

# Jurnal Buana Informatika

- Pemodelan Lingkungan Virtual untuk Interaksi Avatar Berbasis Context Pada Proyek Digital Life at Campus (DiL@C)** 61-74  
*Haruno Sajati, Lukito Edi Nugroho, Ridi Ferdiana*  
(Universitas Gadjah Mada)
- Developing a Prototype of Mobile Dental Information System in Indonesia** 75-84  
*Leo Willyanto Santoso, Alexander Setiawan, Samuel Gunawan*  
(Universitas Kristen Petra)
- Aplikasi Penjualan dan SMS Gateway untuk Penagihan Hutang, Studi Kasus : PT. DEWATA Yogyakarta** 85-95  
*Brigitta Stellani Sukamto*  
(Universitas Atma Jaya Yogyakarta)
- Pengembangan Single-Account Untuk Berbagai Aplikasi Berbasis Web dan Desktop Memanfaatkan Account Windows Live** 96-107  
*Kusworo Anindito*  
(Universitas Atma Jaya Yogyakarta)
- Sistem Pendukung Keputusan Untuk Menentukan Pemberian Beasiswa di STMIK Atma Luhur** 108-118  
*Hilyah Magdalena*  
(STMIK Atma Luhur Pangkalpinang)
- Rancangan Model Manajemen Pengetahuan Untuk Perencanaan Pola Tanam Efektif Tanaman Pangan Berbasis Spatial Mining** 119-130  
*Eko Sedyono, Kristoko Dwi Hartono, Sri Yulianto*  
(Universitas Kristen Satya Wacana Salatiga)

UNIVERSITAS ATMA JAYA YOGYAKARTA

JBI

Volume 3

Nomor 2

Halaman 61-130

Yogyakarta, Juli 2012

ISSN 2087-2534

**JURNAL BUANA INFORMATIKA**  
**ISSN 2087-2534**  
**ISSN Online 2089-7642**

**Volume 3, Nomor 2, Juli 2012**

**Alamat Redaksi & Distribusi**

Tata Usaha Fakultas Teknologi Industri  
Universitas Atma Jaya Yogyakarta  
Jln. Babarsari No. 43, Yogyakarta 55281  
Telp. (0274) 487711 Fax. (0274) 485223

**E-mail** : [jbi@uajy.ac.id](mailto:jbi@uajy.ac.id)

**Website** : <http://jurnal.uajy.ac.id/jbi>

---

**DEWAN PENYUNTING**

**Penanggung Jawab**

Dr. B. Kristyanto

**Ketua Penyunting**

Findra Kartika Sari Dewi

**Penyunting Ahli**

Paulus Mudjihartono

Y. Sigit Purnomo WP

**Penyunting Pelaksana**

Th. Devi Indriasari

Bening Parwitasukci

**Mitra Bestari**

Prof. Suyoto

(UAJY, Indonesia)

Dr. Pranowo

(UAJY, Indonesia)

Dr. Lukito Edi Nugroho

(UGM, Indonesia)

**Setter**

Agustinus Kris Handoyo

**Administrasi**

Susana Juwiasih

**Distribusi**

Al. Susilo Harno

Y. Sumardi

*Jurnal Buana Informatika* diterbitkan oleh Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta sebagai media untuk menyalurkan pemahaman tentang aspek-aspek teknologi teknologi informasi berupa hasil penelitian lapangan atau laboratorium maupun studi pustaka. Jurnal ini terbit dua kali dalam setahun yaitu pada bulan Januari dan Juli.

Redaksi menerima naskah yang belum pernah diterbitkan dalam media lain dari dosen, peneliti, mahasiswa maupun praktisi dengan ketentuan penulisan seperti tercantum pada halaman dalam sampul belakang (Petunjuk Untuk Penulis). Naskah yang masuk akan dievaluasi dan disunting untuk keseragaman format, istilah, dan tata cara lainnya.

**JURNAL BUANA INFORMATIKA**

Volume 3, Nomor 2, Juli 2012

---

**DAFTAR ISI**

Kulit Muka

Dewan Redaksi

Daftar Isi

Permodelan Lingkungan *Virtual* untuk Interaksi *Avatar* Berbasis *Context* Pada 61-74  
Proyek *Digital Life at Campus (DiL@C)*  
*Haruno Sajati, Lukito Edi Nugroho, Ridi Ferdiana*  
*Universitas Gadjah Mada*

*Developing a Prototype of Mobile Dental Information System in Indonesia* 75-84  
*Leo Willyanto Santoso, Alexander Setiawan, Samuel Gunawan*  
*Universitas Kristen Petra*

Aplikasi Penjualan dan *SMS Gateway* untuk Penagihan Hutang, Studi Kasus : PT. 85-95  
DEWATA Yogyakarta  
*Brigitta Stellani Sukamto*  
*Universitas Atma Jaya Yogyakarta*

Pengembangan *Single-Account* Untuk Berbagai Aplikasi Berbasis *Web* dan *Desktop* 96-107  
Memfaatkan *Account Windows Live*  
*Kusworo Anindito*  
*Universitas Atma Jaya Yogyakarta*

Sistem Pendukung Keputusan Untuk Menentukan Pemberian Beasiswa di STMIK 108-118  
Atma Luhur  
*Hilyah Magdalena*  
*STMIK Atma Luhur Pangkalpinang*

Rancangan Model Manajemen Pengetahuan Untuk Perencanaan Pola Tanam Efektif 119-130  
Tanaman Pangan Berbasis *Spatial Mining*  
*Eko Sedyono, Kristoko Dwi Hartomo, Sri Yulianto*  
*Universitas Kristen Satya Wacana Salatiga*

Formulir Berlangganan

Petunjuk Untuk Penulis

Kulit Belakang

## Pengembangan *Single-Account* Untuk Berbagai Aplikasi Berbasis *Web* dan *Desktop* Memanfaatkan *Account Windows Live*

Kusworo Anindito

Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta

Jl. Babarsari 43, Yogyakarta 55281, Indonesia

Email: [kusworo@staff.uajy.ac.id](mailto:kusworo@staff.uajy.ac.id)

**Abstract.** Along with the increasing number of applications provided for students, a student needs to have different accounts to access each application. The accounts are used to access each specific application from campus or applications provided by another partner, such as Microsoft Live @.edu. As a result, these students have many accounts to access email, courses site, the study plan, result of study, etc. This situation often makes students forget their usernames or passwords to access the existing system applications. As a result, the administrator or operator often gets requests to reset passwords from students. This research is conducted in the effort to minimize the number of students who cannot access the information system because they cannot login. A student only needs to remember one account, i.e. his/her account of windows live, to be able to access various applications of information system, both on desktop and via web, provided by the campus.

**Keywords:** single-account, desktop, web, live@edu, windows live

**Abstrak.** Seiring dengan bertambahnya aplikasi layanan teknologi informasi yang disediakan bagi mahasiswa, maka seorang mahasiswa harus mengingat berbagai *account* yang dimiliki untuk mengakses berbagai aplikasi tersebut. *Account-account* tersebut digunakan untuk mengakses ke masing-masing aplikasi khusus dari perguruan tinggi tersebut ataupun *account* untuk mengakses aplikasi yang disediakan pihak lain yang menawarkan kerjasama, misalnya Microsoft live@edu. Akibatnya, mahasiswa tersebut memiliki *account* untuk mengakses email, situs kuliah, pengisian rencana studi, dan sebagainya. Hal ini sering membuat mahasiswa tersebut lupa *username* atau *password* salah satu aplikasi yang ada, sehingga tidak bisa mengakses sistem. Akibatnya, administrator/operator mendapatkan permohonan untuk mereset *password* para mahasiswa yang lupa tersebut. Penelitian ini dilakukan sebagai usaha untuk meminimalkan kemungkinan mahasiswa tidak bisa mengakses sistem informasi karena tidak bisa melakukan *login*. Mahasiswa cukup mengingat sebuah *account*, yaitu *account* windows live, untuk bisa mengakses berbagai layanan sistem informasi, baik *desktop* maupun *web*, yang disediakan kampus.

**Kata Kunci:** single-account, desktop, web, live@edu, windows live

### 1. Pendahuluan

Pengembangan sistem informasi untuk melayani mahasiswa sering kali dilakukan secara bertahap. Hal ini dilakukan suatu kampus karena keterbatasan dana dan sumber daya, ataupun munculnya layanan baru. Akibatnya tiap aplikasi membuat sistem *account* sendiri. Hal ini membuat seorang mahasiswa yang memanfaatkan berbagai layanan memiliki banyak *account* dengan *username* dan *password* yang berbeda. Misalnya, sebuah perguruan tinggi mungkin telah mengembangkan sistem informasi akademik yang digunakan untuk mengelola informasi pribadi mahasiswa, pengambilan kuliah, nilai kuliah tiap semester, transkrip nilai, dan sebagainya untuk mahasiswa. Aplikasi ini berbasis *desktop*. Pihak perguruan tinggi kemudian juga mengembangkan situs kuliah untuk mengelola informasi mengenai perkuliahan dan materi perkuliahan. Mahasiswa bisa mengunduh materi kuliah dan tugas dari aplikasi berbasis *web*. Kemudian perguruan tinggi tersebut bekerjasama dengan Microsoft untuk memberikan layanan email, blog, dan media penyimpanan virtual, dalam paket live@edu, yang ditujukan bagi seluruh

sivitas akademik. Penggunaan *single-account* tentu saja akan membantu mahasiswa dalam menggunakan berbagai aplikasi ini. Penelitian ini dilakukan untuk mencari cara bagaimana sebuah *account* bisa digunakan di banyak aplikasi dan mengembangkan aplikasi yang ada agar bisa menggunakan *single-account* yang telah ditetapkan. Implementasi dilakukan pada sistem informasi akademik Universitas Atma Jaya Yogyakarta, yaitu SIATMA yang berbasis *web* dan *desktop*.

## 2. Tinjauan Pustaka

Kendali akses adalah cara bagaimana administrator mengendalikan siapa saja yang boleh mengakses server dan layanan apa yang diperbolehkan bagi mereka (CISCO, 2012). Keamanan jaringan dengan *authentication* (otentikasi), *authorization* (otorisasi), dan *accounting* (AAA) menyediakan *framework* utama bagaimana administrator mengendalikan pengaksesan terhadap *router* atau server. Otentikasi menyediakan metode pengidentifikasian pengguna, termasuk dialog *login* dan *password*, *challenge* dan *response*, dukungan komunikasi pesan (*messaging*) dan enkripsi (tergantung protokol keamanan yang dipilih).

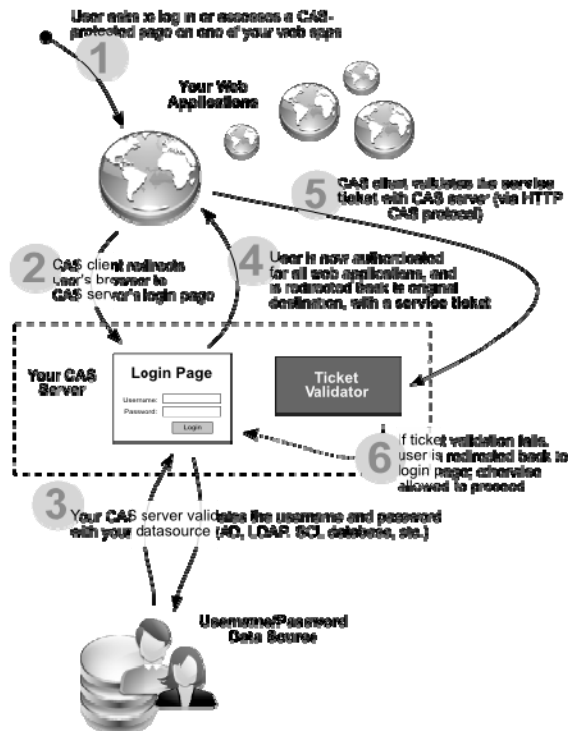
Seiring berkembangnya teknologi informasi yang mendukung proses bisnis, pengguna dan administrator dihadapkan pada semakin banyaknya aplikasi yang digunakan dalam pekerjaan mereka. Pengguna biasanya harus *sign-on* beberapa kali, sebanyak aplikasi yang dibutuhkan, masing-masing mungkin dengan data *login* (*account*) yang berbeda. Administrator juga harus mengelola banyak *account* pada masing-masing aplikasi. Otentikasi pada organisasi dengan berbagai aplikasi, seperti kasus diatas, sering kali mempersulit pengguna. Ada beberapa cara untuk menanganinya; teknik yang paling populer adalah dengan *single sign-on*, *SSO* (Balaji, 2011). *SSO* merupakan komponen yang sangat penting pada arsitektur keamanan sebuah organisasi (Ponnappalli, 2005). Dalam teknologi informasi diyakini bahwa implementasi *SSO* pada perusahaan mahal dan mungkin terus melebar. Bagaimanapun, perhatian perusahaan terhadap keuntungan pengimplementasian *SSO* semakin baik. Sebagian besar produk *SSO* yang ada di pasar, berdasarkan arsitekturnya, dapat dikategorikan menjadi dua tipe: 1) *Web-based* (juga dikenal sebagai *enterprise SSO* atau *ESSO*), 2) *Non web-based* (juga dikenal *legacy SSO*).

Dalam *SSO*, pendekatan sistem diperlukan untuk mengumpulkan informasi dan identifikasi yang diperlukan untuk mendukung otentikasi dari pengguna untuk setiap domain sekunder. Informasi yang diberikan oleh pengguna ini kemudian digunakan oleh layanan *SSO* dalam domain utama untuk mendukung otentikasi dari pengguna untuk setiap domain sekunder dimana pengguna berinteraksi. *Centralized Authentication Service (CAS)* merupakan sebuah protokol *SSO* untuk *web* (Vinmathi, 2011). Tujuannya memperbolehkan seorang pengguna untuk *log in* (secara otomatis) ke banyak aplikasi secara simultan. *CAS* juga memungkinkan aplikasi web melakukan otentikasi pengguna tanpa mengakses pengenalan pengguna, seperti *password*. Gambar 1 memperlihatkan mekanisme kerja dari *CAS*.

Pendekatan lain dilakukan oleh Fugkeaw dan kawan-kawan (Fugkeaw, 2007), mereka menerapkan konsep *SSO* dan *Multi-Agent System (MAS)* untuk menfalitasi proses otentikasi dan otorisasi dalam mengakses banyak aplikasi dan client secara dinamis dan efisien. Sari (Sari, 2006) mengimplementasikan otentikasi dengan *Lightweight Directory Access Protocol (LDAP)* memungkinkan setiap aplikasi berbasis *web* dapat secara terpadu menggunakan satu informasi identifikasi pengguna yang tersimpan di direktori server *LDAP*. Lewis dan kawan-kawan (Lewis, 2009) mengimplementasikan *Security Assertion Markup Language (SAML)* dan kemampuan untuk menyediakan solusi *SSO* yang aman untuk aplikasi-aplikasi yang diletakkan (*hosted*) di luar. Alkouz dan El-Seoud (Alkouz, 2007) menangani meningkatnya jumlah *account*, yang harus diingat untuk sistem *e-learning* yang terus berkembang, dengan mengimplementasikan *SSO* berbasis *web service*.

Konsep teknologi *web service* muncul untuk mendukung sistem terdistribusi yang memiliki infrastruktur yang berbeda (Deviana, 2011). *Web service* didasarkan pada sekumpulan standar untuk mendukung interoperabilitas antar aplikasi yang dibangun dengan bahasa yang berbeda dan berjalan di *platform* atau sistem operasi yang berbeda (Chen, 2003). *Web service* menyediakan standar komunikasi antara berbagai aplikasi perangkat lunak yang berbeda, yang

dapat dijalankan di berbagai *platform* dan *framework*. Konsorsium *W3C* mendefinisikan: "A *web service* is a software system designed to support interable machine-to-machine interaction over a network" (W3C, 2004). Dalam *web service* komunikasi dilakukan antara *provider* dan *requester*. *Provider* adalah orang atau organisasi yang menyediakan fungsi layanan tertentu, sedangkan *requester* adalah orang atau organisasi yang menggunakan layanan yang disediakan *provider*.

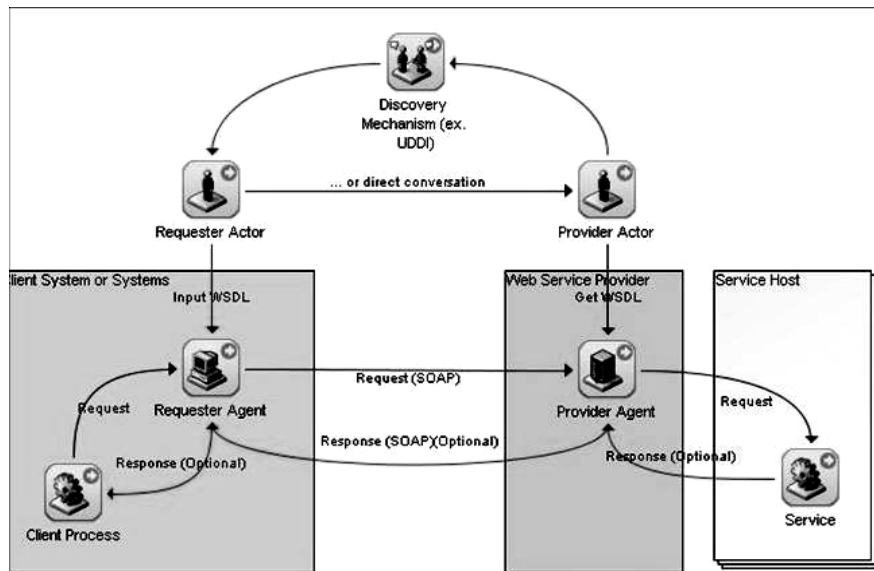


Gambar 1. Mekanisme Centralized Authentication Service (Vinmathi, 2011)

*Web Services* merupakan salah satu bentuk implementasi dari arsitektur model aplikasi *N-Tier* yang berorientasi layanan. Perbedaan *Web Services* dengan pendekatan *N-Tier* lainnya adalah dari segi infrastruktur dan dokumen yang digunakan sebagai format pertukaran data. Dalam implementasinya, *Web Services* tidak mempunyai tampilan, karena *web services* termasuk dalam *Business-Service tier*. Artinya didalam *Web Services* hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh aplikasi lainnya. *Web Services* menggunakan *XML* sebagai format dokumen dalam melakukan pertukaran datanya. Karena *XML* merupakan suatu format dokumen yang berbasis teks, maka *Web Services* memungkinkan berlangsungnya komunikasi antar aplikasi yang berbeda dengan *platform* yang berbeda pula. *Web Services* dapat diimplementasikan dalam berbagai jenis *platform* dengan menggunakan bahasa pemrograman apa pun, dan bisa digunakan oleh berbagai aplikasi yang menggunakan bahasa pemrograman apapun dengan *platform* apapun juga. Selama aplikasi tersebut dapat berkomunikasi dengan *Web Services* menggunakan protokol-protokol komunikasi. Termasuk *HTTP*, *XML*, *SOAP*, *UDDI* (*Universal Description Discovery and Integration*), dan *WSDL* (*Web Services Description Language*).

*XML Web Services* dapat digunakan secara internal oleh suatu aplikasi atau secara eksternal. Digunakan secara eksternal di sini maksudnya yaitu *XML Web Services* terdapat pada internet dan digunakan oleh berbagai macam aplikasi. *XML Web Services* mudah diakses melalui *interface* standar dan *XML Web Services* juga memungkinkan bermacam-macam sistem untuk bekerja bersama-sama dalam sebuah pekerjaan.

Gambar 1 menunjukkan gambaran sederhana konsep *Web Services* serta keterhubungan antara *Web Services* dengan aplikasi. *Web service* akan memberikan pengaruh yang sangat besar pada integrasi dan manajemen proses bisnis, paradigma *grid/utility/cloud computing*, *autonomic computing*, serta aplikasi bisnis dan sains (Shravani, 2011). *Web service* ini menerapkan teori untuk menjembatani gap antara layanan bisnis dan layanan teknologi informasi.



Gambar 2. Konsep *Web Service* [inter-locale.com].

### 3. Metodologi Penelitian

Bagian ini berisi analisis dan perancangan kebutuhan *single-account* serta spesifikasi kebutuhan dan perancangan perangkat lunak untuk mengembangkan sistem informasi yang ada di Universitas Atma Jaya Yogyakarta, yaitu SIATMA *desktop* (khususnya aplikasi pengisian KRS secara *online*, sebagai contoh aplikasi *desktop*) dan SIATMA *web* (Sistem Informasi Akademik Mahasiswa, sebagai contoh aplikasi *web*). *Account* yang dipilih sebagai account utama adalah *account* Windows Live, dengan pertimbangan kemudahan pengaksesan dari luar kampus dan ketidakmungkinan untuk melakukan modifikasi pada aplikasi milik *Microsoft* tersebut.

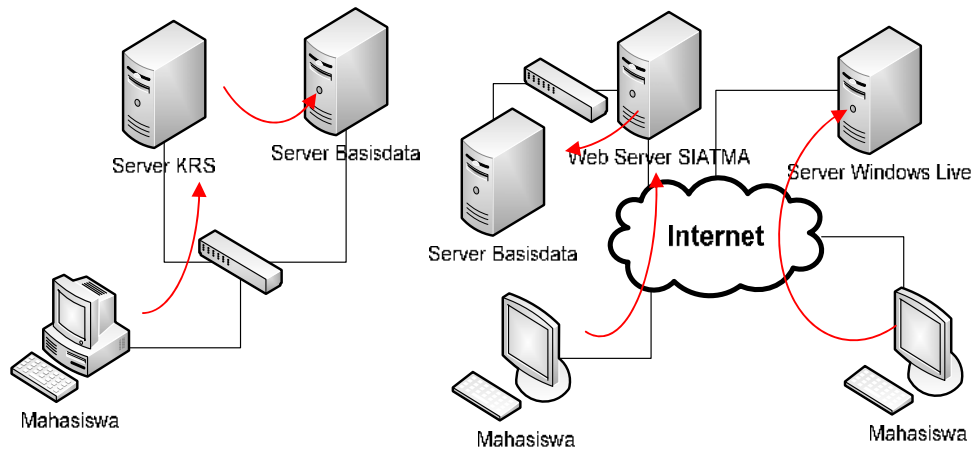
#### 3.1. Analisis Kebutuhan *Single-Account*

SIATMA *desktop* merupakan perangkat lunak *desktop* yang dikembangkan untuk menangani proses layanan akademik yang dilakukan Universitas Atma Jaya Yogyakarta. Sistem informasi ini menangani proses penawaran kelas kuliah, pengisian KRS, presensi, kelola nilai, dan lain-lain. SIATMA *web* merupakan perangkat lunak berbasis *web* yang dikembangkan untuk membantu proses perolehan informasi akademik mahasiswa selama berkuliah di Universitas Atma Jaya Yogyakarta. Sistem ini dapat menampilkan jadwal pribadi mahasiswa, menampilkan Kartu Hasil Studi mahasiswa, menampilkan transkrip mahasiswa, menampilkan persensi mahasiswa untuk matakuliah yang diambil, serta dapat menampilkan batasan-batasan dalam mengambil matakuliah tertentu seperti Tugas Akhir, Kuliah Kerja Nyata, maupun kerja praktek.

Gambar 3 menunjukkan bahwa ketiga sistem tidak saling berhubungan, dengan kata lain, data *account* yang dimiliki juga berbeda. Pemisahan basis data antara SIATMA *desktop* dan *web* sengaja dilakukan karena besarnya kemungkinan terjadinya serangan terhadap data mahasiswa melalui Internet. Data di SIATMA *web* merupakan replikasi dari data SIATMA *desktop*, yang di-*update* secara berkala. Pemisahan kedua basisdata itu membuat pengembang aplikasi membuat *user account* yang berbeda, sehingga mahasiswa harus mengingat banyak



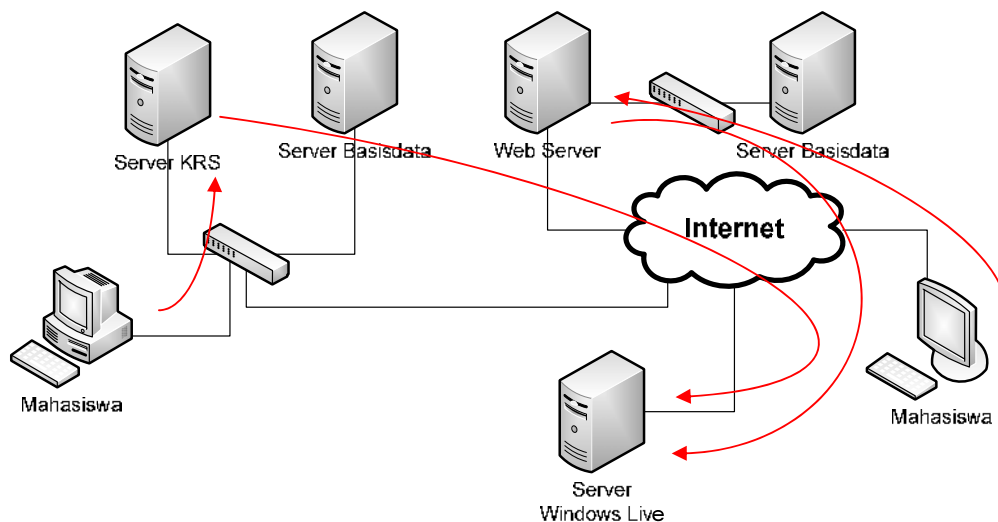
*account*. Hal ini yang menyulitkan mahasiswa, sehingga muncul kebutuhan untuk hanya mengingat satu *account* saja untuk login ke berbagai aplikasi.



Gambar 3. Setiap Aplikasi Mengakses Basisdata yang Berbeda

### 3.2. Perancangan Kebutuhan *Single-Account*

Analisis kebutuhan di atas menunjukkan bahwa terdapat kebutuhan untuk membuat *single-account* tetapi tidak melalui integrasi basisdata (basisdata tetap terpisah). Ada dua pilihan *account* utama, yaitu: 1) Memilih *account* yang tersimpan di salah satu basisdata, lalu dibuatkan layanan untuk melakukan validasi login. 2) Menggunakan *account* lain di luar kedua sistem informasi (basisdata) tersebut.



Gambar 4. SIATMA Web dan Desktop Mengakses Web Service dari Windows Live

Seperti telah diungkapkan sebelumnya, bahwa saat ini Universitas Atma Jaya Yogyakarta bekerjasama dengan *Microsoft* untuk memberikan layanan *email*, *blog*, dan media penyimpan virtual, dalam paket *live@edu*, yang ditujukan bagi seluruh sivitas akademik. Ini berarti, setiap sivitas akademik (termasuk mahasiswa) memiliki *account* di *Windows Live* yang digunakan untuk mengakses layanan-layanan dari *Microsoft* tersebut. *Windows Live* sendiri telah menyediakan layanan berupa *web service* untuk melakukan validasi *login*. Oleh karena itu, *account Windows Live* ini yang dipilih untuk dijadikan *account* utama, sehingga aplikasi-aplikasi tersebut akan mengakses ke server *Windows Live*, seperti terlihat pada Gambar 4.



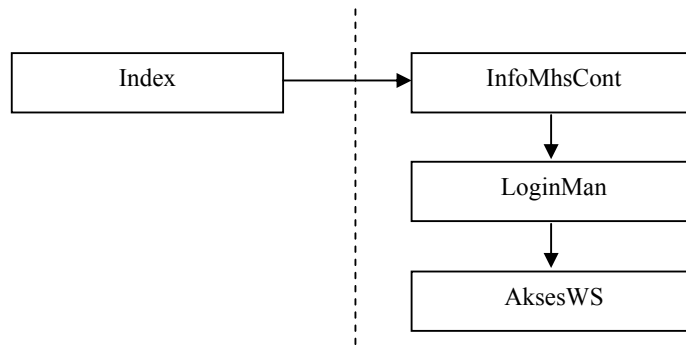
### 3.3. Spesifikasi Kebutuhan Fungsional

Penelitian ini hanya fokus pada proses *login*, oleh karena itu kebutuhan fungsionalitas hanya difokuskan ke *use case login* (dengan penambahan *use case* untuk megakses layanan *web service*) dan kebutuhan data yang diperlukan hanyalah data *account* sederhana, yaitu *username* dan *password*.

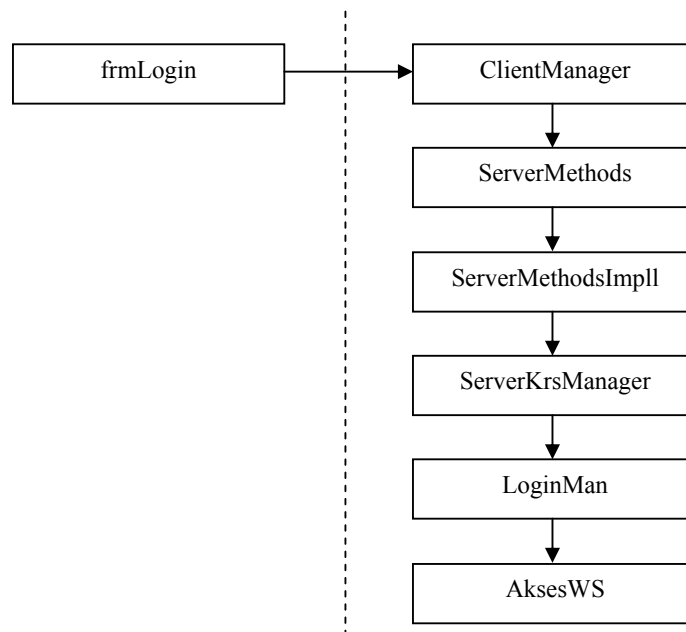
### 3.4. Perancangan Fungsional

#### 3.4.1. Perancangan Arsitektur

Perancangan arsitektur perangkat lunak ini melibatkan beberapa kelas yang ada dalam aplikasi yang akan menggunakan *single-account* ini, yaitu SIATMA *web*, seperti terlihat di Gambar 5, dan SIATMA *desktop*, terlihat di Gambar 6. Kelas *LoginMan* dan *AksesWS* merupakan kelas baru yang dibuat dalam penelitian ini.



Gambar 5. Arsitektur Perangkat Lunak Login SIATMA *web*

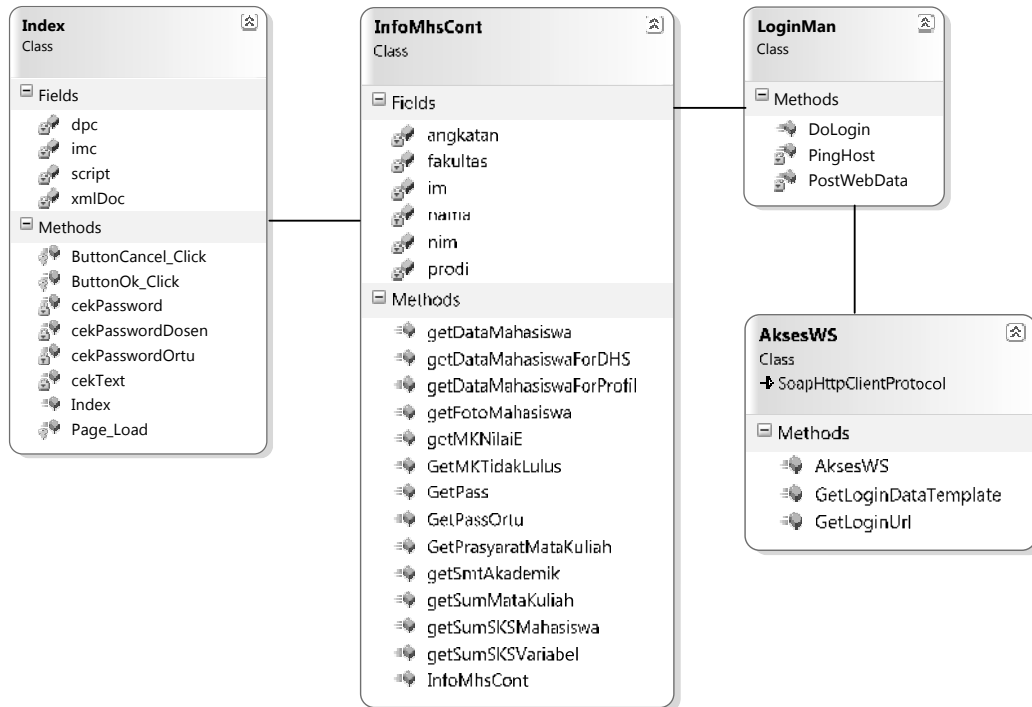


Gambar 6. Arsitektur Perangkat Lunak Login SIATMA *desktop*

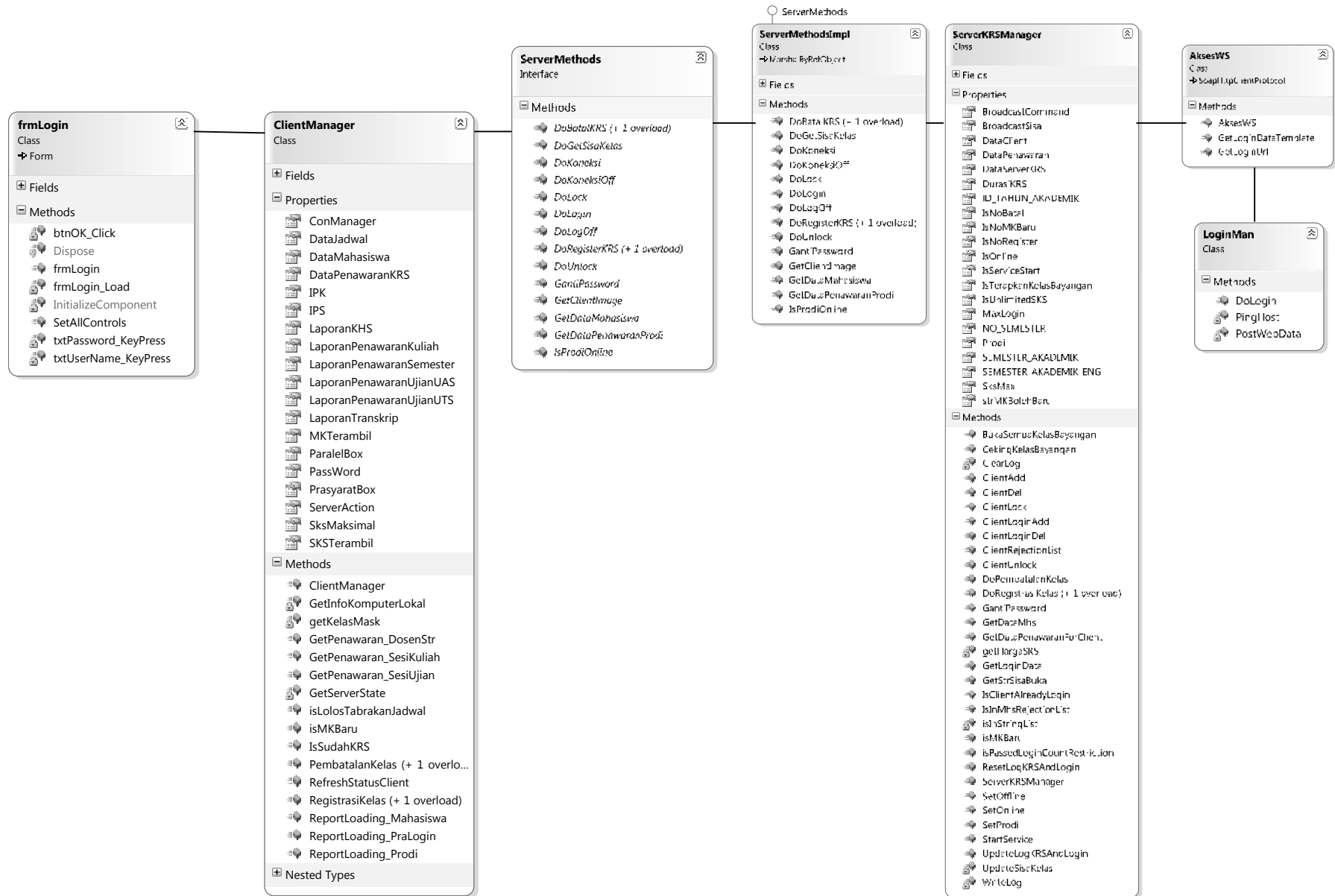
#### 3.5.2 Perancangan Rinci

Relasi antar kelas tersebut kemudian dapat didetilkan dengan memperlihatkan pemanggilan *method* kelas berdasarkan urutan waktu dalam bentuk *sequence diagram*. Gambar 7 menunjukkan urutan eksekusi *method* pada masing-masing kelas pada proses login SIATMA *web* dan Gambar 8 untuk SIATMA *desktop*.





Gambar 9. Class Diagram Proses Login SIATMA Web



Gambar 10. Class Diagram Proses Login SIATMA Desktop

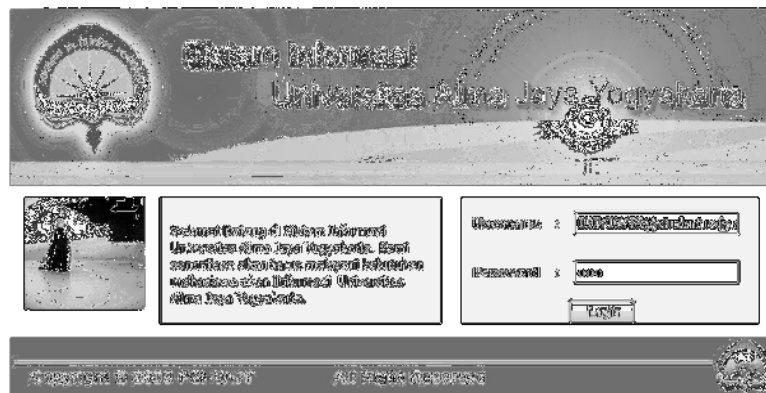
#### 4. Implementasi dan Pembahasan

Berdasarkan analisis dan perancangan yang dilakukan, *account* Windows Live dari pengguna, dalam hal ini mahasiswa, digunakan sebagai *account* bagi aplikasi SIATMA *web* maupun *desktop*. Dalam kelas *AksesWS* berisi fungsi-fungsi untuk memberi *Url web service* untuk *login* dan *template data login* yang sesuai dengan format dari *Windows live*. Kelas *LoginMan* berisi fungsi-fungsi yang digunakan untuk melakukan pengecekan apakah *web service* dapat diakses, meminta *request* untuk melakukan pengecekan terhadap data *login* yang dikirimkan, serta mengembalikan hasilnya ke *client*. Kedua kelas (*AksesWS* dan *LoginMan*) ditambahkan pada project masing-masing aplikasi, SIATMA *web* dan *desktop*. Ada kode program yang harus dimodifikasi pada kedua aplikasi tersebut. Dalam melakukan modifikasi program ini perlu dipertimbangkan lokasi modifikasi kode program untuk *login*. Aplikasi *desktop* SIATMA terdiri dari aplikasi *client* dan *server*. Pemodifikasian kode program dilakukan di aplikasi *server*, tepat pada kode validasi *login* yang terletak di fungsi *DoLogin()* pada kelas *ServerMethodsImpl*. Aplikasi SIATMA *web*, karena berbasis *web*, maka seluruh kode program terletak di *server*. Pemodifikasi kode program dilakukan pada fungsi *GetPass()* di kelas *InfoMhsCont*.

Secara umum penerapan *single-account* dengan menggunakan *account* dari *Windows Live* ini berjalan dengan baik. Seorang mahasiswa bisa *login* ke aplikasi SIATMA maupun SIAMA dengan *login account* yang sama. Tentu saja *account* tersebut juga bisa digunakan mahasiswa untuk mengakses layanan *Live@edu*. Penggunaan *single-account* ini sangat membantu mahasiswa untuk melakukan login di berbagai layanan aplikasi, karena hanya perlu mengingat satu *login account*. Pengguna, dalam hal ini mahasiswa, tidak perlu belajar menggunakan aplikasi lagi meskipun proses login sudah berubah, karena tidak ada perubahan antarmuka pengguna. Gambar 11 dan 12 memperlihatkan antarmuka pengguna untuk *login* dari kedua aplikasi.

Penggunaan *account Windows Live* sebagai *account* tunggal juga memiliki beberapa kekurangan. Karena layanan ini tersedia di Internet, maka penggunaan aplikasi SIATMA yang sebelumnya hanya berjalan di jaringan lokal, setelah penerapan *single-account* ini, kemudian menuntut adanya koneksi Internet. Selain itu, jika suatu saat *Microsoft* mengubah kebijakannya mengenai *Windows Live* ini, maka akan berpengaruh ke aplikasi tersebut.

Penelitian ini belum memecahkan masalah jika suatu saat koneksi Internet terputus atau server layanan bermasalah, padahal sedang dilakukan pengisian Kartu Rencana Studi secara *online*. Masalah tersebut bisa diatasi dengan membuat *account* sementara yang hanya berlaku pada rentang waktu tertentu. *Single-account* ini bisa dikembangkan lagi menjadi *single sign-on*, dimana pengguna hanya perlu *login* sekali untuk bisa mengakses berbagai layanan aplikasi.



Gambar 11. Antarmuka Login dari Aplikasi SIATMA Web



Gambar 12. Antarmuka Login dari Aplikasi SIATMA Desktop

## 5. Kesimpulan

Penerapan sistem *single-account* untuk berbagai aplikasi, baik *desktop* maupun *web* ini telah berhasil dikembangkan dengan baik. Pemilihan *account Windows Live* lebih didasarkan pada fakta bahwa seluruh sivitas akademika Universitas Atma Jaya Yogyakarta memiliki *Live Id*. Otentikasi *account Live Id* dilakukan dengan mengakses layanan *web service* yang disediakan oleh pihak *Microsoft*. *Ticket login* yang diperoleh oleh aplikasi menunjukkan pengguna tersebut memang pemilik *Live Id* tersebut, tetapi aplikasi tetap harus mengatur *role* (hak akses) pengguna tersebut pada aplikasi yang diakses.

Pemodifikasian kode program dilakukan di aplikasi, tepat pada kode validasi *login*. Kode bisa disisipkan dalam bentuk kelas, yang kemudian diakses dari fungsi validasi *login* dari tiap aplikasi. Pengaksesan ke layanan otentikasi harus mengikuti aturan main dari pemilik layanan, misalnya urutan pengaksesan dan format datanya. Pada aplikasi *desktop* yang memiliki aplikasi *server* yang terpisah, pengaksesan layanan otentikasi dilakukan dari aplikasi *server* (bukan di *client*). Sedangkan pada aplikasi *web*, karena semua kode program ada di *server*, maka perubahan dilakukan pada fungsi validasi *login*.

Dengan penerapan *single-account* ini diharapkan, baik pengguna maupun administrator, semakin lebih mudah dalam menggunakan dan mengatur berbagai aplikasi yang harus diakses. Penggunaan *single-account* ini sangat membantu mahasiswa untuk melakukan *login* di berbagai layanan aplikasi, karena hanya perlu mengingat satu *login account*. Meskipun demikian, *single-account* ini masih menuntut pengguna untuk *login* beberapa kali, di masing-masing aplikasi yang diakses. Karena itu, diharapkan integrasi *account* ini dikembangkan lagi menjadi *single sign-on*, dimana pengguna cukup *login* sekali untuk dapat mengakses berbagai aplikasi.

## Referensi

- Alkouz, Akram, S.A. El-Seoud. 2007. Web Services Based Authentication System for E-Learning. International Journal of Computing & Information Sciences Vol. 5, No. 2, August 2007, On-Line.
- Balaji, Prof. N. Prasanna, U. Sreenivasulu, C. Venkateshwar Reddy. 2011. Web-Based System—Authentication to Single Log-on to Several Applications, International Journal of Computer Science and Telecommunications [Volume 2, Issue 7, October 2011].
- Chen, Minder, A.N.K. Chen., B.B.M. Shao. 2003. The Implications and Impacts of Web Services to E-Commerce Research and Practices. Journal of Electronic Commerce Research, VOL. 4, NO. 4, 2003.

- CISCO. 2012. *Cisco IOS Security Configuration Guide, Release 12.2: AAA Overview*, (Online), ([http://www.cisco.com/en/US/docs/ios/12\\_2/security/configuration/guide/scfaaa.html#wp1000871](http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfaaa.html#wp1000871), diakses 12 Maret 2012).
- Deviana, Hartati. 2011. *Penerapan XML Web service Pada Sistem Distribusi Barang*. Jurnal Generic, Vol. 6, No. 2, Juli 2011, pp. 55~62.
- Fugkeaw, Somchart, P. Manpanpanich, S.Juntapremjitt. 2007. *Multi-Application Authentication based on Multi-Agent System*, IAENG International Journal of Computer Science, 33:2, IJCS\_33\_2\_6.
- Open Group, The. 2012. *Introduction to Single Sign-On*, (Online), ([http://www.opengroup.org/security/sso/sso\\_intro.htm](http://www.opengroup.org/security/sso/sso_intro.htm), diakses 12 Maret 2012).
- Ponnapalli, Ravikanth. 2005. *Secure implementation of Enterprise single sign-on product in an organization*, SANS Institute InfoSec Reading Room.
- Phillips, Addison P. 2012. *Web Services and Internationalization*, (Online), (<http://www.interlocale.com/whitepaper/multilingual/ml73-ws-20050524.xml>, diakses 14 Maret 2012)
- Rudy, Riechie, Gunadi, O. 2012. *Integrasi Aplikasi Menggunakan Single Sign On Berbasiskan Lightweight Directory Access Protocol (LDAP) dalam Portal Binus@Access (Bee-Portal)*, (Online), (<http://ict.binus.edu/file/research/jurnal-skripsi-odie-v-2.1-revis-renan-RECEIVED.pdf>, diakses 14 Maret 2012).
- Sari, R.F., Hidayat, S. 2006. *Integrasi Mekanisme Autentikasi Aplikasi Web Server dengan Metode LDAP: Studi Kasus Aplikasi SIPEG UI*, Jurnal Teknologi Edisi Khusus No. 1.
- Shravani, D., Varma, P.S., Rani, B.P, Rao, K.V., Kumar, U. 2011. *Web Services Security Architectures for Secure Service Oriented Analysis and Design*. International Journal of Computer Trends and Technology- March to April Issue 2011.
- Vinmathi, M.S., B.Mohan,B., Prabhakar. 2011. *Centralized Authentication Services*, International Journal of Engineering Trends and Technology- May to June Issue 2011.
- Working Group, W3C. 2004. *Web Services Architecture*, (Online), (<http://www.w3.org/TR/ws-arch/>, diakses 14 Maret 2012).