelum pengembangan maupun pada saat implementasi sistem
- b. dalam merancang sistem GLN ini, harus didasarkan pada serangkaian kegiatan baik di laboratorium maupun di lapangan (perpustakaan), sehingga rancangan penelitian dan pengembangan tepat untuk digunakan
- c. untuk merancang dan membuat sistem GLN, harus didahului dengan kegiatan survey, identifikasi dan analisis yang mendalam mengenai kebutuhan sistem GLN.

#### **4.2 Tahapan Penelitian**

Prosedur penelitian akan dikonsentrasikan dalam beberapa kegiatan yakni:

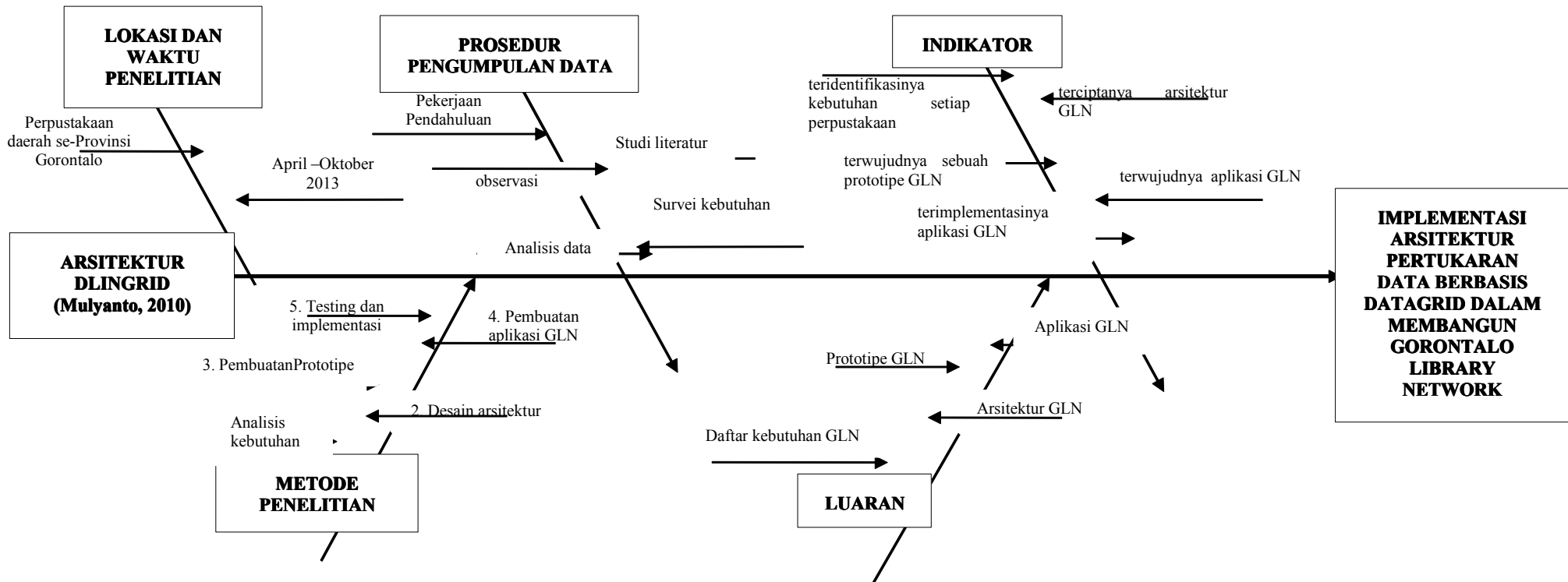
- a. Analisis kebutuhan yang diperlukan dalam pengembangan GLN. Pada tahap ini, akan disurvei mengenai keadaan perpustakaan saat ini dan kebutuhan untuk pengembangan GLN. Keluaran yang diharapkan dari tahapan ini adalah daftar kebutuhan dalam pengembangan perpustakaan digital. Target dari tahapan ini adalah tersurveinya perpustakaan di UNG, perguruan tinggi lain, sekolah dan perpustakaan daerah di provinsi Gorontalo. Sedangkan indikator keberhasilan pada tahap ini adalah teridentifikasi. Keluaran dari tahapan ini adalah arsitektur GLN yang memungkinkan perpustakaan di Gorontalo untuk saling bekerja sama. Target atau indikator keberhasilan pada tahap ini adalah terciptanya arsitektur GLN.
- b. Pembuatan prototipe, untuk membuat prototipe GLN yang akan diterapkan pada perpustakaan daerah Gorontalo. Keluaran dari tahapan ini adalah terwujudnya prototipe GLN, yang akan disosialisasikan kepada pihak-pihak terkait seperti pengelola perpustakaan baik perpustakaan universitas maupun perpustakaan fakultas, untuk mendapatkan masukan dari *user*. Target atau indikator keberhasilan pada tahap ini adalah

terwujudnya sebuah prototipe GLN.

- c. Pembuatan aplikasi, untuk membangun aplikasi GLN yang akan diterapkan di sejumlah perpustakaan di Gorontalo. Keluaran dari tahapan ini adalah terwujudnya aplikasi GLN, untuk diterapkan di perpustakaan universitas dan fakultas. Target atau indikator keberhasilan pada tahap ini adalah terwujudnya sebuah aplikasi perpustakaan digital berbasis data grid.
- d. Implementasi, untuk mengimplementasikan aplikasi GLN di sejumlah perpustakaan di Gorontalo. Keluaran dari tahapan ini adalah terimplementasinya GLN pada beberapa perpustakaan di Gorontalo. Target atau indikator keberhasilan pada tahap ini adalah terimplementasinya GLN yang memungkinkan beberapa perpustakaan di Gorontalo dapat bekerja sama.
- e. Sosialisasi dan publikasi, untuk mensosialisasikan aplikasi GLN kepada pihak-pihak terkait seperti pemerintah daerah di Gorontalo, pengelola perpustakaan, mahasiswa, dosen dan para peneliti di Gorontalo. Keluaran dari tahapan ini adalah sebuah kegiatan untuk mendesiminasikan dan mensosialisasikan aplikasi perpustakaan digital. Target atau indikator keberhasilan pada tahap ini adalah aplikasi GLN diketahui oleh pihak-pihak terkait.

Ringkasan tahapan penelitian digambarkan dalam bagan alur penelitian, seperti gambar 9.





**Gambar 9. Tahapan Penelitian**

## **BAB 5 HASIL DAN PEMBAHASAN**

### **5.1 Hasil**

#### **5.1.1 Analisis Kebutuhan**

Hasil survey pada beberapa perpustakaan daerah di Provinsi Gorontalo diperoleh beberapa data terkait dengan keadaan pada perpustakaan yang ada saat ini :

##### **1) Perpustakaan Daerah Kota Gorontalo**

Di wilayah kota Gorontalo perpustakaan masih bergabung dengan KPDE dimana letak perpustakaan berada di area taman kota Gorontalo dan dikelola oleh seorang petugas perpustakaan. Jumlah koleksi buku pada perpustakaan ini yaitu sebanyak 4000 eksemplar terbagi atas beberapa klasifikasi buku yaitu karya umum, filsafat dan psikologi, ilmu agama, ilmu sosial, bahasa, ilmu-ilmu murni (pasti/alam), ilmu terapan (teknologi), kesenian-hiburan-olahraga, kesusastraan, geografi dan sejarah. Untuk pengelolaan dan sirkulasi buku-buku tersebut, perpustakaan ini masih menggunakan aplikasi perkantoran yang terdapat pada perangkat *notebook* dan belum memiliki fasilitas komputer lainnya untuk mengakses katalog secara *online* maupun *offline*. Berdasarkan keadaan tersebut, diperlukan sebuah aplikasi yang memungkinkan untuk melakukan pencatatan, pengolaan anggota, penatan katalog buku hingga pada sirkulasi buku yang lebih baik.

##### **2) Perpustakaan Daerah Kabupaten Gorontalo**

Pada perpustakaan daerah ini terletak di pusat ibukota kabupaten yaitu di Limboto dimana menjadi satu kesatuan antara perpustakaan dan arsip daerah. Perpustakaan ini dikelola oleh kepala perpustakaan, sekretaris, administrator sistem hingga beberapa petugas sirkulasi. Terdapat beragam koleksi buku yang disertai juga dengan beberapa *e-book* serta terdapat 3 PC yang digunakan untuk sirkulasi buku dan untuk anggota perpustakaan. Pada perpustakaan ini juga sudah menggunakan aplikasi perpustakaan yang terotomatisasi menggunakan SLIMS (*Senayan Library Management System*) yang digunakan untuk pencatatan buku, pengelolaan anggota, penataan katalog hingga sirkulasi buku. Berdasarkan keadaan tersebut perpustakaan daerah kabupaten Gorontalo hanya tinggal menyesuaikan format pertukaran data dengan menggunakan arsitektur GLN yang akan diimplementasikan.

### **3) Perpustakaan Daerah Kabupaten Gorontalo Utara**

Hampir sama dengan beberapa perpustakaan daerah lainnya, di wilayah Kabupaten Gorontalo Utara kantor perpustakaan dengan arsip daerah dimana kantor ini terletak di daerah blok plan kantor pemerintahan daerah Gorontalo Utara. Perpustakaan ini dikepalai oleh seorang kepala perpustakaan dan arsip daerah dibantu oleh kepala bagian tata usah dan beberapa staff. Terdapat sekitar 6000 eksemplar buku pada perpustakaan ini dan dikelola menggunakan aplikasi perpustakaan (SLIMS) untuk penataan katalog, pengolaan anggota dan sirkulasi buku. Perpustakaan ini juga membantu pengolaan perpustakaan sekolah di wilayah Kabupaten Gorontalo Utara. Dari hasil survey yang diperoleh maka yang dibutuhkan adalah penyesuaian format pertukaran data dengan menggunakan arsitektur GLN yang akan diimplementasikan

### **4) Perpustakaan Daerah Kabupaten Boalemo**

Perpustakaan ini tergolong yang sudah cukup baik memberdayakan Teknologi Informasi dan Komunikasi (TIK) untuk pengelolaan perpustakaan. Memiliki hampir 9000 koleksi buku membuat perpustakaan ini menjadi salah satu perpustakaan daerah yang memiliki koleksi buku yang cukup besar. Untuk mengelola perpustakaan yang cukup besar ini, pengelola menggunakan dua buah aplikasi yaitu aplikasi dari Perpustakaan Nasional (Perpusnas) yaitu Inlis dan aplikasi SLIMS. Fungsi aplikasi Inlis dan SLIMS hampir sama namun untuk pelaporan ke Perpusnas aplikasi yang digunakan yaitu Inlis. Dari hasil survey yang diperoleh maka yang dibutuhkan adalah penyesuaian format pertukaran data dengan menggunakan arsitektur GLN yang akan diimplementasikan

### **5) Perpustakaan Daerah Kabupaten Pohuwato**

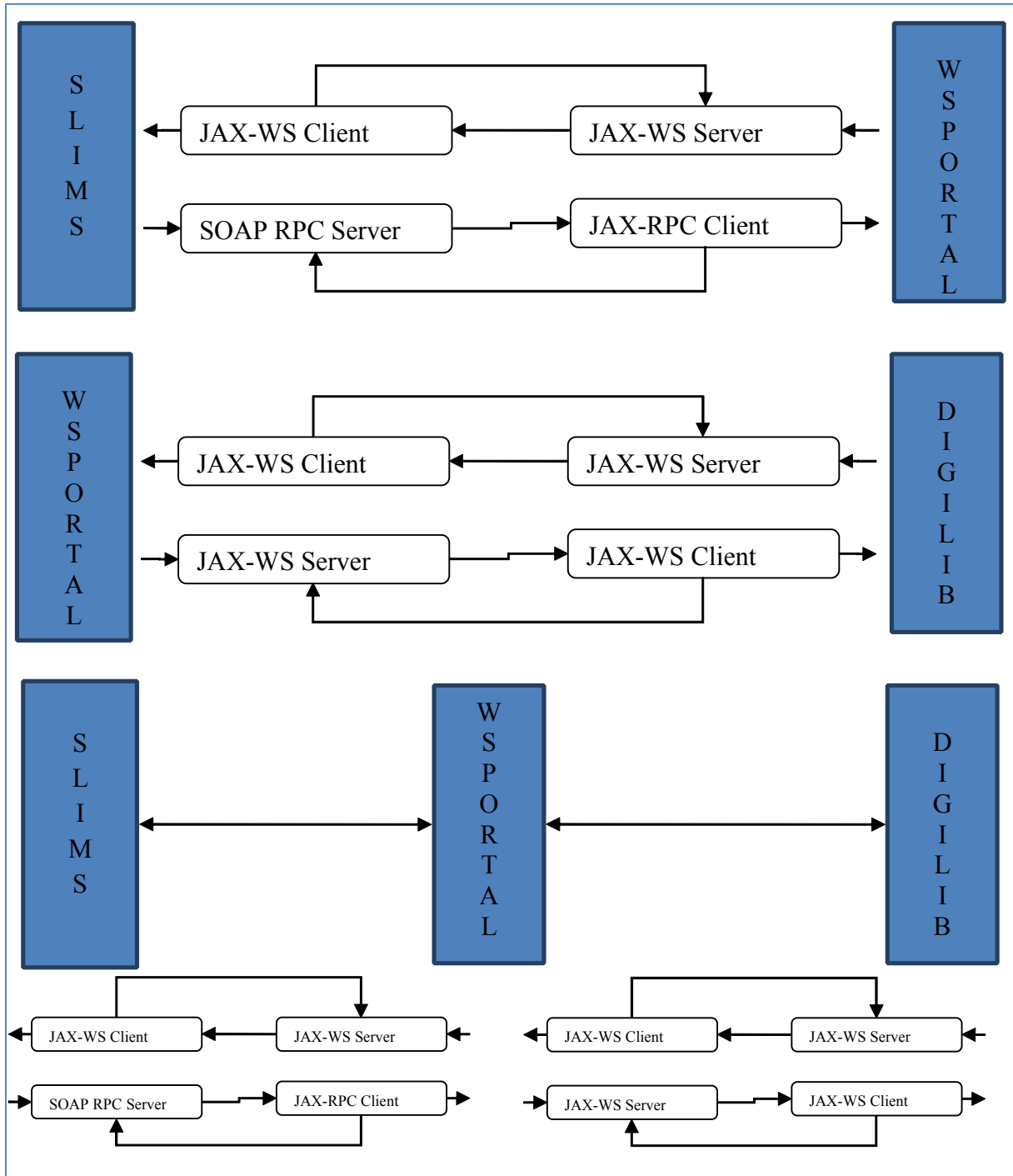
Kabupaten di Gorontalo paling ujung yang berbatasan dengan Sulawesi Tengah ini baru mengoperasikan Perpustakaan Daerah selama 2 bulan, namun penggunaan TIK di perpustakaan ini sudah cukup baik untuk pengelolaan perpustakaan. Memiliki koleksi buku hampir 32.000 ini sudah memprakarsai program kerjasama dengan beberapa perpustakaan di sekolah-sekolah. Perpustakaan ini menggunakan aplikasi SLIMS untuk pengelolaan perpustakaan. Dari hasil survey yang diperoleh maka yang dibutuhkan adalah penyesuaian format pertukaran data dengan menggunakan arsitektur GLN yang akan diimplementasikan.

## **6) Perpustakaan Daerah Kabupaten Bone Bolango**

Kantor perpustakaan daerah di Kabupaten Bone Bolango masih tergabung dengan Arsip Daerah. Perpustakaan daerah di Kabupaten Bone Bolango sama halnya dengan perpustakaan daerah lainnya dimana menggunakan aplikasi SLIMS sebagai aplikasi pengelolaan perpustakaan. Maka dari hasil survey tersebut dibutuhkan penyesuaian format pertukaran data dengan menggunakan arsitektur GLN yang akan diimplementasikan.

### **5.1.2 Desain Arsitektur**

Arsitektur yang didesain pada GLN untuk dapat mempertukarkan data antar perpustakaan daerah yaitu arsitektur client-server memanfaatkan teknologi web service untuk menjembatani aplikasi dengan platform yang berbeda-beda dimana terdapat sebuah Portal (GLN) yang akan menjembatani pertukaran data menggunakan JAX Server dan JAX Client. Perpustakaan daerah yang menggunakan SLIMS memanfaatkan library NuSOAP (SOAP RPC), sedangkan pada perpustakaan daerah yang menggunakan DIGILIB memanfaatkan J2EE (JAX Server dan Client), seperti yang terlihat pada Gambar 10. berikut :

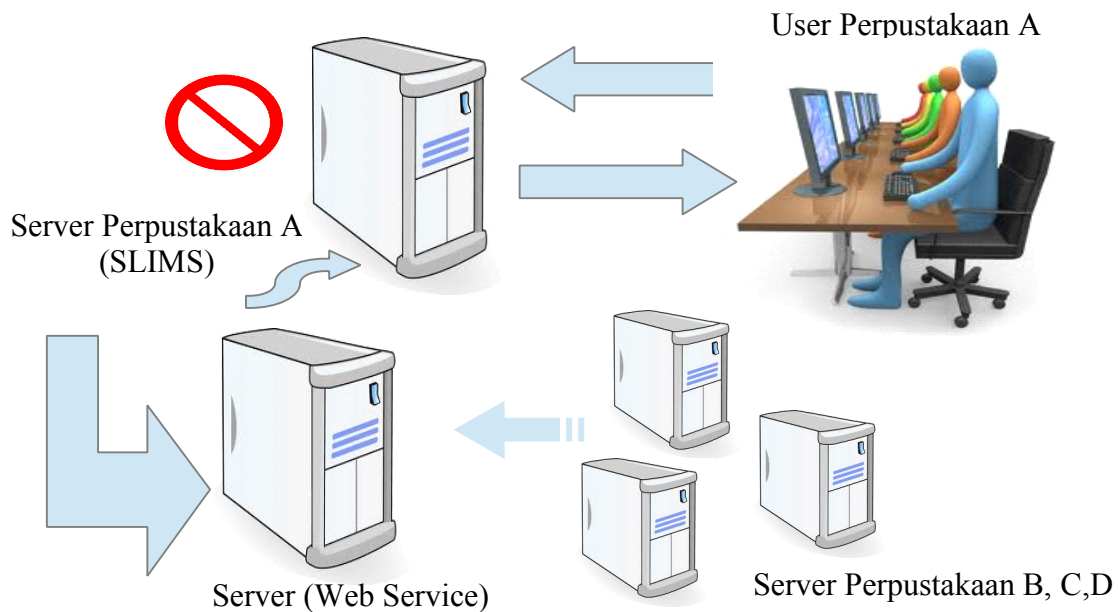


**Gambar 10. Desain Teknis Arsitektur GLN**

Skenario pertukaran data menggunakan teknologi webservice dapat dijabarkan sebagai berikut :

1. Pengguna pada perpustakaan daerah A yang menggunakan SLIMS melakukan pencarian koleksi buku

2. Koleksi buku tidak terdapat pada database perpustakaan A
3. Perpustakaan A mengirimkan request ke server untuk mencari koleksi buku pada perpustakaan daerah yang lain
4. Perpustakaan daerah lain mengirimkan balasan ke server atas koleksi buku yang dicari
5. Server mengirimkan informasi hasil pencarian buku pada perpustakaan A dengan menyertakan lokasi fisik buku yang diminta



**Gambar 11. Skenario proses kerja arsitektur GLN**

### 5.1.3 Pembuatan Prototipe GLN

Berdasarkan arsitektur GLN, tahapan selanjutnya adalah pembuatan prototipe GLN. Prototipe ini dibuat dengan bahasa pemrograman yang dapat menjembatani pertukaran data pada enam aplikasi perpustakaan daerah dengan platform yang berbeda-beda. Pada tahapan pembuatan prototipe GLN ini sebelumnya dilakukan perancangan standarisasi antarmuka, perancangan standarisasi pertukaran data, perancangan XMLParser dan perancangan Function Interface.

#### 5.1.3.1 Perancangan Standarisasi Antarmuka

Antarmuka dengan menggunakan teknologi webservice JAX-WS/JAX-RPC dan SOAP

### 5.1.3.2 Perancangan Standarisasi Pertukaran Data

Standarisasi jenis data yang dipertukarkan antar perpustakaan daerah yaitu :

id; **nomor registrasi buku pada perpustakaan setempat**  
title; **judul buku**  
subject; **kategori buku**  
author; **pengarang**  
publisher; **penerbit**  
year; **tahun terbit**  
language; **bahasa pengantar buku**  
type; **tipe atau wujud dari buku seperti: hardcopy dan atau softcopy**  
issn; **nomor standar buku**  
location; **alamat server tempat buku**  
server; **nama server tempat buku**  
wsdl; **url wsdl server**  
manager; **pengelola**  
link; **URI untuk mendownload buku**  
error; **pesan kesalahan fungsi**

Untuk standarisasi format XML yaitu :

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    <buku>
      <id>...</id>
      <title>...</ title >
      <subject>...</ subject >
      <author>...</author >
      <publisher>...</publisher >
      <year>...</year>
      <type>...</type>
      <location>...</location>
      <server>...</server>
      <wsdl>...</wsdl>
      <manager>...</manager>
      <link>...</link>
    </buku>
    <soap:Fault>
  </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

### 5.1.3.3 Perancangan XMLParser

#### a. Untuk SLIMS

Menggunakan library SOAP, baik sisi client maupun server

#### b. Untuk Digilib



- Menggunakan J2EE, pada sisi client dan server menggunakan JAX-WS
- c. **Untuk Portal Perpustakaan**  
Menggunakan J2EE, pada sisi client dan server menggunakan JAX-WS

#### 5.1.3.4 Perancangan Function Interface

##### ✓ **Perpustakaan Portal (GLN)**

###### a. **Sebagai Server:**

```
/*
Mendaftarkan server baru ke portal dengan parameter:
Location - alamat server
WsdL - alamat wsdl dari server
Nama - nama server
Auth - bentuk autentikasi dari server yang didaftarkan
Admin - bentuk autentikasi yang mendaftarkan server
*/
Void DaftarServer(location,wsdl,nama,auth,admin) ;

/*
Mengambil informasi diportal dengan parameter:
Title - judul buku yang dicari
Auth - bentuk autentikasi server terdaftar

Output:
List<data> - array yang dibungkus dengan SOAP
*/
/*
Portal akan melakukan iterasi pada setiap server selain server
yang request.
Portal mengakses fungsi GetData() pada webservice setiap
server
*/
List<data> GetInfo(title,author,subject,issn,auth) ;

/*
Mengambil Informasi Server yang telah tersambung dengan portal
*/
List<server> GetServer() ;
```

###### b. **Sebagai Client**

```
List<data> GetData(title,auth);
```

##### ✓ **Server Perpustakaan Lain**

###### a. **Sebagai Server:**

```
/*
Mengambil informasi diserver dengan parameter:
```

Title - judul buku yang dicari  
Auth - bentuk autentikasi server terdaftar

Output:  
List<data> - array yang dibungkus dengan SOAP  
\*/

```
List<data> GetData(title,auth);
```

## b. Sebagai Client

```
/*  
Portal mengakses fungsi GetInfo() pada webservice portal  
*/  
List<data> GetInfo(title,auth);
```

## ✓ Webservice server SLIMS

### Sql query:

```
<?php  
error_reporting(0);  
  
//include required class for build nnusoap web service server  
require_once('lib/nusoap.php');  
  
// Create server object  
$server = new soap_server();  
  
// configure WSDL  
$server->configureWSDL('PHP Web Services return array',  
'urn:returnArray');  
  
// Complex Type Struct for return array  
// definisi tiap field data $title, $subject, $author, $publisher,  
$year, $type, $language, $location, $link  
$server->wsdl->addComplexType('array_php',  
    'complexType',  
    'struct',  
    'all',  
    '',  
    array(  
        'id' => array('id' => 'id', 'type' => 'xsd:string'),  
        'title' => array('name' => 'title', 'type' => 'xsd:string'),  
        'subject' => array('name' => 'subject', 'type' =>  
'xsd:string'),  
        'author' => array('name' => 'author', 'type' =>  
'xsd:string'),  
        'publisher' => array('name' => 'publisher', 'type' =>  
'xsd:string'),  
        'year' => array('name' => 'year', 'type' => 'xsd:string'),  
        'type' => array('name' => 'type', 'type' => 'xsd:string'),  
        'language' => array('name' => 'language', 'type' =>  
'xsd:string'),  
        'location' => array('name' => 'location', 'type' =>  
'xsd:string'),  
        'link' => array('name' => 'link', 'type' => 'xsd:string')    )
```

```

    )
);

$server->wsdl->addComplexType('return_array_php',
    'complexType',
    'array',
    'all',
    'SOAP-ENC:Array',
    array(),
    array(
        array('ref'=>'SOAP-
ENC:arrayType', 'wsdl:arrayType'=>'tns:array_php[]')
    ),
    'tns:array_php'
);

$server->register('get_data',
    array(
        'title' => 'xsd:string',
        'subject' => 'xsd:string',
        'author' => 'xsd:string',
        'publisher' => 'xsd:string',
        'year' => 'xsd:string',
        'issn' => 'xsd:string'
    ),
    array('return' => 'tns:return_array_php'),
    $ns,
    $ns.'#get_data',
    'rpc',
    'encoded',
    'Returns array data in php web service'
);

function get_data($title,$subject,$author,$publisher,$year,$issn)
{
    //input = $title,$subject,$author,$publisher,$year,$issn
    //output = $title, $subject, $author, $publisher, $year, $type,
    $language, $location, $link
    //periksa parameter kosong
    if($title==NULL) $title="-";
    if($subject==NULL) $subject="-";
    if($author==NULL) $author="-";
    if($publisher==NULL) $publisher="-";
    if($year==NULL) $year="-";
    if($issn==NULL) $issn="-";

    //koneksi ke database
    $hostmysql = "localhost";
    $username = "root";
    $password = "divv41881dbq";
    $database = "senayan";
    $array_rtr=array();

    $conn
    mysql_connect("$hostmysql","$username","$password");
    if (!$conn) die ("Koneksi gagal");
    mysql_select_db($database,$conn) or die ("Database tidak
ditemukan");

    //query data sesuai parameter

```

```

$query = "SELECT
        b.biblio_id,
        b.title,
        b.edition,
        b.publish_year,
        b.classification,
        b.spec_detail_info,
        g.gmd_name AS `gmd`,
        pb.publisher_name AS `publisher`,
        b.isbn_issn,
        lg.language_name AS `language`,
        au.author_name
    FROM biblio AS b
    LEFT JOIN mst_gmd AS g ON b.gmd_id = g.gmd_id
    LEFT JOIN mst_publisher AS pb ON
b.publisher_id = pb.publisher_id
    LEFT JOIN mst_language AS lg ON b.language_id
= lg.language_id
    LEFT JOIN (
        SELECT bib.biblio_id, ma.author_name
    FROM biblio_author AS ba
        LEFT JOIN mst_author AS ma ON
ba.author_id = ma.author_id
        LEFT JOIN biblio AS bib ON
ba.biblio_id = bib.biblio_id
    )
    AS au ON b.biblio_id =
au.biblio_id
    WHERE b.title LIKE '%$title%' OR
au.author_name LIKE '%$author%' OR
pb.publisher_name LIKE '%$publisher%'
OR
        b.publish_year LIKE '%$year%' OR
        b.isbn_issn LIKE '%$issn%'
";

$result = mysql_query($query);
$numrows = mysql_num_rows($result);

//fetch data
/* Ambil data sesuai permintaan parameter
$title,$subject,$author,$publisher,$year,$issn */
$i=0;
while($row = mysql_fetch_array($result)){
    $array_rtr[$i]['id']=$row[0];//$array_rtr[$i]['id']=$row[biblio
_id];
    $array_rtr[$i]['title']=$row[1];//$array_rtr[$i]['title']=$row[
title];
    $array_rtr[$i]['subject']="-";
    $array_rtr[$i]['author']=$row[10];//$array_rtr[$i]['author']=$r
ow[author_name];
    $array_rtr[$i]['publisher']=$row[7];//$array_rtr[$i]['publisher
']=$row[publisher];
    $array_rtr[$i]['year']=$row[3];//$array_rtr[$i]['year']=$row[pu

```

```

blish_year];
        $array_rtr[$i]['type']="Physical Book";

        $array_rtr[$i]['language']=$row[9];//$array_rtr[$i]['language']
=$row[language];
        $array_rtr[$i]['location']="Perpus          Jurusan
Informatika - SLIMS";

        $array_rtr[$i]['link']="www.fatek.ung.ac.id/informatika/perpus";
        $i++;
        //$title, $subject, $author, $publisher, $year,
$type, $language, $location, $link
    }
    //tutup database
    //kembalikan data permintaan
    //$return=array();
    //for($i=0;$i<=$numrows;$i++){
    //    $return[$i]=$array_rtr[$i];
    //}
    return $array_rtr;//$return;
}
$server->service($HTTP_RAW_POST_DATA);
?>

```

## ✓ Webservice client SLIMS

```

//
=====
===== //
    //echo '<div>';
    include "wsclient.php";
    error_reporting(0);
    // includes nusoap class
    require_once('../lib/nusoap.php');
    // Create object
    //change according your server settings
    $client = new
nusoap_client('http://localhost:8080/Perpustakaan_Server/Perpustakaan
Service?wsdl', true);

    //use curl --set here to true
    //$client->use_curl=true;

    // Check for an error
    $err = $client->getError();
    if ($err) {
        // error if any
        echo 'Constructor error' . $err . ' ';
    }
    if($title!=""){
        $result = $client->call(
array('title' => $title,
    'subject'=>';',
    'author'=>';',

```

```

        'publisher'=>';',
        'year'=>';',
        'issn'=>';',
        'issn'=>';'
    ),
    true
        }else if($author!=""){
            $result = $client->call( 'getData',
array('title' => ';',
        'subject'=>';',
        'author'=>$author,
        'publisher'=>';',
        'year'=>';',
        'issn'=>';',
        'issn'=>';'
    ),
    true
        } else if($publisher!=""){
            $result = $client->call( 'getData',
array('title' => ';',
        'subject'=>';',
        'author'=>';',
        'publisher'=>$publisher,
        'year'=>';',
        'issn'=>';',
        'issn'=>';'
    ),

```

```

true

);

}
// jika terjadi kesalahan tampilkan
if ($client->fault) {
    echo 'Fault';
    print_r($result);
    echo "\n";
} else {
    // Check for errors
    $err = $client->getError();
    if ($err) {
        // Display the error
        echo 'Error ' . $err . "\n";
    } else {
        // Display the result
        $jum=0;

        if($result!=false){
            foreach($result as $key=>$val){
                $jum = count($val);

                if($val[0]['id']== ""){$jum=0;}
            }
            if($jum>0){

                echo "<h2>Hasil dari server

                //echo '<table border="1">';
                //echo '<td>ID</td>';
                //echo '<td>TITLE</td>';
                //echo '<td>AUTHOR</td>';
                //echo '<td>PUBLISHER</td>';
                //echo '<td>YEAR</td>';
                $jum=0;//$jum2=0;

                foreach($result as
                $key=>$val){

                    //go through array
                    //print_r($val);
                    $jum = count($val);
                    $num=0;

                    for($i=0;$i<$jum;$i++){

                        if($val[$i]['id'] != ""){

                            if($num==0){echo '<div class="item alterList" >';}else{echo
                            '<div class="item alterList2" >';}

                            echo '<a
                            href="'. $val[$i]['id']. ' " class="titleField" title="Record
                            Detail">'. $val[$i]['title']. '</a><br />';

                            echo
                            '<div class="subItem authorField"><b>Author(s)</b> :
                            '. $val[$i]['author']. ' - </div>';

```

```

    ' <div class="subItem authorField"><b>Publisher(s)</b>          : echo
    ' .$val[$i]['publisher'] .' - </div>';
    ' <div class="subItem authorField"><b>Year(s)</b>          : echo
    ' .$val[$i]['year'] .' - </div>';
        if($num==0){$num=1;}else{$num=1;}
    ' </div><br>';
    }
}
echo "<h2>Hasil dari server
lokal :</h2>";
}else{
echo "<h2>Hasil dari server
lain : $jum ditemukan</h2>";
}
}
}
//echo '</div>';
//
===== //
===== //

```

## ✓ Webservice server DIGILIB

```

/**
 * Web service operation
 */
@WebMethod(operationName = "getData")
public List<Data> getData(@WebParam(name = "title")
String title, @WebParam(name = "subject")
String subject, @WebParam(name = "author")
String author, @WebParam(name = "publisher")
String publisher, @WebParam(name = "year")
String year, @WebParam(name = "issn")
String issn) {
    // java.util.List<org.perpustakaan.model.Buku>
    String sql=null;
    int jumitem=0;

    if(!subject.isEmpty()) {subject=""; jumitem++;}
    if(!author.isEmpty()) {author=""; jumitem++;}
    if(!publisher.isEmpty()) {publisher=""; jumitem++;}
    if(!year.isEmpty()) {year=""; jumitem++;}
    if(!issn.isEmpty()) {issn=""; jumitem++;}
}

```



```

ArrayList<Data> lb = new ArrayList<Data>();
//String[][] lb = null;
try{
    //sql = "select * from BUKU where JUDUL like '%" + title + "%"
    //      + " or PENGARANG1 like '%" + author + "%' or PENGARANG2 like
"
    //      + "%'" + author + "%' or PENERBIT like '%" + publisher + "%' "
    //      + "or TAHUN like '%" + year + "%'";
    if(title.isEmpty()){title="";}
    sql = "select * from BUKU where JUDUL like '%" + title +
        "%' or PENGARANG1 like '%" + author +
        "%' or PENGARANG2 like '%" + author +
        "%' or PENERBIT like '%" + publisher +
        "%' or TAHUN like '%" + year + "%'";
    Class.forName ("com.mysql.jdbc.Driver").newInstance ();
    conn = DriverManager.getConnection (url, userName, passWord);
    Statement s = (Statement) conn.createStatement();
    s.executeQuery(sql);
    ResultSet rs = s.getResultSet();

    //int i=0;
    while (rs.next()) {
        Data buku = new Data();
        buku.setPesanError("");
        buku.setId(rs.getString("ID_BUKU"));
        buku.setTitle(rs.getString("JUDUL"));
        buku.setSubject("-");
        buku.setAuthor(rs.getString("PENGARANG1")+";
"+rs.getString("PENGARANG2"));
        buku.setPublisher(rs.getString("PENERBIT"));
        buku.setYear(rs.getString("TAHUN"));
        buku.setLanguage(rs.getString("BAHASA"));
        buku.setType("Digital Book");
        buku.setLocation("Perpustakaan Digital Service");
        buku.setLink("ht
tp://localhost:8080/Perpustakaan_Server/PerpustakaanService");
        /*
        lb[i][0]="";
        lb[i][1]="";
        lb[i][2]="";
        lb[i][3]="";
        lb[i][4]="";
        lb[i][5]="";
        lb[i][6]="";
        lb[i][7]="";
        lb[i][8]="";
        lb[i][9]="";

```

```

        lb[i][0]=rs.getString("ID_BUKU").toString();
        lb[i][1]=rs.getString("JUDUL").toString();
        lb[i][2]="-";

lb[i][3]=rs.getString("PENGARANG1").toString()+" "+rs.getString("PENGARANG
2");

        lb[i][4]=rs.getString("PENERBIT").toString();
        lb[i][5]=rs.getString("TAHUN").toString();
        lb[i][6]=rs.getString("BAHASA").toString();
        lb[i][7]="Digital Book";
        lb[i][8]="Perpustakaan Web Service";

lb[i][9]="http://localhost:8080//Perpustakaan_Server//PerpustakaanService";
        i++;
    */
        lb.add(buku);
    }
    sql="";
    s.close();
    rs.close();
    conn.close();
    return lb;

} catch(Exception e){
    Data buku = new Data();
    buku.setPesanError(e.getMessage());
    lb.add(buku);
/*
    lb[0][0]="-";
    lb[0][1]="-";
    lb[0][2]="-";
    lb[0][3]="-";
    lb[0][4]="-";
    lb[0][5]="-";
    lb[0][6]="-";
    lb[0][7]="-";
    lb[0][8]="-";
    lb[0][9]="-";
*/
    return lb;
}

```

### ✓ Webservice client DIGILIB

```

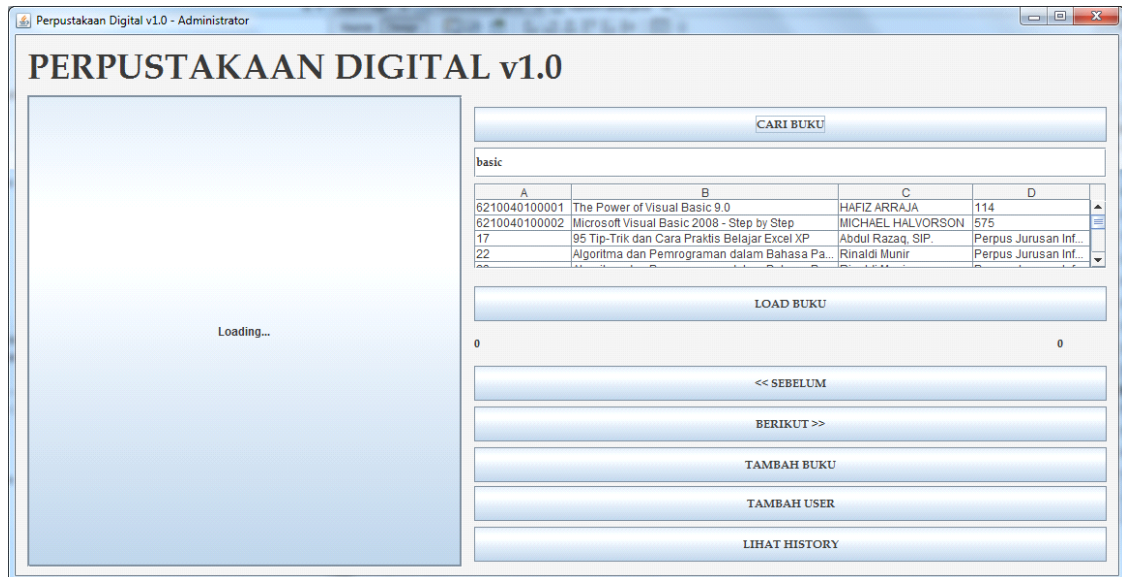
arrayDigiLib = digilib.getPHPWebServicesReturnArrayPort().get_data("",
"", "", "", "", "");

```

## 5.1.4 Ujicoba Prototipe GLN

### 1. Pengujian Client Digilib ke Server SLIMS

Sebelumnya web service di kedua server (SLIMS & Digilib) harus online, selanjutnya login di client Digilib untuk masuk admin dan lakukan pencarian buku.

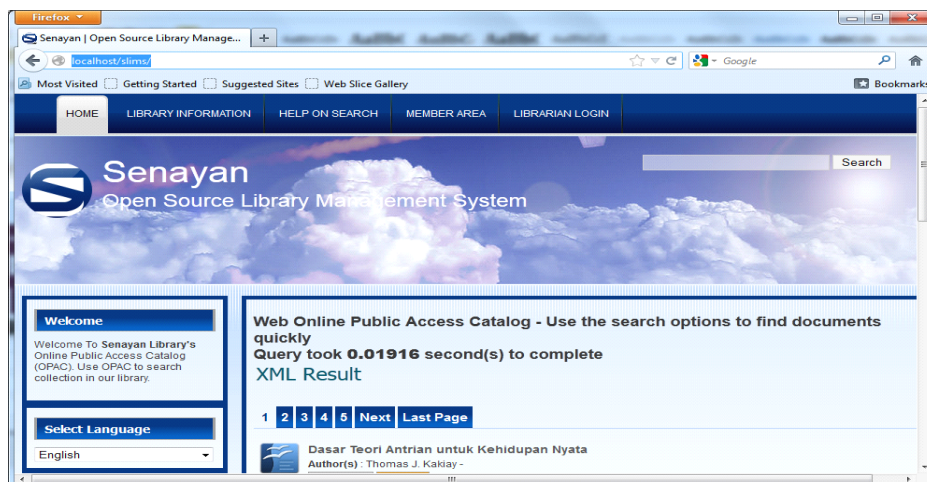


**Gambar 12. Digilib ke SLIMS**

Aplikasi client dapat menampilkan pencarian dari server Digilib dan server SLIMS dan menempatkan hasil pencarian pada table yang sama. Baris 1&2 hasil pencarian dari server Digilib dan baris-baris selanjutnya adalah hasil dari server SLIMS.

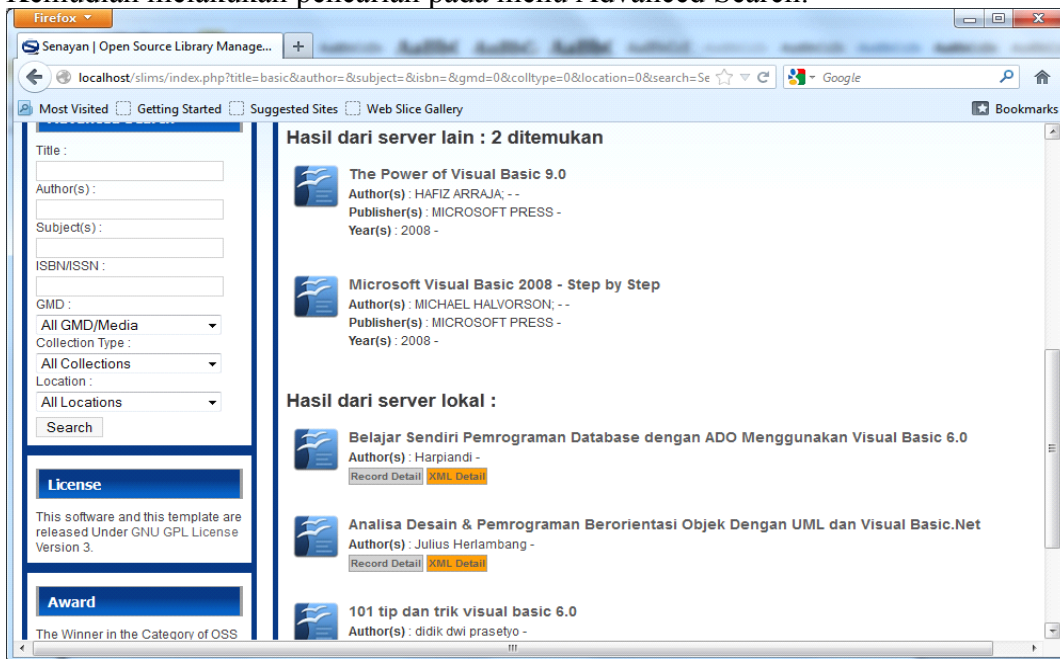
### 2. Pengujian Client SLIMS ke Server Digilib

Jalankan browser untuk masuk situs aplikasi SLIMS:



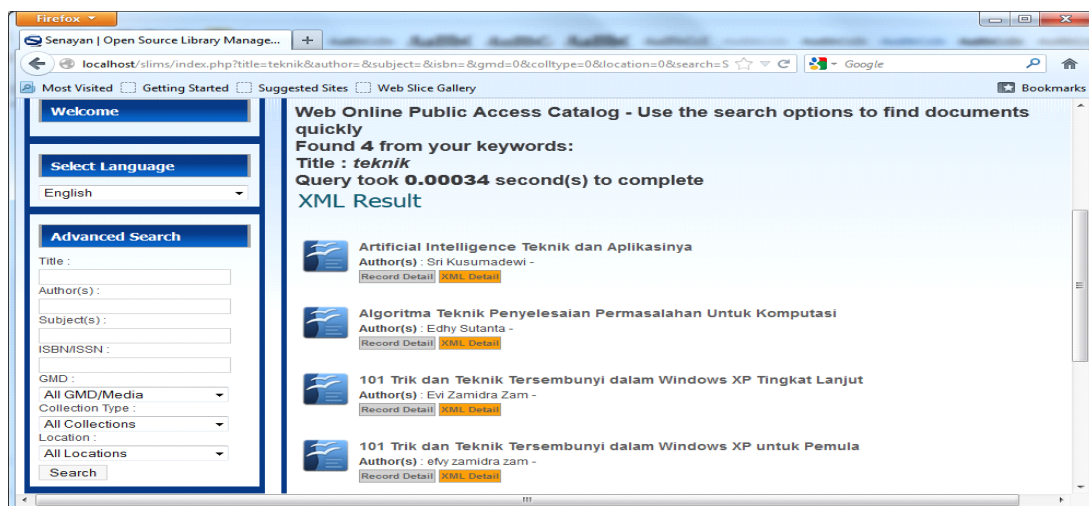
**Gambar 13. SLIMS ke Server Digilib**

Kemudian melakukan pencarian pada menu Advanced Search:



**Gambar 14. Advanced Search**

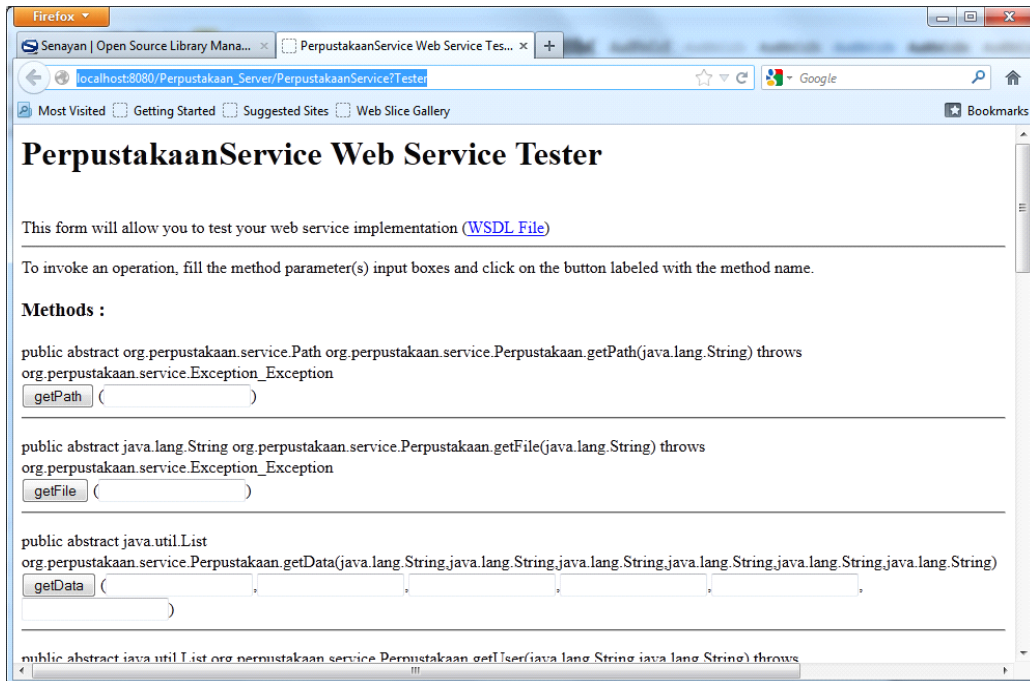
Aplikasi client dapat menampilkan pencarian dari server Digilib dan server SLIMS dan menempatkan hasil pencarian pada halaman yang sama dengan keterangan. Aplikasi client akan menampilkan hanya hasil yang didapatkan, apabila hanya didapat dari server SLIMS/local maka tampilan sebagai berikut:



**Gambar 15. Hasil dari SLIMS local**

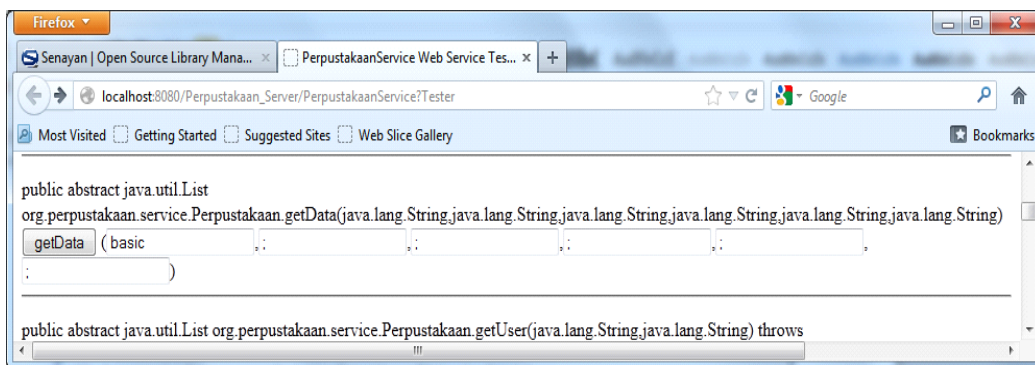
### 3. Pengujian Server Digilib

Tes server Digilib dengan menggunakan web browser dan mengetik pada addressbar: `http://localhost:8080/Perpustakaan_Server/PerpustakaanService?Tester` dan menampilkan:



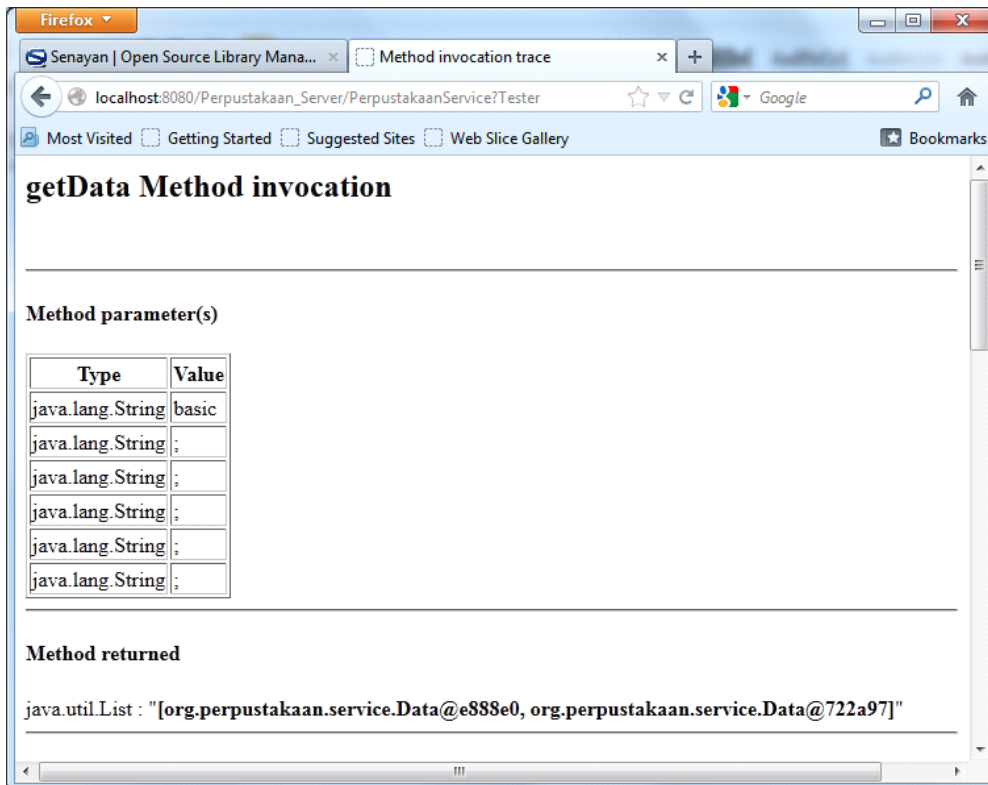
**Gambar 16. Server Digilib**

Selanjutnya menguji fungsi `get_data()` pada server dengan mengisi pada parameternyam parameter yang lain harus diisi dengan karakter `”;` untuk mencegah query Mysql melakukan exception:

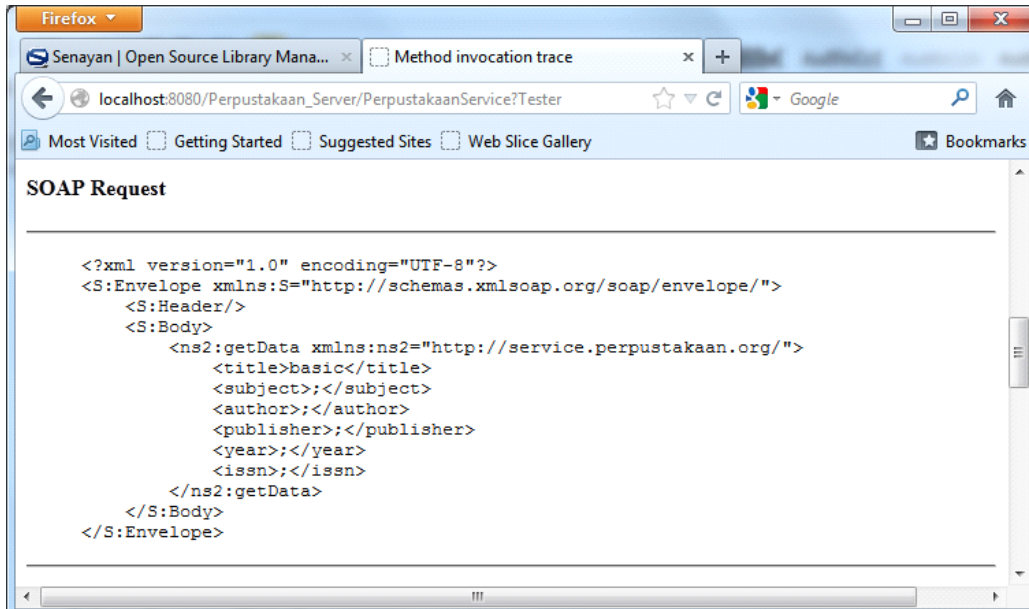


**Gambar 17. Pengujian fungsi getData()**

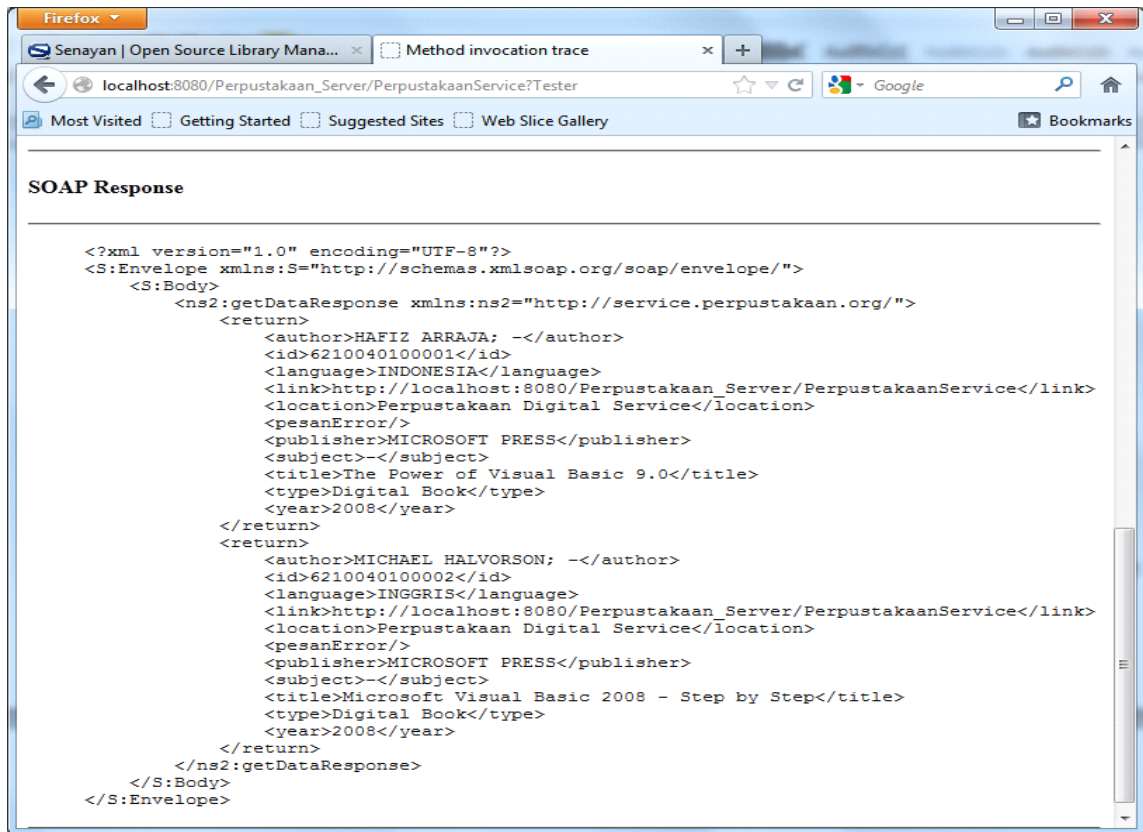
Selanjutnya menghasilkan sebagai berikut:



**Gambar 18. Hasil Pengujian getData() - 1**



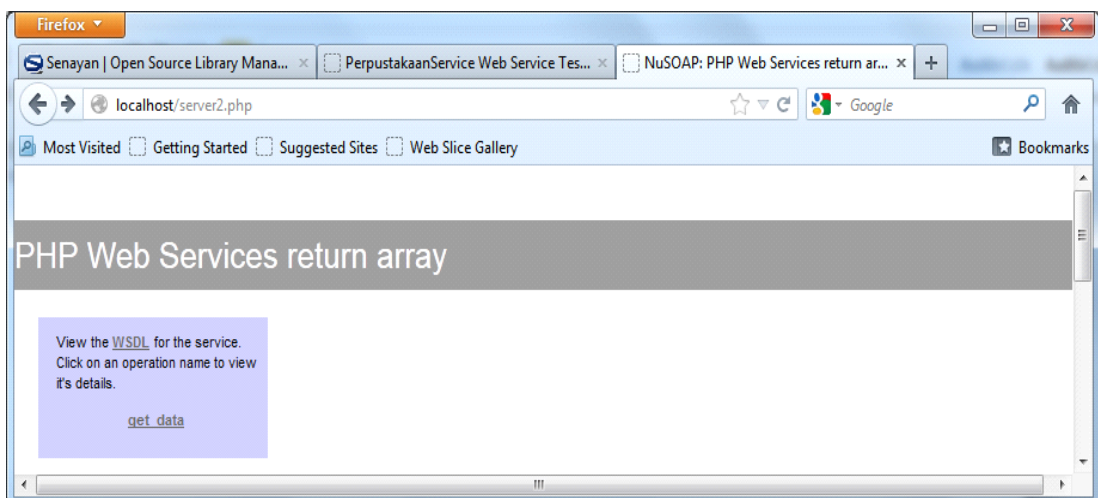
**Gambar 19. Hasil Pengujian getData() - 2**



**Gambar 20. Hasil Pengujian getData() - 3**

#### 4. Pengujian Server SLIMS

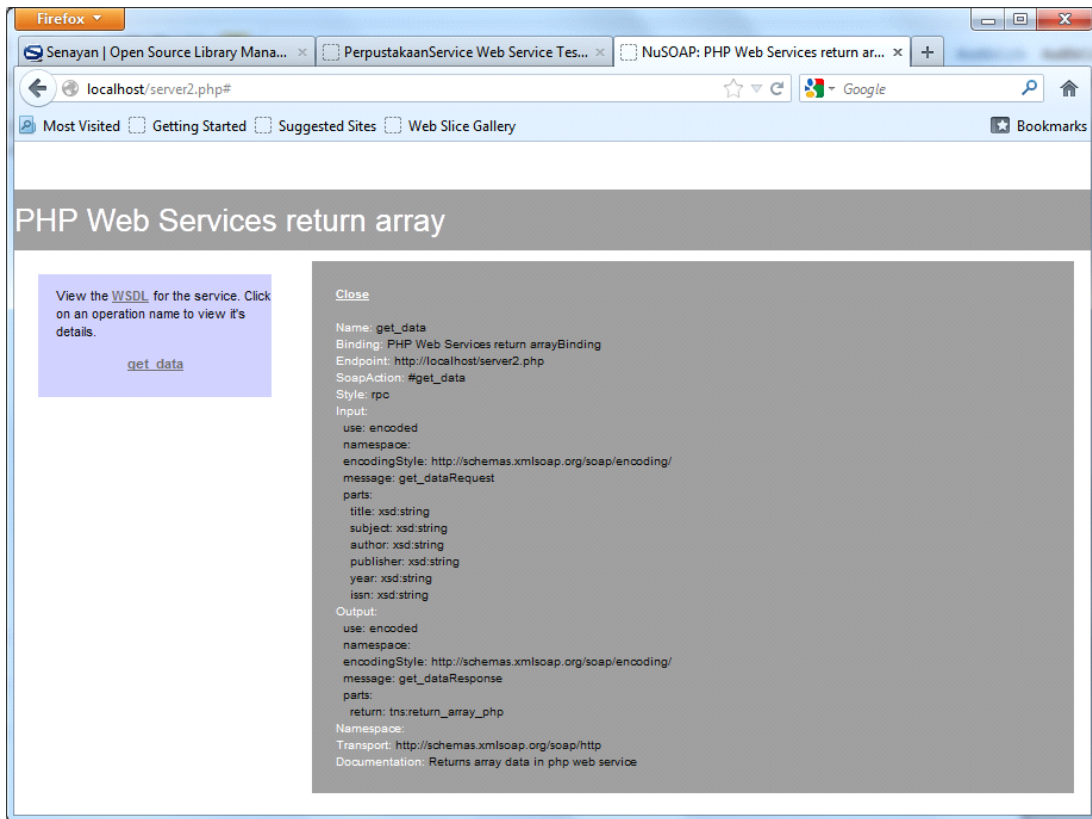
Dengan mengakses langsung file aplikasinya dari browser: <http://localhost/server2.php>



**Gambar 21. Pengujian server SLIMS**

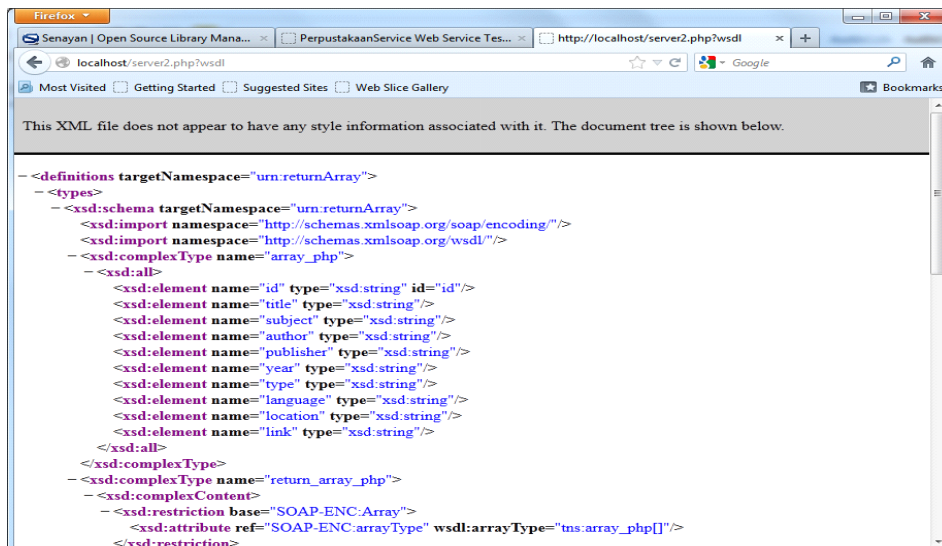


Untuk melihat struktur fungsi dari webservice SLIMS dapat klik pada link “get\_data”



**Gambar 22. Link get\_data**

Untuk melihat wsdl yang degenerate dapat klik pada link WSDL:



**Gambar 23. Link WSDL**

Aplikasi client dapat menampilkan pencarian dari server Digilib dan server SLIMS dan menempatkan hasil pencarian pada table yang sama. Baris 1&2 hasil pencarian

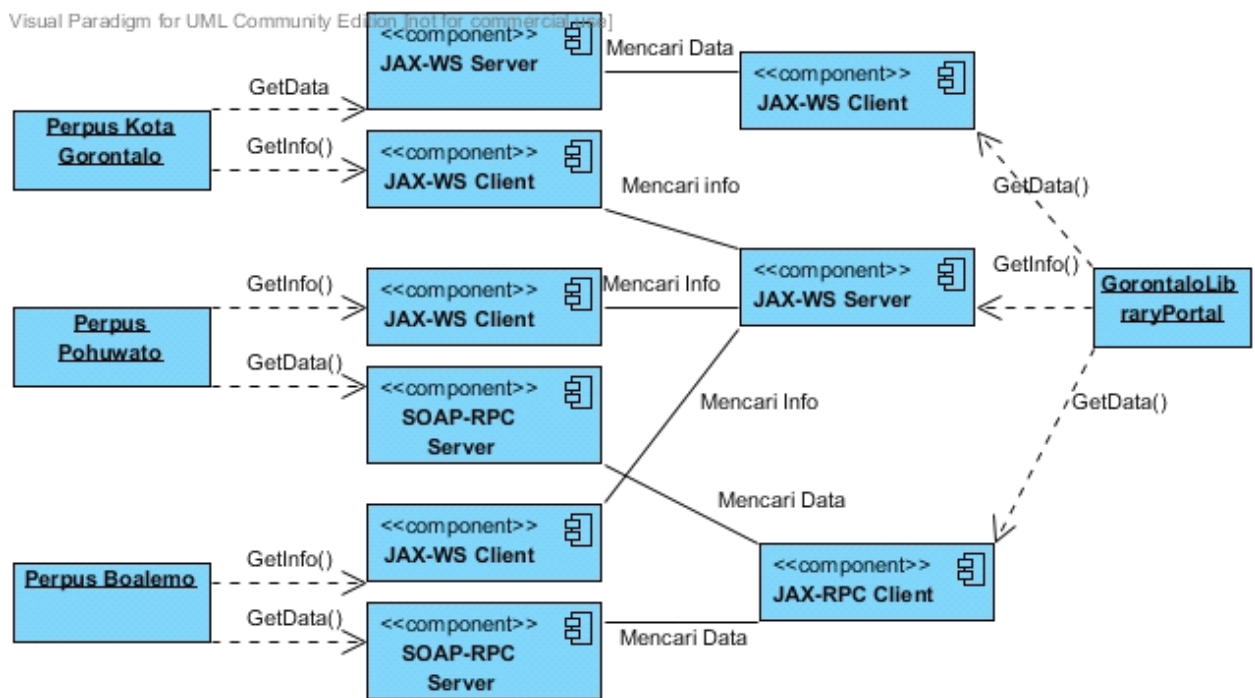


dari server Digilib dan baris-baris selanjutnya adalah hasil dari server SLIMS.

## 5.1.5 Pembuatan Aplikasi

### 5.1.5.1 Component Diagram

Component diagram menggambarkan komponen dalam sistem dan hubungan diantara komponen tersebut. Komponen yang dimaksud adalah modul fisik dari kode. Berikut ini adalah component diagram dari GLN :



**Gambar 24. Component Diagram**

### 5.1.6 Implementasi

Implementasi sistem yang dilakukan pertama kali adalah dengan melakukan hosting VPS Server, hal ini dilakukan karena aplikasi GLN yang dibangun merupakan aplikasi berbasis Java (J2EE) sehingga tidak hanya dibutuhkan hosting web melainkan hosting server sebagai server GLN (WSPortal). Penyedia layanan hosting server yang kami pilih adalah rumahweb.com karena menyediakan layanan VPS server yang dibutuhkan dalam menjalankan aplikasi GLN tersebut.

#### Spesifikasi server GLN :

CPU : 800 MHz (4x200)

Diskspace : 15 GB

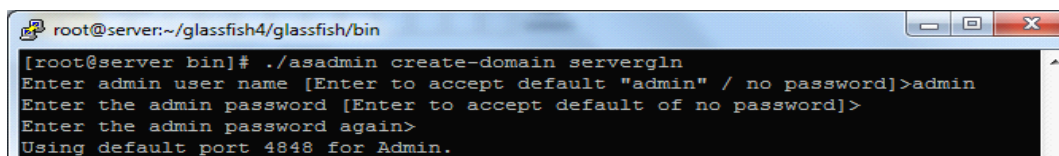
Memory	: 1 GB
VSwap	: 512 MB
Sistem Operasi	: GNU/Linux CentOS 6.4
Bandwidth International	: 1024 kbps (1:8)
Bandwidth IIX	: 1 Gbps
Services (Application)	: Web Server (httpd), Database Server (MySQL-Server), Java 1.7.0_45 , Glassfish Server 4.0

### 5.1.6.1 Glassfish Server

Glassfish Server digunakan sebagai Web dan Webservice Server untuk menjalankan aplikasi GLN. Versi Glassfish server yang digunakan adalah Glassfish Server 4.0. Untuk menjalankan Glassfish server langkah-langkah yang dilakukan adalah :

- ✓ Membuat domain baru :

Perintah yang digunakan adalah **./asadmin create-domain <nama-domain>**, ketika server dibuat maka perlu didefinisikan user dan passwordnya seperti yang terlihat pada gambar 25 berikut :



```

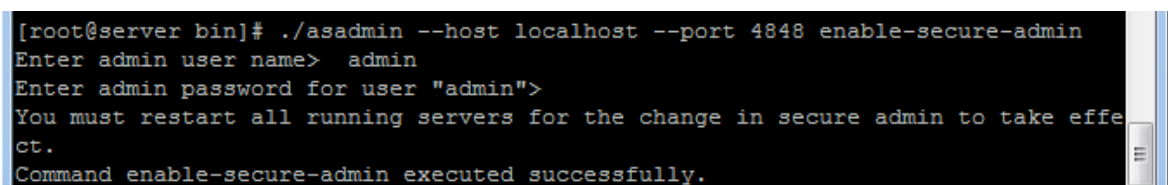
root@server:~/glassfish4/glassfish/bin
[root@server bin]# ./asadmin create-domain servergln
Enter admin user name [Enter to accept default "admin" / no password]>admin
Enter the admin password [Enter to accept default of no password]>
Enter the admin password again>
Using default port 4848 for Admin.

```

**Gambar 25. Membuat domain baru**

- ✓ Mengaktifkan secure admin :

Perintah yang digunakan adalah **./asadmin --host <nama-host> --port <nomor-port> enable-secure-admin**. Ketika mengaktifkan secure admin maka diperlukan user dan password, seperti yang terlihat pada gambar 26 berikut :



```

[root@server bin]# ./asadmin --host localhost --port 4848 enable-secure-admin
Enter admin user name> admin
Enter admin password for user "admin">
You must restart all running servers for the change in secure admin to take effect.
Command enable-secure-admin executed successfully.

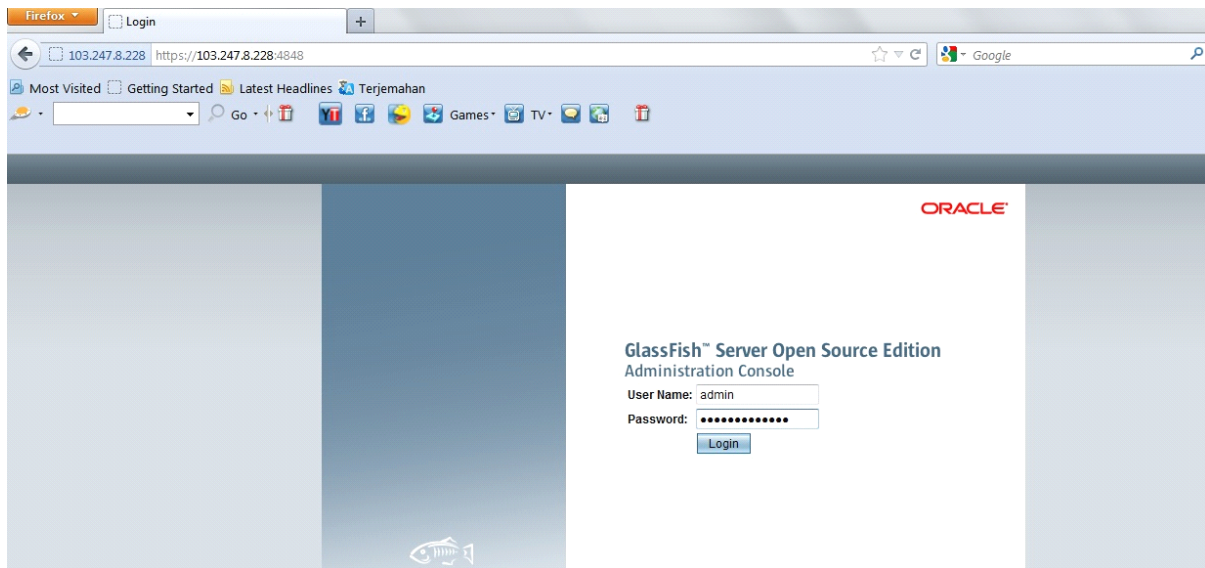
```

**Gambar 26. Mengaktifkan secure admin**

- ✓ Mengakses Glassfish Server :

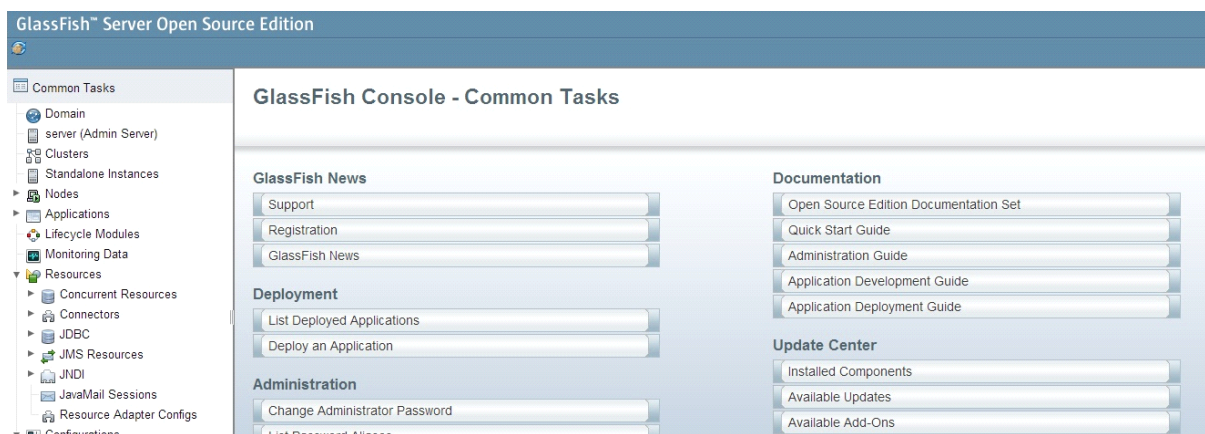
Untuk dapat mengakses glassfish server diperlukan web browser untuk mengakses halaman utama glassfish server, cukup dengan mengetikkan pada *address bar* alamat dari glassfish server dan port yang digunakan yaitu <https://103.247.8.228:4848>, maka akan muncul halaman

login glassfish server kemudian diperlukan username dan password untuk masuk kedalam halaman utama, seperti yang terlihat pada gambar 27 berikut :



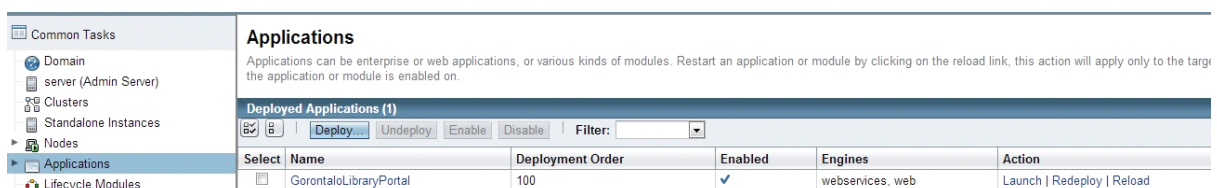
**Gambar 27. Halaman login glassfish server**

Setelah berhasil login selanjutnya akan masuk pada halaman utama glassfish server (*common tasks*) seperti yang terlihat pada gambar 28 berikut :



**Gambar 28. Halaman utama glassfish server**

Untuk menambah aplikasi GLN yang telah dibuat sebelumnya, maka perlu untuk mengakses menu applications dan menambahkan file berekstensi .raw untuk dikembangkan (*deploy*) seperti yang terlihat pada gambar 29 berikut :



**Gambar 29. Deploy applications**

Untuk melihat apakah aplikasi yang dibangun sudah dapat berjalan atau tidak dapat diakses halaman WSDL seperti yang terlihat pada gambar 30 berikut :

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

<!--
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.3 (tags/2.3-7528; 2013-04-29T19:34:10+0000) JAXWS-RI/2.2.8 JAXWS/2.2 svn-revisi
-->
<!--
  Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.3 (tags/2.3-7528; 2013-04-29T19:34:10+0000) JAXWS-RI/2.2.8 JAXWS/2.2 svn-revisi
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wspi_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://Portal.Digilib/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://Portal.Digilib/" name="GorontaloLibraryPortal">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://Portal.Digilib/" schemaLocation="http://10.12.113.214:8080/GorontaloLibraryPortal/GorontaloLibraryPortal?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="DaftarServer">
    <part name="parameters" element="tns:DaftarServer"/>
  </message>
  <message name="DaftarServerResponse">
    <part name="parameters" element="tns:DaftarServerResponse"/>
  </message>
  <message name="Exception">
    <part name="fault" element="tns:Exception"/>
  </message>
  <message name="GetInfo">
    <part name="parameters" element="tns:GetInfo"/>
  </message>
  <message name="GetInfoResponse">
    <part name="parameters" element="tns:GetInfoResponse"/>
  </message>
</definitions>
```

**Gambar 30. Halaman WSDL**

## 5.2 Pembahasan

Berbagi data koleksi antar perpustakaan daerah dibutuhkan infrastruktur yang memadai terutama terkait dengan konektivitas internet. Namun yang paling penting adalah kesediaan pimpinan dan pengelola perpustakaan daerah untuk mau berbagi data koleksi perpustakaan.

Kendala yang dihadapi dalam penerapan berbagi data koleksi antar perpustakaan diantaranya adalah perubahan teknologi antara aplikasi yang berbasis desktop (digilib) dengan aplikasi yang berbasis web (SLIMS) sangat signifikan. Diantaranya aplikasi yang dikembangkan sebelumnya menggunakan Java (J2EE dan JDK) versi 6 sudah tidak didukung lagi ketika Java yang digunakan adalah versi 7, sedangkan teknologi web (PHP) tidak menyesuaikan perkembangan yang terjadi pada Java.

## **BAB 6**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil survey keenam perpustakaan daerah, terdapat lima perpustakaan daerah yang menggunakan aplikasi pengelola perpustakaan (SLIMS) dan satu perpustakaan daerah yang belum menggunakan aplikasi. sehingga dikembangkan aplikasi untuk perpustakaan daerah tersebut yang berbeda platform dengan kelima perpustakaan daerah lain.

Kendala yang dihadapi pada keenam perpustakaan daerah adalah belum dapat melakukan pertukaran data disebabkan aplikasi yang digunakan pada perpustakaan daerah tersebut masih berdiri sendiri.

Untuk mengatasi kendala tersebut dirancang sebuah arsitektur yang menjembatani pertukaran data antar perpustakaan daerah dengan platform yang berbeda-beda.

#### **6.2 Saran**

Implementasi aplikasi GLN yang telah dilaksanakan baru pada perpustakaan daerah kota Gorontalo dan kabupaten Boalemo. Oleh karenanya, diperlukan penelitian lanjutan untuk mengimplementasikan aplikasi GLN pada perpustakaan daerah yang tersisa.

## DAFTAR PUSTAKA

- Aji, R.F dan Wibowo, W.C. (2012). *Arsitektur Pertukaran Data di Indonesia*. Proceedings of National Conference on Computer Science & Information Technology. Faculty of Computer Science. University of Indonesia.
- Arms, W. (2000). *Digital Libraries*. MIT Pers.
- Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., et al. (2007). *The DELOS Digital Library Reference Model: Foundations for Digital Libraries*. Basel, Switzerland: DELOS Network of Excellence on Digital Libraries .
- Drabenstott, K. (1994). *Analytical Review of The Library of The Future*. Washington, D.C.: Council Library Resources.
- Foster, I., & Kesselman, C. (2004). *The Grid: Blueprint for a New Computing Infrastructure, Second Edition*. San Fransisco, USA: Morgan Kaufmann Publisher, Elsevier Inc.
- Greenstein, D., & Thorin, S. E. (2002). *The Digital Library: A Biography*. Washington, D.C.: Digital Library Federation.
- Kurniawan, A., & Hasibuan, Z. (2009). Education Resource Sharing in The Heterogeneous Environment Using Data Grid, Case Study:Indonesia Higher Education Network. *Proceedings of International Conference on Creative and Inovative Technology (ICCIT)* (pp. 66-71). Tangerang, Indonesia: STMIK Rahardja.
- Larson, R., & Sanderson, R. (2005). Grid-based Digital Libraries: Chesire3 and Distributed Retrieval. *JCDL*
- Mulyanto, Arip dan Jassin, RT. (2012). Peningkatan Akses Koleksi Perpustakaan Menggunakan Perpustakaan Digital berbasis Data Grid. Prosiding Seminar Nasional Ilmu Komputer Universitas Diponegoro. Semarang.
- Mulyanto, Arip. (2010). Berbagi Sumber Daya Pendidikan Menggunakan Perpustakaan Digital Berbasis Data Grid. Prosiding Seminar Internasional Konvensi APTEKINDO. Universitas Pendidikan Ganesha. Singaraja. Bali.
- Sebestyen-Pal, G., Banciu, D., Balint, T., Moscaiu, B., & Sebestyen-Pal, A. (2008). Towards a Grid-based Digital Library Management System. *Distributed and Parallel Systems* , 77-90.
- Yang, C.-T., & Ho, H.-C. (2005). Using Data Grid Technologies to Construct a Digital Library Environment . *Proceedings of the 3rd International Conference on Information Technology Research and Education (ITRE)* (pp. 338-392). Hsinchu, Taiwan: NTHU.