Spektra Virtual Library berbasis Web Services

Yendry Ferdinand Tandoro[1], Justinus Andjarwirawan[2], Alexander Setiawan[3]
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236 Telp. (031) – 2983455, Fax. (031) – 8417658
E-mail: yendryft@gmail.com, justin@petra.ac.id, alexander@petra.ac.id

ABSTRAK :

Software New SPEKTRA digunakan oleh banyak perpustakaan. Peralihan dari *New SPEKTRA* menjadi *iSPEKTRA* yang berbasis *online* memberikan peluang dibuatnya *unified database* yang menyimpan data dari semua *client iSPEKTRA* dengan nama *Spektra Virtual Library (SVL)*.

Teknologi *Web Services* berbasis *REST* digunakan untuk sinkronisasi data agar seragam dan memudahkan publik dalam mencari koleksi karya tulis baik fisik maupun digital. *Twitter bootstrap* digunakan untuk *website interface* dan *PostgreSQL* digunakan sebagai *database*.

Spektra Virtual Library akan menjembatani perbedaan struktur data antar pengguna *software iSPEKTRA* dan membantu publik mencari koleksi karya tulis berbagai perpustakaan dalam *website* tunggal.

Penelitian ini dilakukan untuk membuat *web services* sebagai sarana sinkronisasi database dan penyetaraan data, serta website berbasis twitter bootstrap untuk *Spektra Virtual Library*. Hasil dari penelitian ini berupa *website Spektra Virtual Library* dan program *web services*.

Kata kunci : *php, postgresql, ispektra, web services.*

ABSTRACT :

New SPEKTRA software used by many libraries. The shifting of New SPEKTRA to iSPEKTRA that is online based creates opportunity for the creation of unified database that saves data from all iSPEKTRA's client.

Web Services technology based on *REST* used to syncronize data to ensure uniformity and make user easier to find papers such as physical collection and digital collection. Twitter bootstrap used for website interface and PostgreSQL used for database.

Spektra Virtual Library will fill the gap of data structure difference between iSPEKTRA users and help public to search for book collections in many libraries on single website.

This study is accomplished to make web services as a tool to synchronize database and data equalization for Spektra Virtual Library. Result of this study are Spektra Virtual Library Website and web services software.

Keyword: php, postgresql, ispektra, web services.

1. PENDAHULUAN

iSPEKTRA yang merupakan singkatan dari Sistem Informasi Perpustakaan Universitas Kristen Petra adalah sebuah *software* terpadu yang berfungsi untuk mengelola data dari sebuah perpustakaan. *Software* ini memudahkan *user* sebagai pengguna fasilitas perpustakaan untuk mendapatkan *update* informasi terbaru melalui internet mengenai status sebuah buku yang dibutuhkan, apakah buku tersebut sedang dipinjam atau *available* untuk dipinjam.

Saat ini *software New SPEKTRA* telah digunakan oleh lebih dari 40 institusi lain yang ingin mengelola *database* buku, jurnal atau karya tulis lainnya. Nantinya seluruh pengguna *New SPEKTRA* akan beralih dan menggunakan *software iSPEKTRA*. Peralihan dari *New SPEKTRA* ke *iSPEKTRA* memberikan kesempatan bagi untuk dapat menggabungkan *database* dari semua institusi yang menggunakan *iSPEKTRA* sehingga nantinya hasil dari pencarian tidak hanya sebatas pada *database* perpustakaan Universitas Kristen Petra saja namun juga mampu mencakup *database* dari institusi lainnya.

Didalam penelitian ini dirancang dan dibuat suatu sistem terintegrasi bernama *Spektra Virtual Library (SVL)* dimana dihasilkan suatu *database* terpusat yang menyimpan data dari database semua pengguna *iSPEKTRA* dan sistem sinkronisasi data untuk menghasilkan *database* tersebut. *User* yang mengakses *database* ini dapat melakukan pencarian buku dan dapat mengunduh *e-book* dari buku tersebut apabila tersedia, selain itu *user* juga dapat melihat status dari buku tersebut apakah sedang dipinjam atau dapat dipinjam. Adanya *SVL* diharapkan dapat membantu mahasiswa atau kalangan lainnya untuk melakukan pencarian buku dalam lingkup yang lebih luas dan tidak hanya pada *database* perpustakaan tertentu saja, namun dalam lingkup semua *database* pengguna *iSPEKTRA*. Kenyataannya seringkali sebagai mahasiswa diharuskan mencari buku di perpustakaan sebagai sumber referensi dalam mengerjakan tugas atau sekedar menambah ilmu. Namun tidak semua buku yang dibutuhkan dapat ditemukan di perpustakaan universitas, karena jumlah buku baru yang muncul setiap harinya tidaklah sebanding dengan kemampuan sebuah perpustakaan untuk memperbarui koleksi mereka. Keterbatasan koleksi dari sebuah perpustakaan membuat kadang kala harus melakukan pencarian di perpustakaan lainnya. Hal inilah yang menjadi salah satu alasan mengapa diperlukan aplikasi semacam *SVL*.

Spektra Virtual Library terdiri dari 4 buah aplikasi yaitu *service provider*, *service requester*, database serta *website SVL* yang dapat diakses oleh publik umum. Aplikasi ini dibangun menggunakan database PostgreSQL, s*ervice requester* berbasis *PHP* dan *service provider* berbasis *REST* sedangkan *user* dapat mengakses *SVL* melalui *website* berbasis *PHP*.

2. TINJAUAN PUSTAKA

2.1 *REST*

REST adalah sebuah gaya arsitektur ketika digunakan dalam aplikasi *HTTP* yang memanfaatkan fitur yang ada pada *HTTP* (*URI*, kode respon, dan permintaan-metode *GET*, *POST*, *PUT*, dan *DELETE*) untuk bekerja pada pengguna *API* yang coba untuk dilakukan.[1]

Sebagian besar situs saat ini telah beralih menggunakan teknologi *REST*. Seperti yang telah digunakan oleh *Twitter* saat ini dimana pengembang situs jejaring sosial terbesar ini mempercayakan teknologi *REST* sebagai aplikasi *web service* dalam melayani permintaan pengguna yang cukup besar setiap harinya. Permintaan *RESTful API* menggunakan *HTTP* untuk menggambarkan apa yang akan dilakukan oleh *user* (pengguna). *API* yang diterima oleh *user* merupakan *URI* khusus yang dapat digunakan oleh *user* yang telah mendapatkannya. *REST* mempersatukan teori tentang bagaimana "*distributed hypermedia system*" (terutama *World Wide Web*) diorganisir dan distruktur dengan sebaik mungkin. *REST* merupakan cara baru berpikir tentang arsitektur jaringan berdasarkan pengamatan atas bagian jaringan kerja.

Aplikasi *client* atau *server* dengan dukungan *HTTP* dapat dengan mudah memanggil *service* tersebut dengan command *HTTP GET*. Berdasar pada bagaimana cara *service provider* menulis *script*, hasil respon *HTTP* akan menjadi lebih simpel seperti beberapa *header* standar dan *string* teks yang mengandung *value* terkini untuk *input* yang diberikan. Metode ini mempunyai keuntungan signifikan dibanding *service* berbasis *SOAP*. *Developer* dapat mengetahui bagaimana untuk membuat dan memodifikasi sebuah *URI* untuk mengakses *resource* yang berbeda. *SOAP* disisi lain membutuhkan pengetahuan khusus untuk spesifikasi *XML* dan kebanyakan *developer* akan membutuhkan *SOAP toolkit* untuk membentuk *request* dan menguraikan (*parsing*) hasilnya.[2]

REST merupakan penyederhanaan dari *HTTP*. Dengan pertumbuhan *web* yang semakin popular, banyak keputusan desain asli yang memandu *HTTP* diabaikan. Para pengembang aplikasi *web* cenderung melihat hal-hal seperti *verb HTTP* dan kode status respon sebagai sesuatu yang *incidental* untuk aplikasi, atau sebagai suatu hal tidak penting yang akan ditangani jika waktu masih mengijinkan. Penggunaan *HTTP* sebagaimana yang diharapkan, sering terlihat sebagai sesuatu yang tidak diperlukan atau menyulitkan. Namun, dalam beberapa tahun belakangan ini, dengan hadirnya kembali prinsip-prinsip *REST* telah mengindikasikan bahwasannya *HTTP* telah lebih dari cukup baik di atas segalannya.[3]

2.2 Web Services

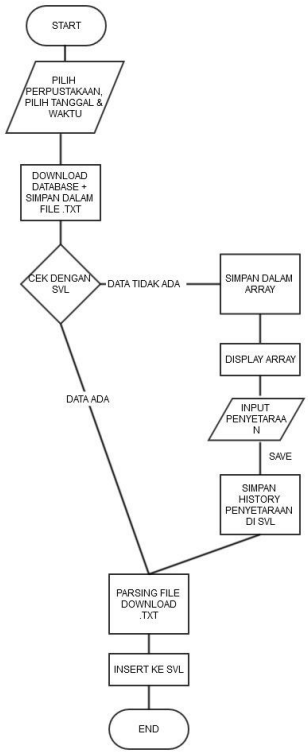
Web services adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web services* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan *web services*. *Web services* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun *compiler*.[4]

Web services bertujuan untuk meningkatkan kolaborasi antar *programmer* dan perusahaan, yang memungkinkan sebuah fungsi di dalam *web services* dapat digunakan oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat didalamnya.[5]

3. ANALISA DAN DESAIN SISTEM

3.1 Website SVL

Melalui *website SVL*, *admin* dapat mengakses *admin page* dan diharuskan melakukan *login* terlebih dahulu. Apabila *admin* lupa *password* miliknya dapat menggunakan fitur lupa *password* yang secara otomatis mengirimkan *password* melalui *email*. Setelah melakukan *login*, *admin* dapat mengakses fitur-fitur khusus admin seperti *add library*,*edit library*,*delete library*, *update database*, *view library information* dan *add admin*. Berbeda dengan *admin*, *user* hanya dapat melakukan search koleksi saja.



Gambar 1. *Flowchart update database*

Spektra Virtual Library atau *SVL* memuat koleksi dari banyak perpustakaan dan koleksi tersebut terus bertambah, karenanya dibutuhkan fitur update database agar database *SVL* selalu terbarukan. Proses *update* database dapat dilakukan melalui halaman *Admin* di *website SVL*. *Admin* dapat memilih perpustakaan mana yang ingin di-*download* databasenya atau terdapat pilihan lain yaitu memilih semua perpustakaan sekaligus. Database yang telah di-*download* akan disimpan dalam bentuk file .txt, kemudian akan dicek apakah dibutuhkan penyetaraan sebelum dapat disimpan ke dalam database *SVL*, apabila perlu maka data-data yang membutuhkan penyetaraan akan disimpan dalam *array* dan akan ditampilkan kepada *admin* untuk kemudian dapat diproses. *History* penyetaraan akan disimpan dalam tabel _history di *SVL* sehingga di kemudian hari tidak dibutuhkan penyetaraan untuk data yang sama. Setelah penyetaraan disimpan maka akan dilakukan *parsing* isi file .txt untuk dimasukkan ke dalam database *SVL*, sebelum dilakukan proses *insert*, data yang masuk akan diubah sesuai penyetaraan yang telah dilakukan sebelumnya.

Web services digunakan untuk melakukan proses update database, dengan detail sebagai berikut :

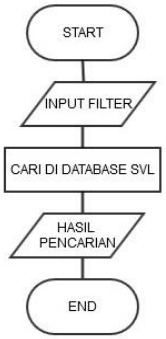
- Parameter : tanggal awal(YYYY-MM-DD) dan tanggal akhir(YYYY-MM-DD)
- Return value berupa array of data dengan isi data dari tabel berikut :
    - Ft_digitalfile
    - Ft_main
    - Lib_judul
    - Lib_buku
    - Ft_m_relationship

Design sistem update database dirincikan sebagai berikut :

- Pilih server client yang ingin diambil datanya
- Looping for sejumlah server yang dipilih (semua atau satu)
    - Request update ke server client berdasar tanggal awal dan tanggal akhir
    - Convert return data dari format json menjadi array
    - Simpan dalam file text dengan nama file_i.txt (i = angka berdasarkan looping for jumlah server)
    - Parsing *data array*, cek apa ada penyetaraan
        - Jika ada penyetaraan tampilkan data penyetaraan
        - Jika tidak ada maka data langsung disimpan dan masuk dalam *database SVL*

Gambar 2. *Flowchart search physical collection*

Salah satu fitur utama *website SVL* adalah pencarian koleksi karya tulis baik berupa koleksi fisik semisal buku maupun koleksi digital semisal file .pdf. *User* dapat menggunakan fitur filter untuk mendapatkan hasil pencarian yang lebih akurat dan sesuai kebutuhan. Variabel yang dapat dimasukkan antara lain : perpustakaan pemilik koleksi, tipe dari media, bahasa, tipe koleksi dan tahun. Koleksi fisik tentunya dapat dipinjam oleh anggota perpustakaan pemilik koleksi yang bersangkutan, dengan demikian dibuatlah fitur pengecekan status buku untuk mengetahui apakah koleksi tersebut sedang dipinjam atau tersedia. Pengecekan ini dilakukan dengan membandingkan hasil pencarian dengan salah satu tabel di database *server client SVL* yang menyimpan sejarah peminjaman buku. Apabila koleksi hasil pencarian dideteksi terdapat di tabel tersebut maka dapat dipastikan bahwa koleksi tersebut sedang dipinjam.



Gambar 3. *Flowchart search digital collection*

Koleksi digital merupakan koleksi non fisik atau file digital seperti file .pdf, video cd, dsb. Sama halnya seperti *physical collection*, proses pencarian *digital collection* pada *website SVL* juga tidak berbeda jauh. Perbedaan hanya terletak pada variabel pencarian yang dapat digunakan *user* pada fitur filter yaitu : perpustakaan pemilik koleksi, kategori, sub kategori, bahasa, format, tipe dan tahun. Koleksi digital tidak dapat dipinjam karena itu tidak ada status ketersediaan peminjaman atau tidak, dengan demikian tidak dibutuhkan fitur pengecekan status ketersediaan koleksi seperti halnya physical collection.

3.2 Database SVL

Sebuah perpustakaan tidak lepas dari adanya sebuah *database* yang digunakan untuk menyimpan data buku atau karya tulis yang merupakan koleksi perpustakaan tersebut. *Database SVL* berbasis pada *database* yang digunakan *iSPEKTRA* dengan beberapa penambahan dan pengurangan. Penambahan itu diantara lain berupa tabel untuk menyimpan *history* penyetaraan serta penambahan *primary key* untuk setiap tabel. Pengurangan yang dilakukan berupa penghapusan *constraint* agar memudahkan proses *update database* serta penghapusan beberapa tabel yang dinilai tidak diperlukan karena tidak menyimpan data yang dibutuhkan oleh *user* atau *admin*.

Setelah dilakukan analisa terhadap kebutuhan *SVL* maka dilakukan penghapusan 80 tabel dari semula 121 tabel menjadi 41 tabel, penambahan 9 tabel baru dan penghapusan sejumlah kolom pada 43 tabel tersebut. Tabel pada database *iSPEKTRA* yang dihapus dengan alasan tidak digunakan adalah sebagai berikut :

9 Tabel tambahan pada *database SVL* selain tabel dari *database iSPEKTRA* adalah sebagai berikut :

1. Ft_m_category_history
   Menyimpan *record* penyetaraan kategori koleksi digital dari proses *update database*. Terdiri dari 3 kolom yaitu fctype_category (PK), fckd_library (PK), fnkd_category.
2. Ft_m_format_history
   Menyimpan *record* penyetaraan format koleksi digital dari proses *update database*. Terdiri dari 3 kolom yaitu fcmime (PK), fckd_library (PK), fnkd_format.
3. Ft_m_relationship_history
   Menyimpan *record* penyetaraan *relationship* koleksi digital dari proses *update database*. Terdiri dari 3 kolom yaitu fcdescription (PK), fckd_library (PK), fnkd_reltype.
4. Ft_m_subcategory_history

Menyimpan *record* penyetaraan sub kategori koleksi digital dari proses *update database*. Terdiri dari 3 kolom yaitu fctype_subcategory (PK), fckd_library (PK), fnkd_subcategory.

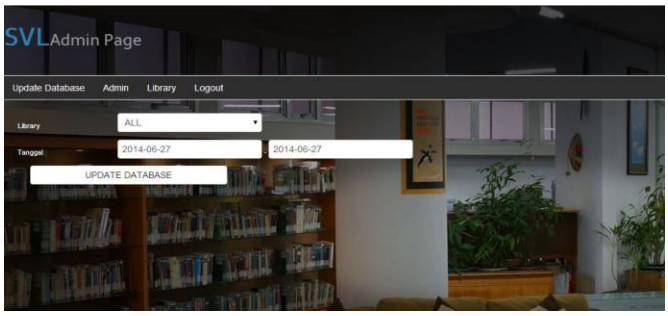
5. Ft_m_type_history
   Menyimpan *record* penyetaraan tipe koleksi digital dari proses *update database*. Terdiri dari 3 kolom yaitu fctype (PK), fckd_library (PK), fnkd_type.
6. Lib_ada_m_jnstbt_history
   Menyimpan *record* penyetaraan jenis terbitan koleksi fisik dari proses *update database*. Terdiri dari 3 kolom yaitu fcket_jnstbt (PK), fckd_library (PK), fnkd_jnstbt.
7. Lib_m_jenisav_history
   Menyimpan *record* penyetaraan jenis audio visual koleksi fisik dari proses *update database*. Terdiri dari 3 kolom yaitu k245h (PK), fckd_library (PK), fckd_jnsav.
8. Lib_m_jnskol_history
   Menyimpan *record* penyetaraan jenis koleksi fisik dari proses *update database*. Terdiri dari 3 kolom yaitu k099l (PK), fckd_library (PK), fckd_jnskol.
9. M_admin
   Menyimpan data *admin SVL*. Terdiri dari 5 kolom yaitu *username* (PK), *email*, jenisuser, *password*, *secret*. Kolom *password* dan *secret* di tabel ini dienkripsi menggunakan enkripsi MD5. Kolom *secret* digunakan pada fitur *forget password* halaman *login admin page*.

4. HASIL

Hasil aplikasi berupa website dengan domain http://ta30.petra.ac.id/SVL/boostrap.

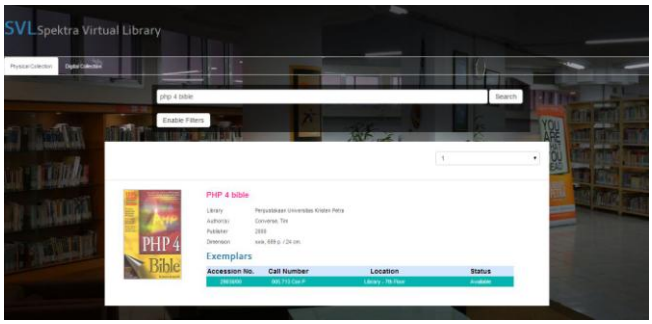
.**Gambar 4. *Admin homepage***

Website Spektra Virtual Library (SVL) memiliki *admin page* dimana memiliki 3 fitur utama yaitu *update database SVL*, *manage library* dan *manage admin*. Sebelum dapat menggunakan fitur-fitur tersebut, admin diharuskan melakukan login terlebih dahulu. Proses login ini membutuhkan input username dan password dengan fitur tambahan yaitu forget password yang dapat digunakan oleh admin apabila tidak dapat mengingat password miliknya.



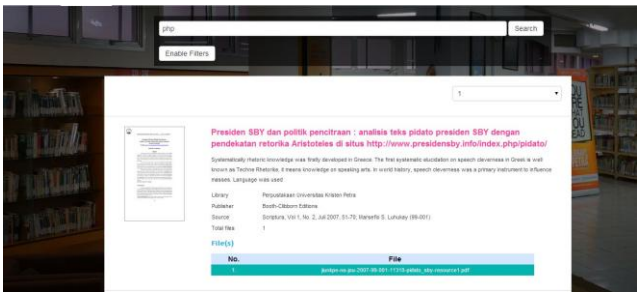
Gambar 5. Halaman *update database*

Update database dapat dilakukan setelah *admin* memilih perpustakaan mana yang ingin diperbarui *database*-nya. *Admin* dapat memilih salah satu perpustakaan saja atau semua sekaligus. Selain itu *admin* juga harus memasukkan tanggal update dimana inputan ini akan digunakan sebagai parameter *update* untuk membatasi data mana saja yang perlu dimasukkan. Inputan tanggal tersebut akan dicocokkan dengan *field fdtgl_update* di tabel *database* perpustakaan yang bersangkutan. Apabila tanggal di *field fdtgl_update* berada di *range* inputan *admin* maka data tersebut akan diunduh untuk kemudian dimasukkan ke dalam *database SVL*. Proses *input* tanggal pada halaman ini juga dibantu oleh *addon jquery datepicker*.



Gambar 6. Halaman pencarian koleksi fisik

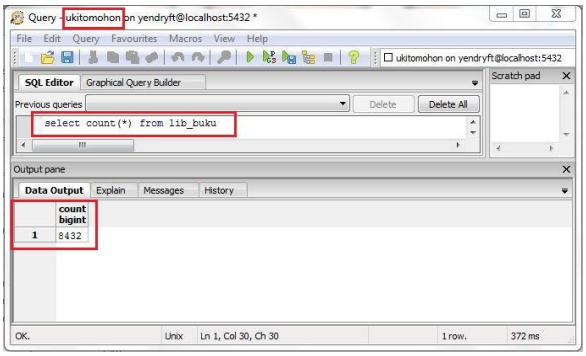
Terlihat hasil pencarian physical collection pada Gambar 14. menggunakan keyword "PHP 4 Bible". *Keyword* tersebut dipilih dengan tujuan untuk menunjukkan fungsi *show image cover* karena banyak koleksi lain yang belum memiliki *cover page*. *Image cover page* ini diambil dari *server client SVL*.". Informasi lain yang ditampilkan adalah data *Library* (perpustakaan pemilik koleksi), *Author* (pengarang karya tulis), *Publisher* (penerbit karya tulis), *Dimension* (jumlah halaman dan panjang buku). Informasi tambahan dibagian bawah berupa *Accession No* (nomor kode akses), *Call Number* (nomor kode panggil), *Location* (lokasi spesifik karya tulis), *Status* (status buku apakah tersedia atau sedang dipinjam).



Gambar 7. Halaman pencarian *digital collection*

Dilakukan pengujian waktu transfer data dengan kondisi data tanpa ada penyetaraan dan ada penyetaraan. *Database* yang digunakan adalah *database* perpustakaan Universitas Kristen Tomohon yang merupakan salah satu client *iSPEKTRA.*
Hardware yang digunakan pada pengujian ini dilakukan menggunakan *Notebook ASUS N82J* dengan spesifikasi sebagai berikut :

- *Processor : Intel Core i5-520M*
- *RAM : 8GB*
- *Hardisk : WD Black 500GB*
- *VGA : Nvidia GT335M 1GB*
- *OS : Windows 7*
- *Software* pendukung : *PostgreSQL 1.16.1 , Google Chrome 35.0.1916.153 M, XAMPP v3.1.0.*



Gambar 8. *Query count* pada tabel lib_buku dengan tanggal update '2010-03-30'

Tabel 1. Data pengujian waktu input data lib_buku tanpa penyetaraan

Tanggal *Update*	Total Data	Total Waktu
2010-03-30	432	> 3 detik
2010-03-29	500	≥ 3 detik
2010-03-28	1500	≥ 4 detik
2010-03-27	3000	≥ 5 detik

Tabel 2. Data pengujian input data lib_buku dan lib_judul dengan penyetaraan

Tanggal Update	Total Data	Total Penyetaraan	Total Waktu
2010-03-30	Lib_judul : 461 Lib_buku : 432	Bahasa : 8 Penerbit : 156 Jenis koleksi : 1	> 7 detik
2010-03-29	Lib_judul : 500 Lib_buku : 500	Penerbit : 144 Jenis Koleksi : 1	
2010-03-28	Lib_judul : 1000 Lib_buku : 1500	Bahasa : 9 Penerbit : 233 Jenis Koleksi : 3	> 9 detik
2010-03-27	Lib_judul: 2000 Lib_buku: 3000	Bahasa : 8 Penerbit : 385 Jenis Koleksi : 4	> 17 detik

5. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut :

- Perlu dilakukan penyetaraan terhadap isi dari tabel-tabel *master* (contoh : tabel m_bahasa) ketika proses *update database* dilakukan, agar proses pencarian lebih akurat dan struktur database lebih efisien. Catatan terhadap proses penyetaraan ini juga disimpan di tabel tersendiri agar sistem dapat melakukan pengecekan dan melakukan otomatisasi apabila dideteksi isi dari tabel client telah mengalami penyetaraan sebelumnya.
- *Client* dari *SVL* menggunakan 1 macam database yaitu *PostgreSQL*, karenanya tidak diperlukan untuk membuat *web services* dengan kemampuan mengambil data dari 2 jenis *database* semisal *PostgreSQL* dan *MySQL* seperti rencana semula.

6. DAFTAR PUSTAKA

[1] Juliandri (2011). Perancangan Situs Jejaring Sosial Menggunakan Konsep Following/Interest. Retrieved July 1, 2014, from http://repository.usu.ac.id/bitstream/123456789/26876/4/Chapter%20II.pdf

[2] Allamaraju,Subbu (2010). *RESTful Web Services Cookbook*. Sebastopol: O'Reilly Media.

[3] Webber, J., Parastidis, S., Robinson, I.(2010). *REST in Practice.* Sebastopol: O'Reilly Media.

[4] Mitchell, Lorna Jane. (2013). *PHP Web Services*. Sebastopol: O'Reilly Media.

[5] Pengenalan Web Service. Retrieved Mei 9, 2013, from http://elib. unikom.ac.id/download.php?id=132941