IMPLEMENTASI SISTEM SINGLE SIGN ON / SINGLE SIGN OUT BERBASIS CENTRAL AUTHENTICATION SERVICE PROTOCOL PADA JARINGAN LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL UNIVERSITAS DIPONEGORO

Muhammad Yanuar Ary Saputro[*)], Kodrat Iman Satoto, and Adian Fatchur Rochim

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

[*)]*E-mail: technaturology@gmail.com*

Abstrak

Aplikasi *web* saat ini berkembang dengan pesat dikarenakan kehandalannya dan modularitas bahasa pemrograman *web* yang digunakannya. Tidak heran pada Universitas Diponegoro memiliki aplikasi email, blog, sistem informasi, ftp dan banyak aplikasi *web* lainnya. Hal ini sangat membanggakan, akan tetapi ada satu masalah yaitu proses *login* yang banyak pada aplikasi *web* di Universitas Diponegoro. Data *username* dan password bisa sama pada setiap aplikasi melalui penggunaan basis data yang terintegrasi pada semua aplikasi. Tetapi tetap saja pada saat kita membuka aplikasi kita selalu *login*. Sehingga dibutuhkan suatu Sistem *Single Sign On* / *Single Sign Out*, yaitu ketika kita sudah *login* atau *logout* satu kali, kita tidak perlu lagi *login* atau *logout* saat membuaka aplikasi lainnya Metodologi penelitian yang digunakan dalam Penelitian ini meliputi studi literatur, perancangan sistem, implementasi dan pengujian sistem. Studi literatur menggunakan metode pembelajaran berdasarkan buku-buku referensi dan paper yang terkait penelitian ini. Perancangan sistem menggunakan sistem operasi Linux dengan LDAP Universitas Diponegoro. Implementasi dilakukan dengan membangun layanan *web* CMS, multiblogging, *webcloud* dan *webmail*. Otentikasi layanan tersebut diintegrasikan Sistem SSO CAS Server dengan LDAP sebagai identity store. Penelitian ini menghasilkan sistem Sistem *Single Sign On* / *Single Sign Out* berbasis CAS Protocol pada jaringan berbasis LDAP. Server SSO CAS yang digunakan adalah CAS Server. Server SSO CAS digunakan sebagai halaman *login* terpusat bagi layanan *web* berbasis CMS, multiblogging, *webcloud* dan *webmail*. Sedangkan akun yang digunakan untuk *login* berasal dari LDAP milik Universitas Diponegoro. Keberhasilan ditentukan pada proses *login* dan *logout* salah satu aplikasi.. Jika salah satu aplikasi *login* / *logout* maka otomastis aplikasi lain akan *login* / *logout*.

Kata kunci : CAS, Single Sign On, Single Sign Out, Otentikasi, LDAP

Abstract

Web applications are currently growing due to the reliability and modularity of *web* programming language that they used. No wonder in Diponegoro University have *webmail* applications, blogs, information systems, ftp and many other *web* applications. It is very heartening, but there is one problem that is user still have too many *login* process at *web* application on Diponegoro University. *Login* can be used by the same *username* and password on each application through an integrated database on all application. But, when we open an application, we must still doing *login* process. So it needs a *Single Sign On* / *Single Sign Out* System, that is when we are doing a *login* or *logout* process in one application, we no longer need to do *login* or *logout* in other applications. The methodology that used in this final research include literature studies, system design, implementation and system testing. Literature studies using teaching methods based on reference books and papers related to this thesis. Designing systems using Linux operating system with the Diponegoro University's LDAP. Implementation is done by building a CMS-based *web* services, multiblogging, *webcloud* and *webmail*. Authentication services are integrated with CAS server *Single Sign On* / *Single Sign Out* System and LDAP as identity store. This final research bring result about *Single Sign On* / *Single Sign Out* System based on CAS protocol on LDAP-based network. SSO server that being used is CAS Server. CAS SSO Server is used as a centralized *login* page for CMS-based *web* services, multiblogging, *webcloud* and *webmail*. While the account that used to *login* is from Diponegoro University's LDAP. Success is determined by *login* and *logout* process on one application.. If one application success in *login* / *logout* then other applications will *login* / *logout* automatically.

Keywords : CAS, Single Sign On, Single Sign Out, Authentication, LDAP.

1 Pendahuluan

1.1 Latar Belakang

Perkembangan aplikasi *web* yang marak telah membuat Universitas Diponegoro memiliki banyak aplikasi *web* yang digunakan untuk meringankan beban pekerjaan dan mempermudah komunikasi serta pertukaran informasi

dalam linkungan akademis Universitas Diponegoro. Sebut saja sistem informasi penggajian, *webmail*, blog, *webcloud*, repositori, sistem informasi akademik dan lainnya.

Hanya saja aplikasi *web* tersebut masih dibangun secara tersendiri (*stand alone*), sehingga berdampak pula pada banyaknya sistem *login*, yang berbeda pada setiap aplikasi *web* di Universitas Diponegoro. Kadang kala *username* dan *password* antara satu aplikasi dengan aplikasi yang lainnya berbeda. Hal ini dikarenakan tidak adanya integrasi basis data antar aplikasi. Sedangkan bagi yang sudah terintegrasi basis datanya, hanya ada satu *username* dan satu *password* per-user, akan tetapi tetap saja setiap kali mengakses aplikasi *web* tersebut, pengguna harus *login* pada setiap aplikasi.

Proses *login* yang banyaknya sebanyak jumlah aplikasi yang tersedia, secara tidak langsung menjenuhkan pengguna. Hal itu dikarenakan pengguna harus menghafal *username* dan *password* pada setiap aplikasi dan menggunakan sebagian waktunya untuk proses *login* yang sama.

Untuk membuat proses *login* menjadi sederhana, maka diperlukan sebuah sistem yang disebut Sistem *Single Sign On/ Single Sign Out*, yaitu dimana kita hanya perlu *login/logout* pada salah satu aplikasi saja, dan tidak perlu *login/logout* lagi pada aplikasi lainnya. Sistem *Single Sign On/ Single Sign Out* akan memudahkan pengguna dalam mengakses banyak aplikasi sekaligus. Pengguna hanya perlu mengingat satu *username* dan satu *password* saja untuk semua aplikasi dan hanya perlu melakukan satu kali *login/logout* untuk mengakses semua aplikasi yang tersedia di Universitas Diponegoro. Salah satu produk Sistem SSO ini adalah *Central Authentication Services* (CAS) yang berbasis *Cental Authentication Service Protocol 2.0*. Sedangkan sebagai *user datastore* nya digunakan sistem direktori terpusat berbasis *Lightweight Directory Access Protocol (LDAP)* yaitu *OpenLDAP*.

1.2 Tujuan

1. Mempelajari, merancang dan mengimplementasikan Sistem *Single Sign On/ Single Sign Out* berbasis *Cental Authentication Service Protocol* pada jaringan *Lightweight Directory Access Protocol* milik Universitas Diponegoro.
2. Mengintegrasikan layanan *webmail*, *web*, *multiblogging*, dan *webcloud* dengan Sistem *Single Sign On/ Single Sign Out* berbasis CAS dan LDAP.

1.3 Batasan Masalah

Penelitian ini memiliki batasan pada permasalahan berikut :

1. *Server* CAS menggunakan sistem operasi Linux distro Ubuntu Server.
2. Integrasi CAS hanya dilakukan pada layanan *webmail*, *web*, *multiblogging*, dan *webcloud* serta otentikasi hanya dilakukan terhadap LDAP.
3. LDAP yang digunakan merupakan openLDAP milik Universitas Diponegoro.
4. Pembuatan layanan *email* menggunakan program *open source Postfix* untuk mesinnya dan *Squirrelmail* untuk aplikasi webnya .
5. Pembuatan layanan *web* menggunakan joomla.
6. Pembuatan layanan *multiblogging* menggunakan Wordpress-MU
7. Hanya digunakan aplikasi *web* berbasis PHP sebagai klien CAS
8. Tidak membahas masalah manajemen akun pengguna, instalasi dan konfigurasi di dalam openLDAP

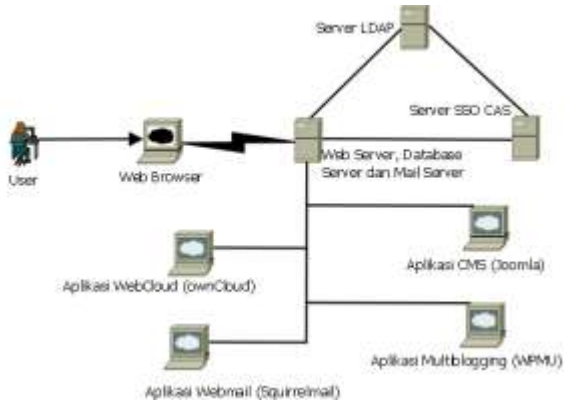
2 Metode

2.1 Perancangan Sistem Secara Umum

Metode yang digunakan adalah studi literatur dan perancangan sistem. Pada perancangan sistem, menggunakan 3 server masing – masing untuk *web server*, Server LDAP dan Server CAS. Layanan *Multiblogging*, *WebCloud*, *Webmail* dan *web* berbasis CMS berada dalam *web server*. Layanan *mail server* serta server basis data juga disatukan dengan *web server*. Kedua server yang lainnya adalah server LDAP dan server Single Sign-On / Single Sign-Out CAS.

Server LDAP dibuat terpisah dikarenakan server LDAP yang digunakan adalah milik Universitas Diponegoro, dan juga selain itu server LDAP hanya digunakan di jaringan lokal sehingga tidak membutuhkan IP Publik. *Web server* dan *mail server* ditempatkan berbeda dengan server LDAP karena merupakan layanan publik.

Server SSO CAS tersendiri, dibuat terpisah dari *web server* karena untuk memudahkan dalam pengembangan di masa depan termasuk dalam penambahan – penambahan aplikasi *web* lainnya. Server CAS ditempatkan terpisah juga agar tidak saling mengganggu antara Apache dan Apache Tomcat, karena CAS tidak berjalan pada mesin *web* Apache melainkan Apache Tomcat



Gambar 1 Topologi Sistem Jaringan yang dirancang

Gambar 1 memperlihatkan bagaimana aplikasi berjalan dan tampak disisi pengguna. Pengguna akan melihat tombol *login* di masing – maing aplikasi, saat pengguna mengakses aplikasi – aplikasi *web* seperti *Webmail, Multiblogging, WebCloud* dan *web* berbasis CMS,. Bila pengguna menekan salah satu saja tombol *login* pada salah satu aplikasi saja maka pengguna akan diteruskan ke halaman *login* server SSO CAS.

Server SSO CAS, memiliki halaman *login* yang meminta pengguna melakukan pengisian *username* dan *password*. *Username* dan *password* tersebut akan dicocokkan dengan data yang ada di dalam server LDAP.

Pengguna akan diteruskan kembali ke halaman klien CAS, jika *login* berhasil, yaitu halaman aplikasi *web* yang tadi coba diakses oleh pengguna yang memiliki sub program untuk koneksi ke klien CAS. Tiket yang didapatkan dari server SSO CAS akan dicek apakah sama dengan tiket di server SSO CAS, lalu didapatkan *username* dari pengguna. Setelah itu dilakukan pengecekan pada basis data lokal aplikasi *web* tersebut, apakah *username* tersebut sudah tersedia atau belum. Jika belum maka dilakukan pembuatan *username* tersebut pada basis data lokal aplikasi *web* tersebut, dengan menggunakan *username* yang didapat dari server CAS dan tingkatan hak akses *username* yang didapat dari proses *ldap_search* Jika *username* sudah tersedia maka akan dilakukan proses *login*, dan pembuatan *session* dan *cookie* lokal pada subdomain aplikasi *web* tersebut.

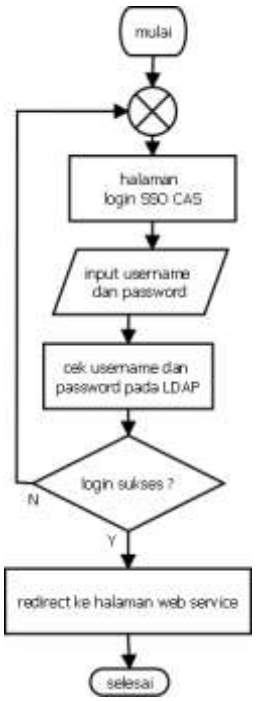
Apabila pengguna ingin mengakses aplikasi *web* lainnya, maka akan dilakukan pengecekan apakah masih ada *cookie* pada server SSO CAS, dalam hal ini disebut CASTGC. Bila *cookie* masih ada, maka akan dilakukan pengecekan tiket yang tersimpan pada CASTGC ke server SSO CAS, bila sama maka akan dilakukan proses *login* pada aplikasi *web* tersebut. Akibatnya saat pengguna begitu membuka aplikasi *web* lain setelah *login* di salah satu aplikasi *web* lainnya, maka pengguna akan langsung masuk ke halaman utama aplikasi *web* tersebut tanpa *login* lagi.

Aplikasi *web* akan meneruskan operasi *logout* ke halaman *logout* server SSO CAS, pada mekanisme *logout*. Setelah berhasil maka server SSO CAS akan meneruskn ke halaman *logout* aplikasi *web* tersebut, agar dapat dilakukan pemusnahan *session* dan *cookie* lokal aplikasi *web* tesebut. Akibatnya saat pengguna begitu membuka aplikasi *web* lain setelah *logout* di salah satu aplikasi *web* lainnya, maka pengguna akan langsung masuk ke halaman yang berisi tombol *login* aplikasi *web* tersebut tanpa perlu *logout* lagi.

2.2 Perancangan Sistem Server CAS

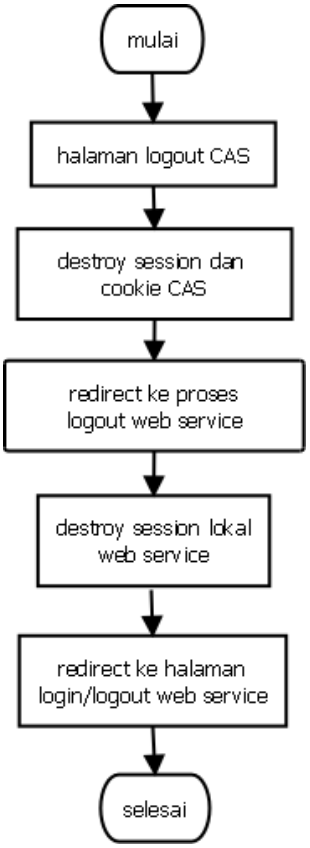
Perancangan Server SSO CAS pada penelitian ini menggunakan sistem operasi Linux distro Ubuntu Server 11.04 yang merupakan turunan Debian yang mana akan ditanamkan Java, Apache Tomcat, Maven dan Aplikasi Server CAS. Maven dibutuhkan untuk *building* aplikasi Server CAS sebelum dapat digunakan. Apache Tomcat berfungsi sebagai *web machine* tempat berjalannya aplikasi Server CAS.

Server CAS dibauat terpisah dari *web server*, dikarenakan untuk memudahkan pengembangan dan penambahan aplikasi – aplikasi *web* lain di masa yang akan datang dan juga agar kedua *web machine* (Apache dan Apache Tomcat) tidak saling mengganggu. Server CAS ini dihubungkan dengan server LDAP milik Universitas Diponegoro yang sudah tersedia, dan pada gambar 2 adalah *Flowchart* sistem *login* aplikasi Server CAS.



Gambar 2 *Flowchart* proses *login* Aplikasi CAS Server

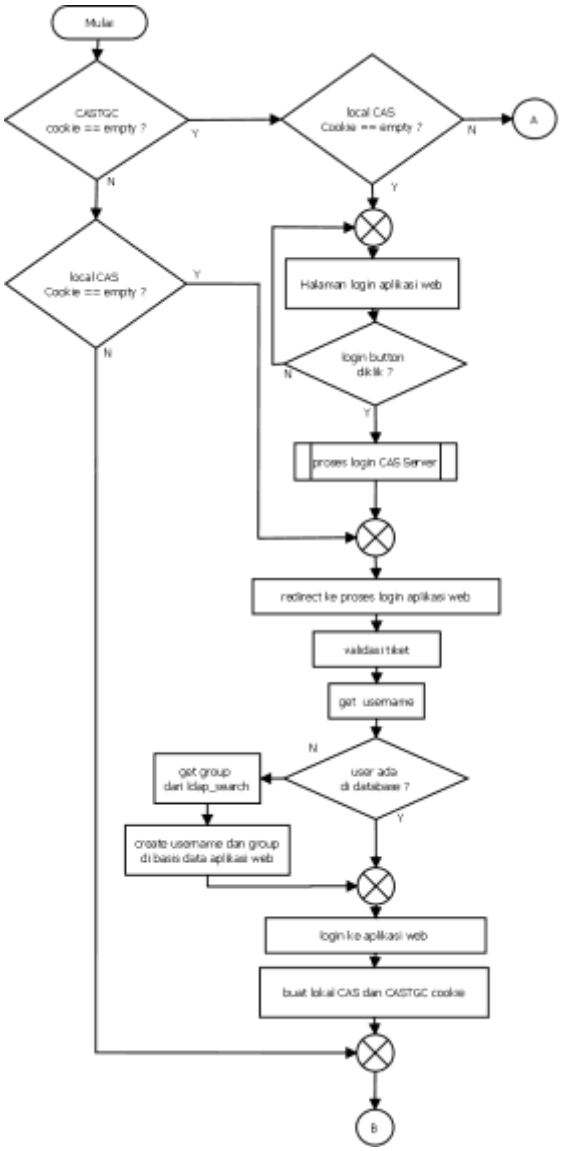
Proses *logout* CAS digambarkan pada *flowchart* di gambar 3



Gambar 3 *Flowchart* proses *logout* Aplikasi CAS Server

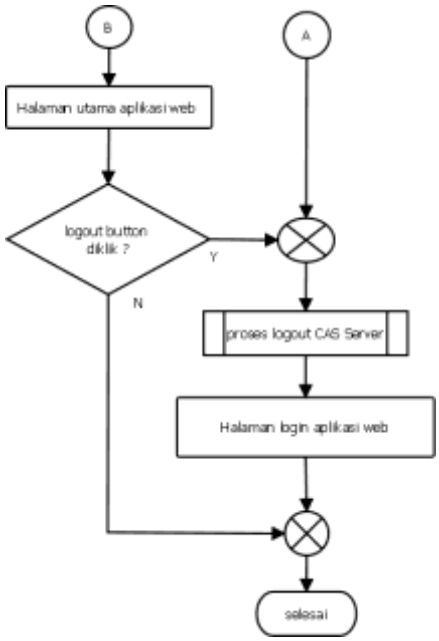
2.3 Perancangan Sistem Klien CAS

Perancangan Klien SSO CAS pada penelitian ini menggunakan sistem operasi Linux distro Linux Mint 10. Server *Web* memerlukan aplikasi *web* berbasis CMS seperti Joomla, *weblog* seperti WPMU, *webcloud* seperti ownCloud dan *webmail* seperti Squirrelmail. Kesemua aplikasi tersebut memerlukan Apache, PHP dan MySQL karena berjalan pada Apache, dan mengunakan bahasa PHP serta basis data MySQL. Program kecil tambahan yang dipasang pada sistem *login* aplikasi *Web* tersebut sangat diperlukan, agar sistem *login* aplikasi *Web* berbasis CMS dapat terintegrasi dengan proses *login* server CAS.



Gambar 4 *Flowchart* proses *login* dan *logout* aplikasi *Web* berbasis CMS

Gambar 4 dan gambar 5 menjelaskan proses *login*/*logout* aplikasi *web* yang telah diubah sedemikian rupa sehingga mendukung *Single Sign On*/*Single Sign Out* terhadap CAS, yang mana dilakukan pengecekan *cookie* sebelum/ sesudah *login*/*logout.*



Gambar 5 Lanjutan *Flowchart* proses *login* dan *logout* aplikasi *Web* berbasis CMS

3. Hasil dan Analisa

Implementasi dilakukan dengan membuat proses *login*/*logout* aplikasi *web* seperti pada perancangan. Dimulai dengan persiapan sistem operasi dan CAS Server, instalasi aplikasi *web* meliputi Joomla, WPMU, Squirrelmail dan ownCloud serta konfigurasi aplikasi *web* tersebut agar mendukung SSO. Setelah semua paket telah terpasang, maka hal terpenting yang harus dilakukan adalah melakukan konfigurasi agar semua proses proses dapat berjalan dengan hasil yang baik.

3.1 Implementasi dan Konfigurasi CAS Server

Aplikasi CAS server merupakan aplikasi utama dalam implmentasi sistem SSO. Aplikasi ini berfungsi melayani layanan *Single Sign On* / *Single Sign Out*. Aplikasi ini meneruskan ke halaman *web* service dan memberikan tiket sebagai bukti hak akses. Otentikasi CAS menggunakan LDAP. Langkah pertama adalah mempersiapkan server Ubuntu Server 11.04 LTS dengan Tomcat Java Server dan adanya aplikasi Apache Maven serta SSL. Kemudian dilanjutkan dengan mengunduh source Aplikasi CAS Server terlebih dahulu.

Server CAS yang telah terunduh, perlu dikonfigurasi agar menggunakan LDAP sebagai otentikasinya, yaitu dengan mengedit konfigurasi berkas deployerConfigContext.xml, cas-servlet.xml dan pom.xml. Stelah konfigurasi selesai dilakukan building dan hasil dari building berekstensi war dapat dijalankan di Apache Tomcat.

3.2 Implementasi dan Konfigurasi Aplikasi Web

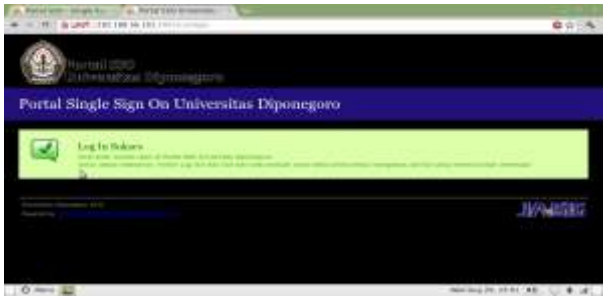
Aplikasi *web* yang dibutuhkan berbasis PHP, sehingga perlu diinstall Apache *web server*, php5, php5-dev, dan Mysql *server*. Apache *web server* bertugas menerima dan membalas permintaan situs yang datang dari klien di *web server*. Php5 berfungsi mendukung bahasa pemrograman php pada *web server*. Php5-dev digunakan sebagai syarat implementasi *PHP Accelerator* dengan perangkat lunak eAccelerator. Mysql *server* sebagai basis data berfungsi sebagai syarat mengimplementasi Joomla, WPMU dan ownCloud.

Konfigurasi agar mendukung SSO dilakukan pada berkas yang berbeda-beda pada masing – masing aplikasi web. Akan tetapi dibuat menggunakan konsep CASTGC. Tombol *login* dan *logout* dari masing – masing *web* harus diarahakan ke halaman *login* dan *logout* CAS dengan service URL halaman pemrosesan *login* dan *logout* dari aplikasi. Lalu halaman pemrosesan *logout* harus memuat fungsi GET terhadap tiket yang dikirimkan CAS, fungsi parsing xml terhadap protokol CAS yang mengirim validasi *login* berupa *username*, dan fungsi autocreate *username* di basis data aplikasi serta fungsi *login* aplikasi dengan hanya menggunakan informasi *username*. Halaman *logout* tidak perlu memuat fungsi khusus selain fungsi pemusnahan *session* / *cookie* lokal aplikasi web.

Selain halaman *login* dan logot, halaman utama aplikasi / halaman yang sering dimuat ulang, harus memiliki fungsi pengecekan *cookie* CASTGC dan *cookie* CAS lokal untuk auto*login* atau auto *logout* jika pengguna telah *login* atau *logout* dari aplikasi lain.

3.3 Hasil Pengujian *Single Sign On* / *Single Sign Out*

Pengujian sistem *Single Sign On* dilakukan terhadap beberapa aplikasi *web* yang telah terinstall (Joomla, WordpressMU, ownCloud dan Squirrelmail), dengan cara melakukan *login* pada salah satu aplikasi saja, kemudian membuka ketiga aplikasi lainnya setelah *login* pada salah satu aplikasi saja.



Gambar 6 Percobaan *login* pada *webcloud*

Gambar 7 *Login* pada *webcloud* berhasil

Gambar 8 Aplikasi *webmail* tidak membutuhkan *login* lagi saat dibuka

Gambar 9 Aplikasi *web* CMS tidak membutuhkan *login* lagi saat dibuka

Gambar 10 Aplikasi *weblog* tidak membutuhkan *login* lagi saat dibuka

Pada gambar 6 dilakukan percobaan *login* pada salah satu aplikasi yaitu aplikasi *webcloud* ownCloud, ketika *login* berhasil, pengguna akan diteruskan ke halaman utama aplikasi (gambar 7). Keberhasilan *Single Sign On* terlihat pada gambar 8-10 yang mana pengguna langsung dapat mengakses halaman utama *webmail*, *web* dan *weblog*.

Sedangkan pengujian sistem *Single Sign Out* dilakukan terhadap beberapa aplikasi *web* yang telah ada (Joomla, WordpressMU, OwnCloud dan Squirrelmail), dengan cara dilakukan *logout* pada salah satu aplikasi saja, kemudian merefresh/membuka menu yang ada pada ketiga aplikasi lainnya setelah *logout* pada salah satu aplikasi saja.

Gambar 11 Pengujian *logout* pada aplikasi *webmail*

Gambar 12 Aplikasi *webcloud* tidak membutuhkan *logout* lagi saat dibuka

Gambar 13 Aplikasi *weblog* tidak membutuhkan *logout* lagi saat dibuka

Gambar 14 Aplikasi *web* tidak membutuhkan *logout* lagi saat dibuka

Gambar 11 menunjukkan pengujian *logout* pada aplikasi yang berbeda, yaitu *webmail*. Maka didapatkan hasil berupa *logout* dati *webmail*. Sama halnya dengan *Single Sign On*, keberhasilan *Single Sign Out* terlihat pada gambar 12-14 yang mana pada setiap aplikasi, pengguna akan langsung *logout* tanpa perlu menekan tombol *logout* lagi di tiap aplikasi

3.4 Kelebihan dan Kelemahan Sistem SSO CAS Server

Dalam sebuah sistem pasti terdapat kelebihan dan kelemahan. Hal itu juga berlaku pada Sistem *Single Sign On / Single Sign Out* CAS Server yang telah diimplementasikan ini. Pada pengujian *Single Sign On* dan *Single Sign Out*, terlihat kelebihan dari Sistem *Single Sign On / Single Sign Out* CAS Server, dimana pengguna hanya perlu melakukan satu kali *login / logout* saja untuk *login / logout* pada salah satu, beberapa atau semua aplikasi *web* yang tersedia (Joomla, WordpressMU, ownCloud dan Squirrelmail). Sistem ini memudahkan pengguna dalam melakukan proses *login / logout* pada aplikasi *web* yang tersedia. Pengguna pun tidak perlu menghafalkan *username* dan *password* pada setiap aplikasi web, yang mana bisa berbeda, karena sistem ini hanya menggunakan satu sumber otentikasi, yaitu LDAP milik Universitas Diponegoro.

Sistem *Single Sign On / Single Sign Out* CAS Server yang telah diimplementasikan ini juga memiliki kelemahan, disamping kelebihan berupa semua kemudahan yang diberikan kepada pengguna untuk *login/logout*, yaitu melalui otentikasi yang bersifat terpusat dan tunggal, mengakibatkan otentikasi dilakukan melalui server tunggal.

Jika dilakukan pengujian otentikasi saat server CAS dan server LDAP berada dalam kondisi mati, maka pengguna tidak akan bisa melakukan otentikasi. Jika pengguna tidak bisa melakukan otentikasi maka pengguna juga tidak dapat membuka salah satu, beberapa atau semua aplikasi *web* yang menggunakan otentikasi sistem *Single Sign On / Single Sign Out* CAS Server.

Perawatan secara berkala sangatlah diperlukan untuk mencegah matinya server SSO CAS atau server LDAP. *Clustering* juga dapat menjadi sebuah alternatif untuk mengatasi maslaha matinya server tersebut. Masalah lain yang timbul sebagai kelemahan sistem *Single Sign On / Single Sign Out* CAS Server, adalah masalah keamanan. Sebagai contoh adalah pencurian *cookie* atau adanya *LDAP injection*. Masalah keamanan sistem *Single Sign On / Single Sign Out* CAS Server tersebut dapat diatasi dengan pemeriksaan secara berkala dan pemberian sistem keamanan yang memadai. Sebagai contoh dengan menggunakan metode *CAS Proxy* dan menambal setiap lubang keamanan atau *vulnerability* pada aplikasi *web* yang ada.

4 Kesimpulan

1. Sistem *Single Sign On/ Single Sign Out* berbasis *Cental Authentication Service Protocol* di jaringan berbasis *Lightweight Directory Access Protocol* milik Universitas Diponegoro telah berjalan ditandai dengan keberhasilan otentikasi aplikasi *web* berbasis CMS, multiblogging, *webcloud* dan *web* mail melalui halaman *login* SSO CAS Server.
2. Sistem *Single Sign On* pada server SSO CAS telah berjalan ditandai dengan hanya membutuhkan satu kali operasi *login* melalui halaman *web* SSO CAS Server pada salah satu aplikasi saja, sedangkan saat membuka aplikasi lain, pengguna secara otomatis telah *login*
3. Sistem *Single Sign Out* pada server SSO CAS telah berjalan ditandai dengan hanya membutuhkan satu kali operasi *logout* pada salah satu aplikasi saja, sedangkan saat pengguna membuka menu atau merefresh aplikasi lain, pengguna secara otomatis telah *logout*.
4. Operasi otentikasi CAS Server menggunakan LDAP sebagai user data store dengan metode Fast Bind LDAP untuk otentikasi terhadap admin dan metode Bind LDAP untuk otentikasi pengguna yang berasal dari banyak direktori yang berbeda tetapi masih di domain yang sama.
5. Otentikasi CAS Server menggunakan metode tiket dengan menyimpan Ticket Granting *Cookie* (TGC) pada *cookie* sebagai bukti otentikasi pada browser dan mengirimkan Service Ticket (ST) kepada aplikasi *web* klien CAS sebagai bukti otentikasi dan pengganti password.
6. Sistem *login* pada klien CAS menggunakan mekanisme pengecekan validitas tiket untuk mendapatkan data *username* melalui protocol CAS, yang digunakan untuk *login* ke aplikasi dan membuat *session* lokal aplikasi.
7. Tingkatan hak akses dan grup pengguna yang dibutuhkan dalam pembuatan akun di aplikasi

webmail Squirrelmail dan aplikasi *webcloud* ownCloud didapatkan dengan cara melakukan pencarian RDN pada server LDAP dengan menggunakan *username* yang didapat melalui protocol CAS.

Referensi

[1]. Addison, Marvin S.,Scott Battaglia, Andrew Petro (2011). *Jasig CAS Documentation*. Jasig.

[2]. Arkills, Brian (2003). *LDAP Directories Explained: An Introduction and Analysis*. Addison Wesley. Boston, MA 02116, U.S.A.

[3]. Carter, Gerald (2003). *LDAP System Administration*. O'Reilly. 1005 Gravenstein Highway North Sebastopol, CA 95472, U.S.A.

[4]. Chopra, Vivek, Sing Li, Jeff Genender (2007). *Professional Apache Tomcat 6* .Wiley Publishing. Indianapolis, Indiana, USA.

[5]. Dent, Kyle D (2003). *Postfix: The Definitive Guide*. O'Reilly. 1005 Gravenstein Highway North Sebastopol, CA 95472, U.S.A.

[6]. Gilmore, W. Jason (2008). *Beginning PHP and MySQL from Novice to Professional*.Apress. Berkeley, USA.

[7]. Greenhill, Kathryn (2011) Flexible, customisable and good looking: multiple uses for Wordpress MU in two Australian Libraries, in 15th ALIA Information Online Conference & Exhibition, Feb 1-3 2011. Sydney, NSW: Australian Library and Information Association.

[8]. Grubb, Michael Flemin, Rob Carter. *"Single Sign On and System* Administrator*"* .Paper, Universitas Duke, Boston, 1998.

[9]. Huggins, Ronnie Dale. "*Web Access Management and Single Sign On*". Paper , IJCSI International Journal of Computer Science Issues, Vol. 2, 2009.

[10]. Nursyamsi. "*Implementasi Sistem Single Sign On berbasis Java*". Skripsi S-1, Universitas Sumatera Utara, Medan, 2009.

[11]. Putera, R. Fibrian Satya. "Sistem Otentikasi Terpusat Berbasis *Lightweight Directory Access Protocol*". Skripsi S-1, Universitas Diponegoro, Semarang, 2011.

[12]. Riyanto, Slamet. *Membuat Web Portal dengan Joomla*. Paper, IlmuKomputer.com, 2006.

[13]. Roberts, Craig, Richrd Shipman (2011). *Integrating Version Control into OwnCloud*. Department of Computer Science Aberystwyth University, Dyfed SY23 3AL, Britania Raya

[14]. Roger, Stuart J, Heesun Park .*"Single Sign On Configuration and Trobleshooting for SAS 9.2 Enterprise BI Web Applications"* . Paper, SAS Global Forum 2011, SAS Institute,Inc.

[15]. Rudy, Riechie, Ody Gunadi. "*Integrasi Aplikasi Menggunakan Single Sign On Berbasiskan Lightweight Directory Access Protokol (Ldap) Dalam Portal Binus@Ccess (Bee-Portal)*". Skripsi S-1, Universitas Bina Nusantara, Jakarta, 2009.

[16]. Sing Li. *Introduction to Apache Maven 2*. Paper, Wrox Press. 2006

[17]. Taylor, Carl, Alistair McDonald, David Rusenko, Patrick Ben Koetter, Ralf Hildebrandt, Magnus Back (2005). *Linux Email: Set up and Run a Small Office Email Server*. PACKT Publishing. Livery Street, Birmingham

[18]. Twist, J. . *Better understand cookies with ieHTTPHeaders*., from http://www.thejoyofcode.com/Better_understand_*Cookies*_with_ieHTTPHeaders.aspx, 2005