

SINKRONISASI DATA USER ANTARA SISTEM INFORMASI PERPUSTAKAAN DENGAN SISTEM INFORMASI AKADEMIK

Muhammad Nasir

^{1,2)}Fakultas Ilmu Komputer Universitas Bina Darma Palembang
Jl. Jend. A. Yani no. 12 Palembang Telp (0711)-515679
e-mail : nasir@mail.binadarma.ac.id

Abstrak

Banyak cara yang digunakan komputer untuk dapat melakukan *remote procedure call*, salah satunya adalah menggunakan XML-RPC yang merupakan protokol sederhana serta dapat mendukung distribusi pemrosesan berstandar internet. XML-RPC merupakan fenomena dalam sistem jaringan, spesifikasi dan implementasinya dapat membuat software bekerja pada sistem operasi yang berbeda untuk melakukan *procedure call* melalui internet atau intranet. Untuk dapat melakukan permintaan data dan transfer data pada sistem yang akan dibangun, dilakukan penelitian dengan mengimplementasikan XML-RPC pada database sistem informasi akademik yang merupakan bagian dari sistem informasi perguruan tinggi di Universitas Bina Darma yang memberikan informasi tentang mahasiswa yang aktif pada masa studi untuk dapat ditransfer ke dalam sistem informasi perpustakaan khususnya pada data keanggotaan. Penelitian ini dilakukan dengan cara mengimplementasikan XML-RPC dengan menggunakan database Sistem Informasi Akademik dan Sistem Informasi Perpustakaan.

Kata Kunci : Sinkronisasi, XML-RPC, Remote Procedure Call, Basis Data

1. PENDAHULUAN

Saat ini perangkat lunak sudah dapat tersebar skala dunia lewat internet, dimana terdapat jutaan pemakai yang potensial yang dapat terkoneksi antar pemakai dan aplikasi tersebut memungkinkan dalam melakukan interaksi dengan aplikasi yang telah ada yang berada di *platform* yang berbeda.

Salah satu fasilitas yang dimiliki oleh Universitas Bina Darma adalah perpustakaan. Dalam menjalankan aktivitasnya, perpustakaan dibantu dengan pemanfaatan teknologi informasi berupa sebuah sistem otomatisasi layanan, yang dapat membantu dalam proses pengolahan keanggotaan, pengolahan koleksi bahan pustaka serta proses sirkulasi bahan pustaka. Pendataan keanggotaan merupakan salah satu aktifitas yang dilakukan oleh staf pengolahan data perpustakaan. Pada aktivitas ini dilakukan proses *entry* data anggota ke dalam sistem sistem informasi perpustakaan, dimana data anggota didapat dari proses pendaftaran mahasiswa menjadi anggota.

Namun pada perkembangannya, pihak manajemen Universitas Bina Darma mengeluarkan kebijakan bahwa pada setiap periode penerimaan mahasiswa baru. Mahasiswa yang melakukan registrasi akan otomatis terdaftar menjadi anggota perpustakaan, sehingga perpustakaan perlu memasukkan semua data mahasiswa baru untuk diregistrasi kedalam sistem perpustakaan. Hal ini berdampak pada banyaknya data yang akan di *entry* , serta penanganan data dalam jumlah banyak.

Semenjak diberlakukan kebijakan tersebut, langkah yang diambil perpustakaan adalah berkoordinasi dengan pihak Unit Pelaksana Teknis (UPT) selaku pengelola sistem informasi akademik. Disini pihak perpustakaan bekerjasama dengan pihak UPT dalam penyediaan data mahasiswa yang selanjutnya menjadi bahan untuk melakukan *entry* data kedalam data Keanggotaan. Hal ini dilakukan karena adanya kesamaan beberapa elemen data antara data Mahasiswa di sistem Informasi Akademik dengan data Anggota di sistem informasi perpustakaan. Setelah terdaftar menjadi anggota perpustakaan barulah mahasiswa dapat menjadi anggota dan dapat melakukan transaksi peminjaman bahan pustaka.

Beberapa permasalahan yang muncul dari aktifitas yang dijalankan saat ini antara lain (1) Bagian pengolahan data perpustakaan melakukan pengisian data keanggotaan dalam jumlah yang banyak serta membutuhkan waktu yang tidak sedikit. (2) Dalam proses pendataan anggota perpustakaan, dilakukan dengan melakukan *entry* data yang berasal dari Sistem Informasi Akademik. (3) Tidak adanya sinkronisasi data antara Sistem Informasi Akademik dan Sistem Informasi Perpustakaan, hal ini sangat diperlukan karena adanya beberapa persamaan data yang digunakan pada kedua sistem tersebut.

Agar dapat saling berkomunikasi antara mesin dan sistem operasi yang berbeda diperlukan suatu *interface*. Salah satu sarana *interface* yang dipakai adalah dengan menggunakan *Remote Procedure Call* (RPC), dimana suatu *client* dapat mengakses program atau *web service* secara *remote* pada *server*. RPC merupakan suatu *request* dari *client* ke *server* aplikasi untuk melakukan operasi atau proses pertukaran data dan *server* akan mengirimkan kembali informasi yang diminta ke *client*.

Dengan diterapkannya proses sinkronisasi data ini diharapkan dapat bermanfaat yaitu membantu pihak perpustakaan dalam mempercepat proses pendataan keanggotaan perpustakaan melalui aplikasi yang dihasilkan dari penelitian ini khususnya pada proses sirkulasi dan pendataan pengunjung perpustakaan.

2. TINJAUAN PUSTAKA

2.1 Sinkronisasi Basis Data

Sinkronisasi antar basis data dimulai dengan memproses dokumen XML hasil representasi suatu basis data sumber kemudian membandingkan hasilnya dengan basis data tujuan. Terdapat dua buah dokumen yang akan diproses, yaitu dokumen yang berisi informasi skema basis data dan dokumen yang berisi data itu sendiri. (Riskadewi & Gede Karya, 2004)

Sinkronisasi yang dilakukan pada penelitian ini dilakukan terhadap sistem basis data yang digunakan. Sistem manajemen basis data adalah aplikasi perangkat lunak yang menyimpan struktur basis data, hubungan antardata dalam basis data, serta berbagai formulir dan laporan yang berkaitan dengan basis data. Basis data yang dikendalikan oleh sistem manajemen basis data adalah satu set catatan data yang berhubungan dan saling menjelaskan. (Simarmata, 2006)

2.2. Remote Procedure Call

Remote Procedure Call (RPC) merupakan suatu teknologi *message-passing programming* yang dikembangkan oleh Sun Microsystems dan secara luas dikembangkan oleh Open Software Foundation (OSF) yang mengizinkan melakukan remote aplikasi untuk mengeksekusi prosedur dan berinteraksi dengan servis yang ada pada jaringan komputer. RPC merupakan metode yang memungkinkan pemrosesan client/server pada platform Microsoft Windows 2000, Windows NT dan variannya maupun Linux. Implementasi pada Microsoft's menggunakan kekompatibelan RPC dengan implementasi-implementasi lain seperti pada IBM AIX, HP-UX, dan Sistem operasi Sun Solaris. (Johnston, 2001)

2.3. MiddleWare

Middleware adalah perangkat lunak yang berisi sekumpulan layanan yang memungkinkan aplikasi-aplikasi dan pemakai mempertukarkan informasi lewat jaringan-jaringan. Layanan-layanan ini berada di tengah (*Middle*) diatas sistem operasi dan perangkat lunak jaringan serta dibawah aplikasi tersebar. (Hariyanto, 2004 : 611). *Middleware* menyediakan banyak layanan. Layanan-layanan ini dikategorikan menjadi tiga bidang antara lain :

- a. Layanan komunikasi (pertukaran informasi) yang menangani komunikasi antara client dan server. Layanan ini menyediakan format, representasi pesan, layanan pesan (seperti marshaling, unmarshaling dan kompresi), pengikat pesan, komunikasi sinkron dan asinkron. Tidak semua layanan ini disediakan produk *middleware*, layanan ini berjalan pada satu protokol jaringan atau lebih seperti TCP/IP, IPX/SPX dan SNA.
- b. Layanan manajemen dan dukungan yang menyediakan layanan manajemen nama (pemetaan nama logi ke alamat fisik), keamanan (kendali dan kerahasiaan akses), penanganan kegagalan (timeout, deadlock), pengukuran dan monitoring kinerja, dan manajemen memori.
- c. Layanan spesifik aplikasi yang menyediakan layanan-layanan aplikasi seperti pengaksesan basis data SQL, pengolahan transaksi, dan layanan replikasi data. (Hariyanto, 2004 : 614)

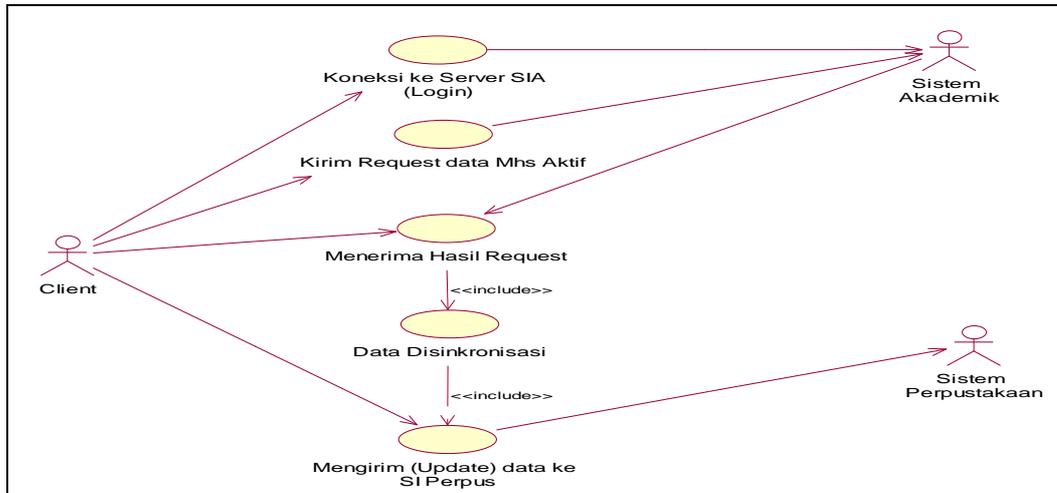
3. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah metode penelitian tindakan (*action research*). Metode ini merupakan metode penelitian yang bertujuan mengembangkan keterampilan-keterampilan baru atau cara pendekatan baru dan untuk memecahkan masalah dengan penerapan langsung di dunia kerja atau dunia aktual lainnya.

Analisis yang digunakan pada penelitian ini menggunakan analisis Analisis berorientasi objek, yaitu suatu tahapan dimana kita berusaha mengenali segenap permasalahan yang muncul pada pengguna dengan mendekomposisi use case diagram lebih lanjut, mengenali komponen sistem, objek-objek, hubungan antarobjek, dan sebagainya. (Sutopo, 2005)

Sistem yang dikembangkan penulis pada penelitian ini merupakan komunikasi antara beberapa objek yaitu : server sistem informasi akademik sebagai sumber data dengan client sebagai pengelola proses sinkronisasi serta terhadap sistem informasi perpustakaan sebagai penerima data hasil sinkronisasi. Dengan demikian dapat dideskripsikan aktor dan use case yang terlibat pada sistem ini sebagai berikut :Aktor, terdiri dari Client sebagai pengelola sistem sinkronisasi, Sistem Informasi Akademik sebagai sumber penyedia data Sistem Informasi Perpustakaan sebagai sistem yang dikoneksikan dengan sistem akademik dan penerima data hasil sinkronisasi. Use Case, terdiri dari Koneksi ke Server SIA (Login), Kirim Request data Mahasiswa Aktif, Menerima Hasil Request, Sinkronisasi Data, Mengirim (Update) data ke Sistem Perpustakaan

Adapun use case diagram untuk sinkronisasi data user antara sistem informasi akademik dengan sistem informasi perpustakaan dapat dilihat pada gambar berikut ini

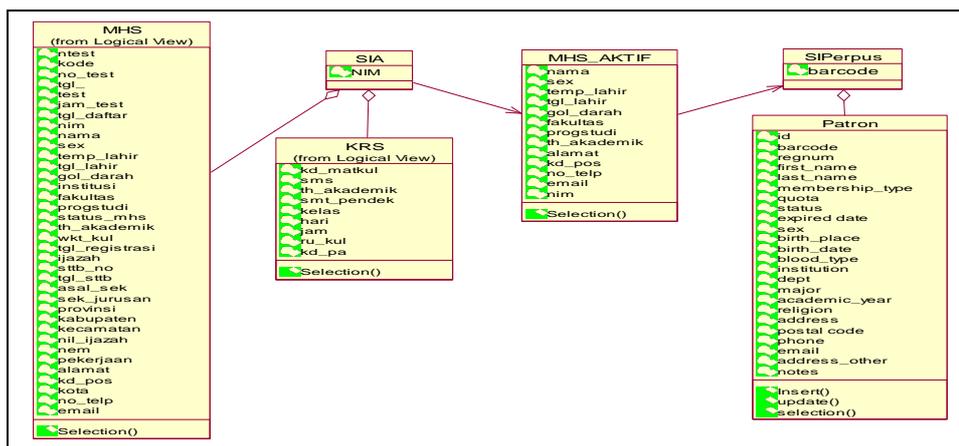


Gambar 1. Use case diagram

2.2 Perancangan

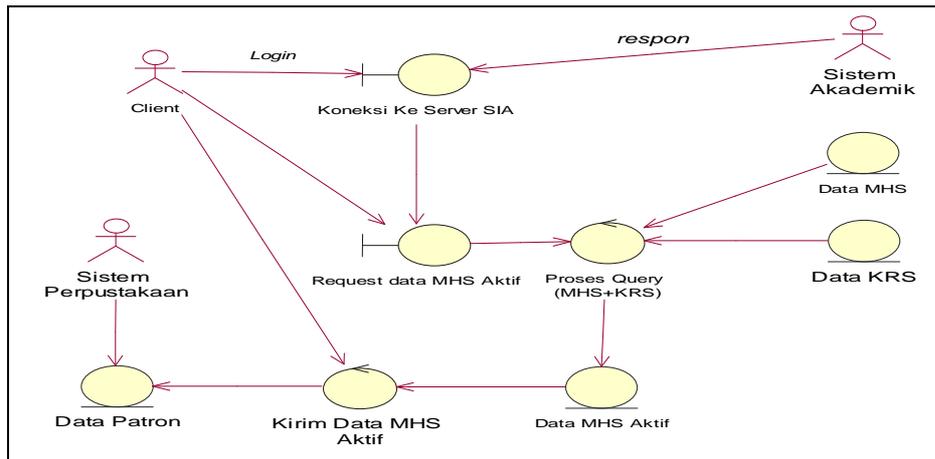
Setelah mengidentifikasi aktor dan use case diatas, analisis mulai memfokuskan pada pemodelan struktur antar kelas-kelas. Struktur merupakan ekspresi dari kompleksitas domain masalah, berkaitan dengan tanggung jawab sistem.

Dari diagram use case dan skenario yang dibuat sebelumnya, maka dapat diketahui terdapat beberapa kelas yang terlibat pada sistem ini yang terdiri dari kelas-kelas sebagai berikut : Kelas Sistem Informasi Akademik (SIA), Kelas Mahasiswa, yang merupakan pewarisan dari kelas SIA, Kelas KRS, yang merupakan pewarisan dari kelas SIA, Kelas Mahasiswa aktif (MHS_AKTIF), Kelas Sistem Informasi Perpustakaan (SIPerpus), Kelas Patron, yang merupakan pewarisan dari kelas SIPerpus. Dari kelas-kelas yang terbentuk pada analisis diatas selanjutnya masing-masing kelas akan berelasi terhadap kelas yang lain. Relasi ini adalah koneksi semantik antar kelas yang memungkinkan suatu kelas mengetahui atribut-atribut, operasi-operasi serta relasi-relasi yang dimiliki kelas yang lainnya. Dengan tujuan saling mengirim pesan (message) antar kelas. Jika digambarkan, relasi antar kelas yang terbentuk pada sistem sinkronisasi ini sebagai berikut :



Gambar 2. Class Diagram

Dari gambar diatas terdapat enam kelas, dimana masing-masing kelas berelasi dengan kelas lainnya. Kelas MHS dan KRS berelasi terhadap kelas SIA, dimana keduanya merupakan agregasi dari kelas SIA. Begitu juga dengan kelas patron yang merupakan agregasi dari kelas SIPerpus. Sedangkan kelas MHS_AKTIF berelasi dengan kelas SIA dimana kelas ini memberikan pesan berupa informasi data mahasiswa aktif dan kelas MHS_AKTIF akan mengirimkan pesan kepada kelas SIA berupa data mahasiswa aktif untuk selanjutnya dijadikan sebagai data anggota perpustakaan pada kelas Patron



Gambar 3. Collaboration Diagram

Pada proses perancangan basis data, terdapat beberapa tabel yang diperlukan untuk melakukan sinkronisasi data user. Tabel tersebut meliputi tabel mahasiswa dan tabel KRS yang berasal dari database akademik, tabel mhs_aktif yang digunakan untuk menampung data sementara hasil request data pada server akademik, dan tabel patron yang berasal dari database perpustakaan.

Berikut ini merupakan struktur tabel-tabel yang berkaitan dengan sistem ini, beberapa tabel tersebut untuk tabel mahasiswa dan krs sudah mengalami penyederhanaan dimana struktur yang disebutkan merupakan item data yang diperlukan dalam proses sinkronisasi.

Tabel 1. Struktur tabel Mahasiswa

| No | Field | Type | Size |
|----|------------|---------|------|
| 1 | nim | varchar | 13 |
| 2 | nama | varchar | 30 |
| 3 | sex | Enum | |
| 4 | temp_lahir | varchar | 25 |
| 5 | Tgl_lahir | date | |
| 6 | gol_darah | enum | |
| 7 | institusi | varchar | 6 |
| 8 | fakultas | char | 1 |
| 9 | Progstudi | Char | 3 |
| 10 | alamat | varchar | 250 |
| 11 | kd_pos | varchar | 5 |
| 12 | kota | varchar | 25 |
| 13 | no_telp | varchar | 12 |
| 14 | email | varchar | 25 |

Tabel 2. Struktur tabel KRS

| No | Field | Type | Size |
|----|-------------|------|------|
| 1 | Nim | Char | 13 |
| 3 | Sms | Char | 1 |
| 4 | Th_akademik | Char | 9 |

Tabel 3. Struktur tabel Mhs_Aktif

| No | Field | Type | Size |
|----|------------|---------|------|
| 1 | nim | varchar | 13 |
| 2 | nama | varchar | 30 |
| 3 | sex | Enum | |
| 4 | temp_lahir | varchar | 25 |
| 5 | tgl_lahir | date | |
| 6 | gol_darah | enum | |

| | | | |
|----|-----------|---------|-----|
| 7 | institusi | varchar | 6 |
| 8 | fakultas | char | 1 |
| 9 | progstudi | Char | 3 |
| 10 | alamat | varchar | 250 |
| 11 | kd_pos | varchar | 5 |
| 12 | kota | varchar | 25 |
| 13 | no_telp | varchar | 12 |
| 14 | email | varchar | 25 |

Tabel 4. Struktur tabel Patron

| No | Field | Type | Size |
|----|-----------------|---------|------|
| 1 | Id | int | 11 |
| 2 | Barcode | varchar | 13 |
| 3 | Reg_num | varchar | 14 |
| 4 | First_name | varchar | 50 |
| 5 | Last_name | varchar | 50 |
| 6 | Membership_type | TinyInt | 4 |
| 7 | Quota | TinyInt | 4 |
| 8 | Status | Varchar | 15 |
| 9 | Expired_date | Date | |
| 10 | Sex | Char | 1 |
| 11 | Birth_place | Varchar | 25 |
| 12 | Birth_date | Date | |
| 13 | Blood_type | Char | 2 |
| 14 | Institution | Char | 1 |
| 15 | Dept | Char | 3 |
| 16 | Major | Char | 3 |
| 17 | Academic_year | Varchar | 9 |
| 18 | Religion | Char | 1 |
| 19 | Address | Varchar | 255 |
| 20 | Postal Code | Varchar | 5 |
| 21 | Phone | Varchar | 12 |
| 22 | Email | Varchar | 50 |
| 23 | Address_other | Text | |
| 24 | Notes | Text | |

4. HASIL DAN PEMBAHASAN

Hasil penelitian ini berupa sebuah percobaan penggunaan XML-RPC pada sistem client server, untuk membuktikan salah satu keunggulan yang dimiliki oleh XML-RPC yaitu dapat diterapkan pada pendistribusian client-server dengan mudah, dikarenakan penggunaan HTTP sebagai transportasi data dan XML sebagai encodingnya, sehingga sistem tidak tergantung pada koneksi yang terus menerus antara client dan server. Client tidak perlu mengetahui di server mana method yang melayani permintaan tersebut berada, karena semua permintaan layanan method tersebut diatur oleh web service. Client hanya perlu mengetahui alamat penyedia layanan web service. Hal ini menunjukkan kemampuan interoperabilitas yang dimiliki XML-RPC tinggi dan dapat dilakukan pendistribusian dengan baik.

Sinkronisasi data dilakukan dengan menghubungkan dua aplikasi server web service, yaitu server akademik dan server perpustakaan. Langkah awal dalam proses tersebut dilakukan sesuai dengan permintaan layanan yang dilakukan oleh client yang dikirim melalui http dalam bentuk html (PHP) yang selanjutnya akan dikelola oleh web server (client web server). Selanjutnya permintaan akan dikirimkan ke server akademik melalui http dalam format XML-RPC (PHP).

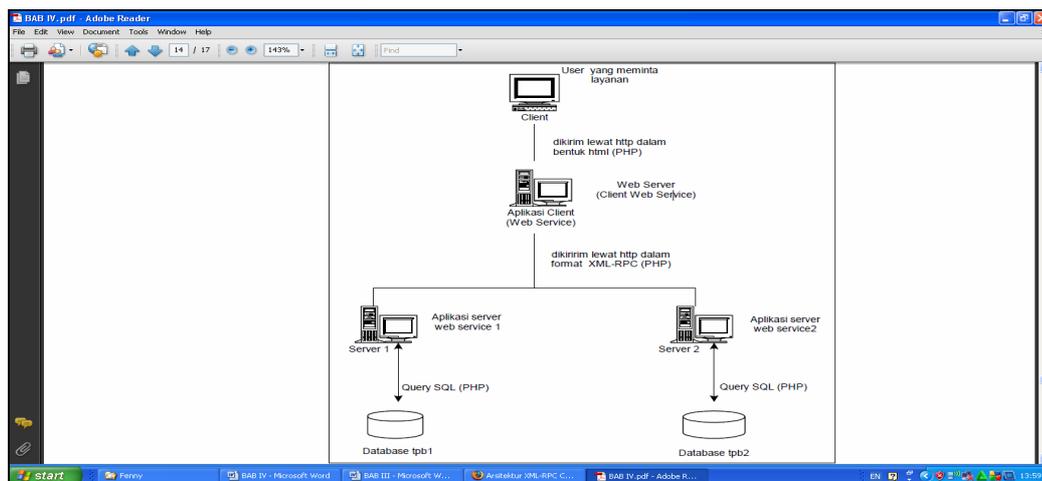
Dari pesan permintaan layanan tersebut selanjutnya server akademik akan menindaklanjuti permintaan dengan mengambil data dari database melalui perintah query yang selanjutnya akan dikirimkan ke aplikasi web service (middleware). Kemudian aplikasi ini akan mengirimkan data tersebut ke server perpustakaan melalui perintah query.

Dalam penelitian ini juga dilakukan percobaan penggunaan XML-RPC pada sistem *client server*, untuk membuktikan salah satu keunggulan yang dimiliki oleh XML-RPC yaitu dapat diterapkan pada pendistribusian client-server dengan mudah, dikarenakan penggunaan HTTP sebagai transportasi data dan XML

sebagai *encodingnya*, sehingga sistem tidak tergantung pada koneksi yang terus menerus antara client dan server. Client tidak perlu mengetahui di *server* mana method yang melayani permintaan tersebut berada, karena semua permintaan layanan method tersebut diatur oleh *web service*. Client hanya perlu mengetahui alamat penyedia layanan *web service*. Hal ini menunjukkan kemampuan interoperabilitas yang dimiliki XML-RPC tinggi dan dapat dilakukan pendistribusian dengan baik.

Untuk semua transaksi permintaan client ke server dan respon dari server ke client proses kerjanya sama, yaitu setiap selesai satu transaksi akan selalu kembali ke aplikasi client. Karena itu analisis dan hasil pada penelitian ini tidak dilakukan pada keseluruhan method yang ada, tetapi akan dilakukan pada beberapa proses yang mewakilinya.

Saat client meminta atau mengambil data dari server maka diperlukan inisialisasi objek `xmlrpc_client` ke XML-RPC server yang akan digunakan. Untuk menganalisis hasil pembacaan data dari server ke client dapat dilihat dari dua sisi, yaitu sisi client dan sisi server. Adapun gambaran arsitektur dari sistem sinkronisasi data ini sebagai berikut :



Gambar 4. Arsitektur Sinkronisasi Data

Untuk dapat masuk ke dalam sistem, user harus melakukan login dan memasukkan password. Form userid yang dibuat dengan HTML akan dihubungkan ke aplikasi *client* yang kemudian akan mengirimkan *message* ke *server* apakah *userid* dan *password* tersebut sesuai. Di bawah ini adalah contoh message client ke server untuk user login.

```
if ($userid!="" && $passwd!="") { $msg=new xmlrpcmsg('server.login',array(new xmlrpcval($userid,"string"), new xmlrpcval($passwd,"string")));
```

Untuk semua transaksi permintaan client ke server dan respon dari server ke client proses kerjanya sama, yaitu setiap selesai satu transaksi akan selalu kembali ke aplikasi client. Karena itu analisis dan hasil pada penelitian ini tidak dilakukan pada keseluruhan method yang ada, tetapi akan dilakukan pada beberapa proses yang mewakilinya.

Saat client meminta atau mengambil data dari server maka diperlukan inisialisasi objek `xmlrpc_client` ke XML-RPC server yang akan digunakan. Untuk menganalisis hasil pembacaan data dari server ke client dapat dilihat dari dua sisi, yaitu sisi client dan sisi server.

Untuk dapat koneksi client harus mengirimkan requestnya ke server dengan mengirimkan message, `xmlrpc_client` harus dibentuk untuk melakukan koneksi (`$client`) ke server seperti contoh berikut :

```
$msg=new xmlrpcmsg('server.carimhsaktifnim', array(new xmlrpcval($nim, "string")));  
$client=new xmlrpc_client('/sinkronisasi/server1.php', $Lokasi, 80);
```

Pada contoh di atas nama method `xmlrpcmsg` yang dikirimkan ke server adalah "server.cariMHSAKTIFnim" yang digunakan untuk mencari `$nim` dengan tipe data string pada database. Nama method tersebut harus sesuai dengan method yang ada pada server. Sedangkan objek `xmlrpcval` berfungsi sebagai tempat semua member dari parameter array misalnya `$Valnim` yang menunjukkan nilai (*value*) yang diberikan dengan tipe string. Data yang baca dikirimkan oleh *server* ke *client* objek juga dalam bentuk format XML. Saat data diterima oleh client objek dalam maka data tersebut harus diubah kembali kedalam bentuk semula. Dimana bentuk format XML harus diubah menjadi *value* dengan menggunakan `xmlrpcval` dan dimasukkan ke dalam form yang telah disediakan pada *client*.

Setelah proses sinkronisasi data berhasil selanjutnya data yang diperoleh akan dipergunakan pada beberapa modul aplikasi yang ada pada sistem informasi perpustakaan. Adapun beberapa modul tersebut terdiri dari proses registrasi anggota perpustakaan, proses pendataan pengunjung perpustakaan, proses sirkulasi atau peminjaman dan pengembalian buku serta proses pengurusan surat keterangan bebas pustaka bagi mahasiswa yang akan mengikuti yudisium dan wisuda dimana modul tersebut saat ini sedang dikembangkan oleh staf perpustakaan.

5. KESIMPULAN

Berdasarkan data serta pembahasan hasil yang dilakukan dalam penelitian ini, berikut ini dapat disimpulkan beberapa hal yaitu :

- a. Penelitian ini menghasilkan sebuah aplikasi sinkronisasi basis data antara sistem informasi akademik dan sistem informasi perpustakaan yang dikembangkan dengan bahasa pemrograman PHP, DBMS MySQL serta menggunakan teknologi *middleware XML Remote Procedure Call*
- b. Aplikasi yang dihasilkan mampu melakukan sinkronisasi data sehingga dapat mengatur transformasi data dari kedua sistem yang berbeda dengan menggunakan format XML saat data dikirimkan dan saat diterima data tersebut ditrasformasikan kembali ke bentuk semula, tetapi hal ini tidak signifikan dengan keluwesan dan kemampuan yang dimiliki XML-RPC.
- c. Implementasi aplikasi menggunakan XML-RPC pada sistem *client server*, dapat dibuat menjadi suatu *web service* yang melayani permintaan client ke server pada sistem yang berbeda serta dapat saling terkoneksi dan menukarkan datanya dengan mudah.

DAFTAR PUSTAKA

- Hariyanto, Bambang. 2004. *Sistem Manajemen Basis Data*. Informatika : Bandung
- Johnston, J., 2001., Using XML-RPC, [http://www.oreilly.com/catalog /progxmlrpc](http://www.oreilly.com/catalog/progxmlrpc). Diakses tanggal 25 Oktober 2010.
- Riskadewi., Gede Karya. 2004. Representasi dan Sinkronisasi Basis Data Relasional Dengan Dokumen XML. *Jurnal Integral*. Vol. 9 No. 1, Maret 2004 : Bandung
- Simarmata, Janner, Iman Paryudi., 2006. *Basis Data*. Yogyakarta : Andi Offset
- Sutopo, Ariesto, Hadi, 2005. *Analisis & Desain Berorientasi Objek*. Jakarta : J&J Learning