ANALISIS TEKNOLOGI *SINGLE SIGN ON* (*SSO*) DENGAN PENERAPAN *CENTRAL AUTHENTICATION SERVICE* (*CAS*) PADA UNIVERSITAS BINA DARMA

Gilang Ramadhan [1], Yesi Novaria Kunang [2] , Suryayusra [3]
Mahasiswa Universitas Bina Darma [1], Dosen Universitas Bina Darma [2]
Jalan Jenderal Ahmad Yani No.12 Palembang
Pos-el : gilang_09142162@yahoo.co.id[1], ykunang@yahoo.com[2],
Suryayusra@mail.binadarma.ac.id[3]

Abstract : *The continued development of technology and the many systems that make the quality of academic work and learning systems become more easily and efficiently . Universitas Bina Darma already have a quality system , in previous research studies have been conducted to develop a method of single sign on (SSO) based lightweight directory access protocol (LDAP) . But the central authentication service (CAS) at the SSO itself has not been implemented . CAS SSO protocol that is aimed at giving the user permission to access multiple applications , while providing the user credentials (such as user id and password) only once , and allow web applications to authenticate users without clicking - gain access to user credential security . this can facilitate the user in using existing applications and also to facilitate the organization of user data , so that the security of user data more secure , because using a centralized storage of user data . Therefore in this thesis research will be carried out with the title " Technology Analysis Single Sign On (SSO) with Application of Central Authentication Service (CAS) At Universitas Bina Darma " .*

Keywords *: Single Sign On (SSO) Lightweight Directory Acces Protocol (LDAP), Central Authentication Service (CAS).*

Abstrak : *Semakin berkembangnya teknologi dan banyaknya sistem berkualitas yang menjadikan sistem kerja akademik dan pembelajaran menjadi lebih mudah dan efisien. Universitas Bina Darma sudah memiliki sistem yang berkualitas, pada penelitian terdahulu telah dilakukan penelitian untuk mengembangkan metode single sign on (SSO) berbasis lightweight directory access protocol (LDAP). Akan tetapi central authentication service (CAS) pada SSO itu sendiri belum diterapkan. CAS merupakan protocol SSO yang bertujuan memberikan ijin pada pengguna dalam mengakses beberapa aplikasi, sekaligus menyediakan credential pengguna (seperti user id dan password) hanya sekali, dan mengizinkan aplikasi web untuk meng-otentikasi pengguna tanpa mendapatkan akses ke security credential pengguna. hal ini dapat mempermudah pengguna dalam menggunakan aplikasi yang ada dan juga dapat mempermudah dalam pengorganisasian data pengguna, sehingga keamanan data pengguna lebih terjamin, karena menggunakan tempat penyimpanan data user yang terpusat. Oleh karena itu pada tugas akhir ini akan dilakukan penelitian dengan judul* ***"Analisis Teknologi Single Sign On (SSO) dengan Penerapan Central Authentication service (CAS) Pada Universitas Bina Darma".***

Kata Kunci*: Single Sign On (SSO) Lightweight Directory Acces Protocol (LDAP), Central Authentication Service (CAS).*

---

1. PENDAHULUAN

Pada saat ini, perkembangan dunia informasi semakin hari semakin pesat. Hal ini sangat berpengaruh pada perkembangan internet. Berdasarkan Emarketer (2013), pengguna dari internet (www) akan meningkat dari 26.3% di tahun 2012 menjadi 30.7% di tahun 2016 untuk

wilayah Asia-Pasific. Internet membawa pengaruh besar atas ilmu dan pandangan dunia. Di Indonesia pengguna internet mencapai 63 juta pengguna pada tahun 2012, diperkirakan meningkat 30% pada tahun 2013 yaitu 82 juta pengguna menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII).

Internet adalah salah satu bentuk dari perkembangan teknologi informasi dan komunikasi (TIK). TIK mempunyai potensi yang sangat besar untuk dimanfaatkan dalam dunia pendidikan. Pada *blue print* TIK Depdiknas, setidak-tidaknya disebutkan ada tujuh fungsi TIK dalam pendidikan, yakni sebagai sumber belajar, alat bantu belajar, fasilitas pembelajaran, standar kompetensi, sistem administrasi, pendukung keputusan, sebagai infrastruktur (Koesnandar : 2008).

Penggunaan internet sebagai sarana pembelajaran menjadikan Universitas Bina Darma memiliki jaringan internet untuk mendukung pendidikan dan kreatifitas dalam proses belajar mengajar yang berkualitas, juga mempermudah komunikasi serta pertukaran informasi dalam lingkungan akademis. Infrastruktur Universitas Bina Darma memiliki banyak aplikasi yang membutuhkan otentikasi. Sebut saja diantaranya media pembelajaran *elearning*, *mail* Universitas, sistem informasi akademik dan lain sebagainya. Akan tetapi, aplikasi *web* yang ada belum terintegrasi dengan baik sebagaimana mestinya. Hal ini memberikan dampak pada banyaknya sistem *login* yang berbeda pada setiap aplikasi di Universitas Bina Darma yaitu pengguna harus *login* pada setiap aplikasi.

Pada penelitian terdahulu telah dilakukan penelitian dengan metode *single sign on (SSO)* berbasis *lightweight directory access protocol (LDAP)* yang telah diteliti oleh Dian Novera dari Universitas Bina Darma. Penelitian tersebut menghasilkan satu *username* dan satu *password* yang dapat mempermudah pengguna karena tidak perlu menggunakan banyak *account* serta menghafal banyak *password*. Akan tetapi *SSO* berbasis *LDAP* itu sendiri memiliki keterbatasan dimana pengguna tetap harus *login* pada setiap aplikasi yang ada satu persatu. Oleh karena itu penulis akan melakukan uji coba intergrasi pada semua aplikasi kedalam sebuah *web portal* untuk pengembangan selanjutnya dengan metode *SSO* berbasis *CAS (Central Authentication Service)* pada universitas Bina Darma.

SSO adalah sebuah system dimana pengguna cukup menggunakan satu *username* dan *password* untuk mengakses dan menggunakan layanan pada semua aplikasi yang ada. Sistem *SSO* memberikan efisiensi dan keamanan bagi pengguna dalam mengelola serta mengakses berbagai layanan aplikasi. Dan *LDAP* untuk sistem *directory* terpusat yang digunakan sebagai *datastore* nya. LDAP didesain untuk meng update dan mencari direktori yang berjalan lewat jaringan *TCP/IP.*

CAS yang berbasis *CAS Protocol* adalah salah satu produk dari *SSO*. *CAS* digunakan untuk menangani masalah komunikasi antara aplikasi yang berbeda. Dengan adanya *CAS* pada *SSO* semua aplikasi yang ada pada Universitas Bina Darma dimasukan kedalam sebuah *site* sehingga terbentuk sebuah integrasi aplikasi dalam bentuk *web* portal. Pengguna hanya perlu

satu kali *login* agar bisa menggunakan semua aplikasi yang ada didalam *web* portal tersebut. Pengguna juga tidak perlu menghafal banyak *account*, cukup satu *account*. Dengan demikian pengorganisasian data pengguna dapat dipermudah, sehingga keamanan data pengguna lebih terjamin, karena tempat yang digunakan untuk penyimpanan data pengguna menjadi terpusat.

Untuk itu salah satu solusi terhadap sistem otentikasi pengguna secara terpusat agar dapat mengakses semua aplikasi Di Universitas Bina Darma yang diharap dapat diterapkan dengan cara melakukan Analisis Teknologi *SSO* Dengan Penerapan *CAS* pada Universitas Bina Darma. Dimana pada penelitian ini bisa membantu untuk kemudahan dan keamanan pengguna dalam mengakses semua aplikasi.

Berdasarkan latar belakang diatas agar dapat terarah pada masalah yang ada serta tidak menyimpang, maka penulis merumuskan masalah dalam penelitian ini yaitu “Bagaimana mengintegrasikan dan memberikan izin dalam mengakses beberapa aplikasi web secara terpusat pada system dengan menggunakan Analisis Teknologi *Single Sign On (SSO)* dengan Penerapan *Central Authentication Service (CAS)* pada Universitas Bina Darma.

Dalam penelitian ini penulis membatasi permasalahan agar tetap terarah dan tidak menyimpang dari apa yang sudah direncanakan sebelumnya. Adapun batasan masalah pada penelitian ini adalah :

1. Pada *Server CAS* digunakan sistem operasi Linux distro Ubuntu *sever*.
2. Integrasi *CAS* hanya dilakukan pada layanan *mail server*, *elearning* dan *blog* serta sistem otentikasi menggunakan *LDAPserver.*
3. Pembuatan layanan *email* menggunakan komponen yang bersifat *client server* yaitu aplikasi *zimbra*.
4. Pembuatan layanan *web blog* dengan *wordpress.*
5. Pembuatan layanan *elearning* menggunakan *moodle*.
6. Hanya digunakan aplikasi *web* berbasis *PHP* sebagai klien *CAS*.
7. Tidak membahas masalah manajemen akun pengguna, instalasi dan konfigurasi di dalam open*LDAP.*

2. METODOLOGI PENELITIAN

2.1. Penelitian *Action Research* (Penelitian Tindakan)

Penelitian tindakan merupakan penelitian yang bertujuan mengembangkan metode kerja yang paling efisien, sehingga biaya produksi dapat ditekan dan produktifitas lembaga dapat meningkat (Sugiyono 2005:9).

Action Research menurut Davison, Martinsons, dan Kock (2004) yaitu penelitian tindakan yang mendeskripsikan, menginterprestasikan dan menjelaskan suatu situasi sosial atau pada waktu bersamaan dangan melakukan perubahan atau interversi dengan tujuan perbaikan atau partisipasi. Adapun tahapan penelitian yang merupakan bagian dari *Action Research* ini,yaitu:

1. *Diagnosing:* Melakukan identifikasi masalah-masalah yang ada pada penelitian sebelumnya, guna menjadi dasar kelompok atau organisasi sehingga terjadi perubahan selanjutnya.

2. *Action Planning:* Memahami pokok masalah yang ada, kemudian dilanjutkan dengan menyusun rencana tindakan yang tepat untuk menyelesaikan masalah yang ada.

3. *Action Taking:* Pada tahap pengimpmentasian rencana tindakan ini diharapkan dapat menyelesaikan masalah.

4. *Evaluating:* Setelah masa implementasi dirasa cukup, kemudian dilaksanakan evaluasi dari hasil mplementasi.

Specifying Learning: Tahap ini adalah bagian akhir yang telah dilalui setelah criteria dalam prinsip pembelajaran sehingga penelitian dapat berakhir dengan melaksanakan review tahapan-tahapan yang telah berakhir.

2.2. Metode Pengumpulan Data

Pada metode pengumpulan data untuk mendapatkan data dan informasi dilakukan dengan cara sebagai berikut:

1. Pengamatan (*Observation*): Peneliti mengadakan peninjauan langsung Ke Universitas Bina Darma Khusus dibagian unit pelayanan teknins (UPT-SIM) yang merupakan pusat sistem informasi di Universitas tersebut.

2. Wawancara (*interview*): Pada tahap ini dilakukan pengajuan pertanyaan-pertanyaan pada UPT-SIM universitas tersebut untuk mendapat informasi dan mendapatkan data-data secara langsung dari sumber yang mengetahui tentang penenelitian yang dilakukan penulis.

3. Studi Kepustakaan (*Literature*): Pada studi kepustakaan guna memperoleh data adalah dengan cara mencari bahan di internet, perpustakaan dan jurnal serta buku yang sesuai dengan objek yang akan diteliti.

3. HASIL

3.1. Evaluating

Pada Tahap ini menjelaskan tentang evaluasi dari hasil instalasi serta konfigurasi pada semua aplikasi web yang diuji coba dalam pembuatan simulasi ini.

3.1.1. Hasil Instalasi Dan Konfigurasi

3.1.1.1 Hasil instalasi Wordpress

Gambar 3.1 Database Wordpress Menggunakan PhpMyadmin

Gambar 3.1 adalah database dari Wordpress untuk web blog Bina Darma menggunakan PhpMyadmin.

Gambar 3.2 Halaman Wordpress Blog Bina Darma

Gambar 3.2 adalah tampilan dari halaman wordpress dalam simulasi web blog Universitas Bina Darma.

Gambar 3.3 Halaman Login Wordpress

Gambar 3.3 adalah tapilan login dari simulasi web blog Wordpress pada Universitas Bina Darma.

Gambar 3.4 Login Admin Wordpress

Gambar diatas menjelaskan tentang uji coba login pada admin.

Gambar 3.5 Halaman Admin Wordpress

Gambar 3.5 adalah halaman setelah proses login sukses, admin dapat masuk dan menggunakan layanan. Untuk melakukan konfigurai langkah yang dilakukan adalah menginstal *plugin* otentikasi *CAS* yang dapat di unduh pada http://downloads.wordpress.org/plugin/wpcas.zip. Setelah diunduh ekstrak plugin tersebut menjadi direktori wp-cas. Lakuakn penyalinan konfigurasi direktori kedalam wordpress/wp-content/plugins. Kemudian nama dari wp-cas-conf-sample.php dirubah menjadi wpcas-conf.php, lalu sesuaikan parameter tersebut dengan parameter yang ada didalam server *CAS*. Untuk mengaktifkan plugin dengan cara masuk kehalaman administarsi pada bagian menu *plugins* pilih wpcas kemudian klik *Activate*.

3.1.1.2 Hasil Instalasi Zimbra Mail

Setelah proses instalasi zimbra aplikasi web mail dari zimbra admin dan zimbra client sudah dapat digunakan.

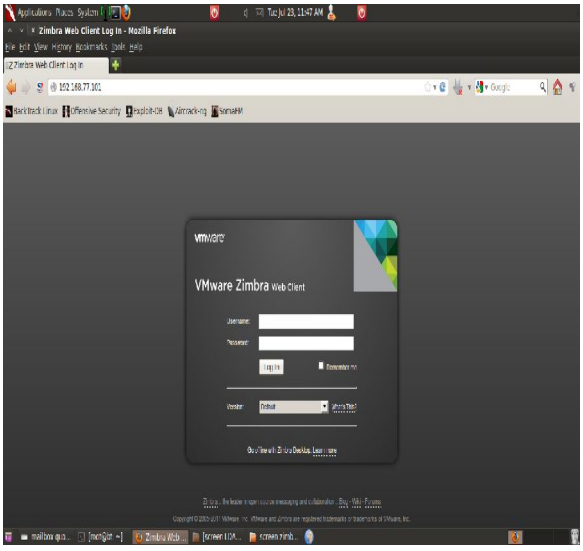
Gambar 3.6 Halaman Login Admin Zimbra Mail Bina Darma

Gambar 3.6 merupakan tampilan otentikasi login admin dari Zimbra yang digunakan sebagai aplikasi web dalam pembuatan simulasi email Bina Darma. Pada portal tersebut yang bisa mengakses hanyalah seorang administrator sendiri, karena portal dari user ada tempatnya sendiri. Disini seorang administrator memanajemen semua pengaturan pada mail server, baik itu penambahan user, penghapusan user, maupun pengaturan hak akses dari user yang terdapat pada mail server.

Gambar 3.7 Halaman Admin Zimbra Mail Bina Darma

Gambar 5.7 merupakan halaman admin zimbra setelah proses login berhasil kemudian akan tampil beberapa fasilitas yang bisa dimanfaatkan seorang administrator dari halaman tersebut.**Gambar 3.8** Otentikasi Mail Zimbra ke LDAP Server

Pada gambar 3.8 adalah proses Otentikasi konfigurasi yang dilakukan pada server LDAP untuk integrasi semua data.



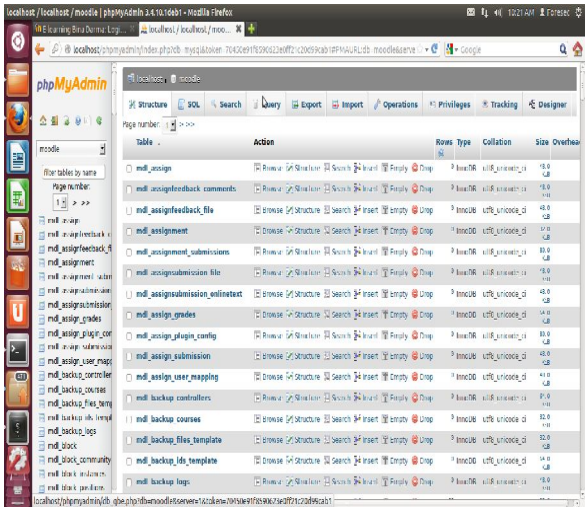
Gambar 3.9 Halaman Client Zimbra Mail Bina Darma

Gambar 3.9 merupakan halaman *client* zimbra pada simulasi Email Bina Darma. Dan ini berbeda dari halaman login seorang admnistrator, karena semua user bisa melakukan login melalui ini.

3.1.1.3 **Hasil Instalasi Elearning Moodle**

Setelah proses instalasi moodle selesai, *elearning* moodle dalam pembuatan simulasi ini adalah sarana untuk menunjang pembelajaran. Pada *elearning* moodle, dilakukan pengintegrasian dimana pada pengintegrasian ini ter bilang lebih sederhana dari pada aplikasi lainnya.

Dalam hal otentikasi aplikasi ini cukup fleksibel. Pada dasarnya moodle sudah mendukung *CAS* itu sendiri. Sebagai bukti dari dukungan moodle pada pengintegrasian CAS itu sendiri yaitu moodle tetap menyediakan form login bawaan dari aplikasi yang disediakan kepada pengguna yang tidak terdaftar di server CAS.

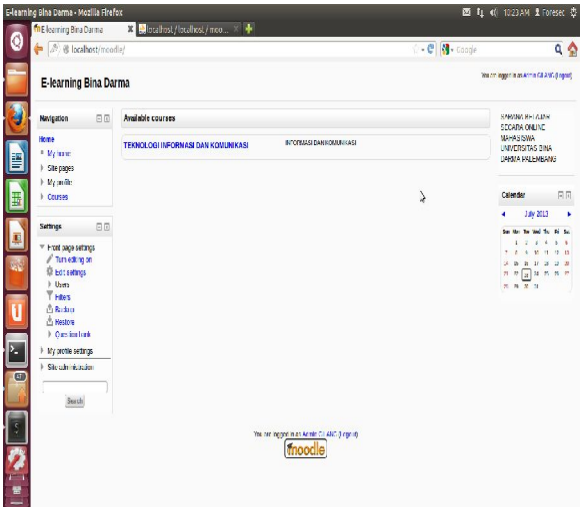


Gambar 3.10 Database moodle menggunakan PhpMyadmin

Gambar 3.10 adalah database dari aplikasi web moodle menggunakan Phpmyadmin.

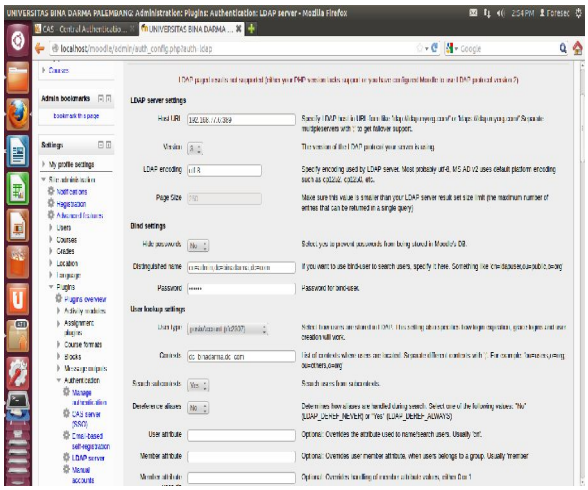
Gambar 3.11 Halaman Login Elearning Bina Darma

Gambar 3.11 adalah halaman login moodle dari pembuatan simulasi elearning pada Universitas Bina Darma setelah proses instalasi selesai. Disini adalah tempat untuk memasukan username dan password, jika pengguna atau admin memasukan kata sandi dan password dengan benar maka akan masuk kehalaman moodle dengan tampilan seperti gambar 4.12 dibawah ini.



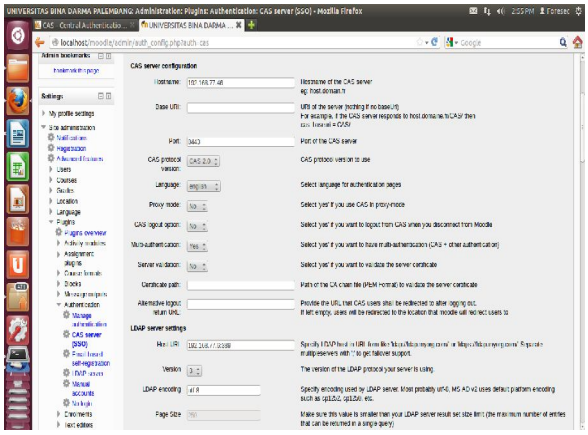
Gambar 3.12 Halaman Admin Elearning Bina Darma

Gambar 3.12 adalah halaman admin moodle setelah proses login admin sukses, pada halaman ini admin dapat menggunakan layanan yang ada.



Gambar 3.13 Konfigurasi Elearning Moodle pada LDAP Server

Gambar 3.13 menampilakan proses konfigurasi dari server elearning ke server LDAP untuk dilakukan integrasi. Untuk mengaktikan otentikasi CAS setelah melakukan konfigurasi ke LDAP dari halaman admin kita Users, Authentication, kemudian CAS server (SSO). Halaman yang akan tampil yaitu seperti gambar 3.14.



Gambar 3.14 Konfigurasi *Elearning* Moodle pada *CAS* server

Gambar 3.14 diatas adalah proses Otentikasi dari *elearning* moodle ke

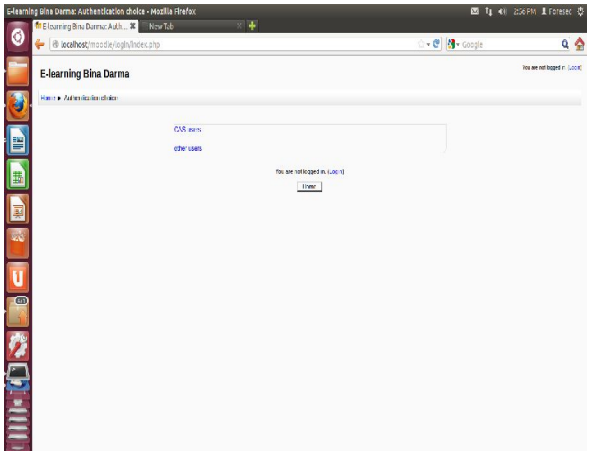
server *CAS*. Yang mana kemudian administrator melakukan pengisian dat pada parameter-parameter tersebut sesuai dengan *CAS* server yang telah dikofigurasi. Pada halaman konfigurasi terdapat sisa kolom dan biarkan tetap kosong seperti kondisi awal. Setelah proses konfigurasi selesai klik tombol *save*, maka konfigurasi akan tersimpan. Selanjutnya administrator melakukan konfigurasi pada pengaturan yang lainya seperti yang akan kita lihat pada gambar 3.15 untuk konfigurasi lanjutan.

Gambar 3.15 Konfigurasi Elearning Moodle Pada CAS server Lanjutan 1

Gambar diatas adalah lanjutan dari proses konfigurasi elearning moodle ke CAS server.

Gambar 3.16 Table database moodle yang sudah dikonfigurasi

Gambar diatas adalah gambar dari *database* yang sudah dirubah secara manual pada *phpmyadmin* karena konfigurasi tidak secara otomatis tersimpan pada database moodle. Oleh karena itu untuk menyimpan *database* ditulis secara manual pada kolom-kolom diatas. Proses tersebut dilakukan dengan cara rubah isi table mdl_user pada kolom *auth* pengguna untuk diganti dari sistem otentikasi yang manual menjadi *CAS*. Karena nilai awal kolom *auth* tersebut manual, dilakukan pengaktifan oetntikasi melalui *CAS* dengan cara merubah isi kolom tersebut menjadi *CAS*, lalu klik tombol *go* untuk menyimpan perubahan. Pastikan nilai dari kolom *auth* tersebut sudah berubah.

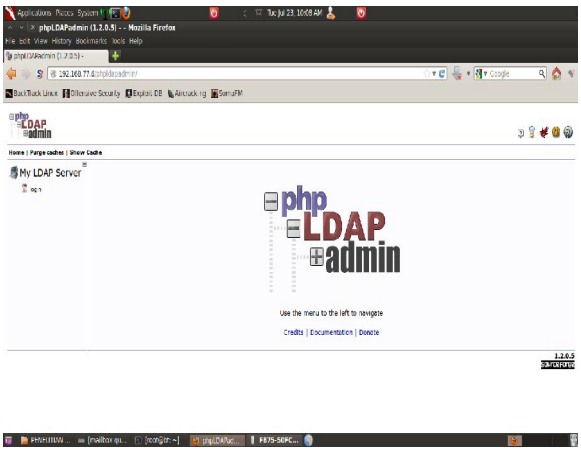


Gambar 3.17 Halaman Login Moodle setelah Dikonfigurasi

Gambar 3.17 adalah tampilan dari hasil konfigurasi moodle ke CAS server, dimana gambar tersebut menampilkan satu halaman login milik CAS server dan yang satunya untuk login menggunakan moodle. Hal itu dikarenakan moodle tetap menyediakan otentikasi login standar dari aplikasi yang disediakan kepada pengguna yang sebelumnya tidak terdaftar pada server CAS. Karena jika hanya ada satu otentikasi login yaitu menggunakan otentikasi milik CAS, maka user yang tidak terdaftar tidak

bisa mengakses moodle dikarenakan tidak punya akses untuk login.

3.1.1.4 Ldap server

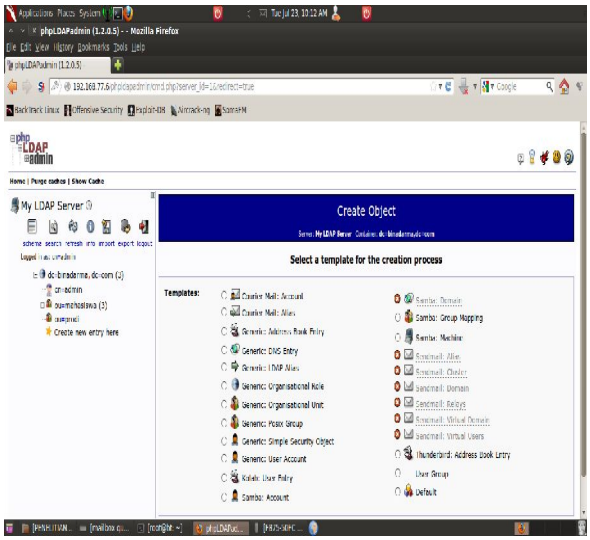


Gambar 3.18 Halaman LDAP server

Gambar 3.18 adalah halaman dari LDAP server dengan nama phpLDAPadmin yang sudah terinstal.

Gambar 3.19 Halaman Login *LDAP* Server

Gambar diatas adalah halaman login dari phpLDAPadmin. Dalam hal ini diperlukan proses otentikasi sebelum melakukan proses penambahan pengguna.



Gambar 3.20 Halaman Admin *LDAP* server

Gambar diatas adalah halaman admin dari phpLDAPadmin setelah melakukan proses login dengan cara memasukan kata sandi dari server *LDAP*.

Gambar 3.21 Halaman create objek

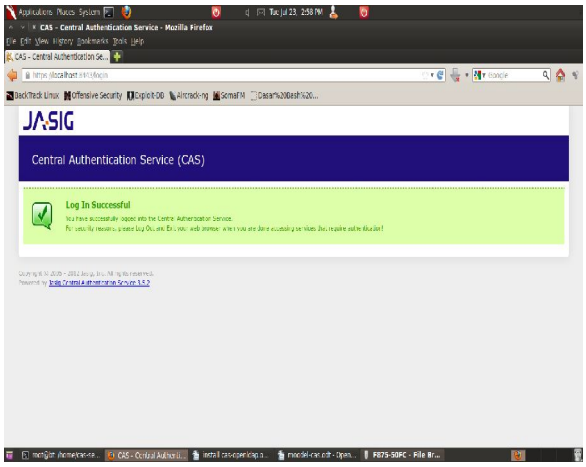
Gambar diatas adalah tampilan dari penambahan pengguna dimana setelah dilakukan penambahan pengguna , aplikasi wordpress, moodle, dan zimbra telah dapat

digunakan dengan aplikasi yang diintegrasikan ke CAS. Pengguna baru yang belum memilki *home* direktori pada server tidak bisa menggunakan aplikasi webmail. Karena pembuatan home belum dilakukan dan pembuatan tersebut dilakukan oleh administrator.

3.1.1.5 CAS Server

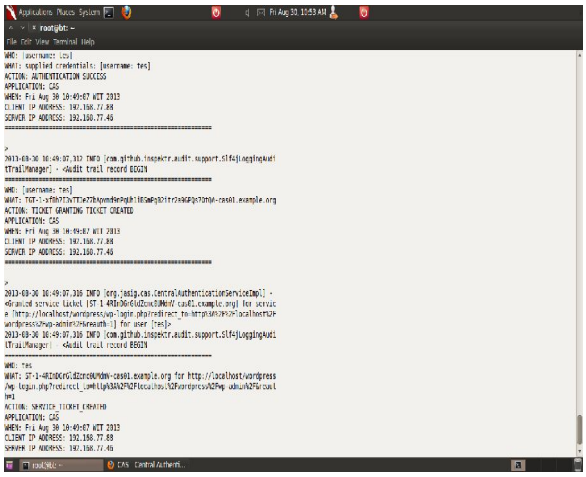
Gambar 3.22 Halaman Login CAS server

Gambar 3.22 merupakan halaman otentikasi login CAS server setelah proses instalasi, pada proses pengujian apakah CAS server telah terkoneksi pada LDAP server sebagai basisdata pengguna yang akan dilanjutkan ketahapan selanjutnya. Pada pengujian tersebut terjadi beberapa kesalahan mulai dari tidak cocoknya kata sandi dengan nama pengguna, sehingga pengguna tidak dapat terotentikasi. Lalu kesalahan lainnya karena server LDAP mati atau tidak bisa dihubungi yang mengakibatkan kegagalan otentikasi. Sehingga itu membuat penelitian menjadi sedikit terganggu karena harus menghilangkan masalah yang sering bermunculan seiring berjalannya penelitian.



Gambar 3.23 Login Sukses Pada *CAS* server

Gambar diatas adalah halaman login sukses setelah server *CAS* bisa berkomunikasi dengan server *LDAP* dalam pencocokan nama pengguna dan kata sandi kemudian *username* serta *password* diisi dengan benar. Dari gambar diatas maka server CAS telah siap untuk digunakan dalam pengotentikasian terhadap layanan yang dibuat karena proses otentikasi dengan server LDAP telah berhasil.



Gambar 3.24 Konfigurasi Integrasi CAS dengan TGC (Tiket Grating Cookie)

Setelah *CAS* server dapat berjalan dengan baik maka *CAS* server telah siap untuk melakukan integrasi dengan pustaka klien. Pada tahapan ini *phpCAS* yang digunakan sebagai pustaka klien. Mengimport pustaka dari *phpCAS* yang telah diekstrack adalah langkah pertama dari tahapan ini. Pengambilan pustaka ini harus dalam format *full path.* Setelah melakukan import pustaka, dilanjutkan dengan melakukan inisialisasi mesin pada server *CAS*. Hal ini dilakukan untuk memaksa pengguna yang belum terotentikasi untuk dibelokkan kehalaman otentikasi dalam server *CAS*. Kemudian program yang akan memanggil fungsi dari *getUser* untuk mengambil nama pengguna yang telah melakukan otentikasi dan menampilkannya dilayar dengan otentikasi yang dilakukan oleh pengguna. Jika pengguna akan melakukan proses *logout* maka program akan memanggil fungsi *logout* karena program akan mengambil data dari permintaan yang dilakukan oleh pengguna.

Layanan-layanan ini membutuhkan tingkat otentikasi lebih tinggi seperti php*CAS* yang membutuhkan sertifikat https yang ditanda tangani oleh *CA* (*Certificate Authority*) yang terpercaya untuk server *CAS.* Oleh karena itu sertifikat ssl yang digunakan harus resmi dan terpercaya. Apabila sertifikat ssl yang digunakan tidak ditandatangani oleh *CA* (*Certificate Authority*) yang terpercaya maka layanan-layanan tersebut tidak akan berjalan. Pada penelitian ini sertifikat ssl itu sendiri belum digunakan karena sertifikat yang resmi tersebut berbayar. Setelah pustaka klien telah berjalan dengan baik saat dijalankan pada *CAS* server maka *CAS* server telah siap dalam pengintegrasian pada layanan-layanan yang dibuat.

3.1.2. Hasil Pengintegrasian

Pada pembuatan simulasi dalam penelitian ini layanan yang akan diintegrasikan dengan *CAS* adalah zimbra sebagai layanan webmail, moodle untuk layanan elearning dan wordpress sebagai klien berbasis web penuh.

Gambar 3.25 Halaman Login Wordpress tanpa *CAS*

Gambar 3.26 Halaman login zimbra tanpa *CAS*

Gambar 3.27 Halaman login eleaning moodle tanpa *CAS*

Setelah dilakukan integrasi pada semua layanan kedalam *CAS* maka

semua halaman *login* akan di redirect ke halaman *login CAS* server.

Gambar 3.28 Halaman *login* Wordpress menggunakan *CAS*

Gambar 3.29 Halaman *login* Zimbra mail menggunakan *CAS*

Gambar 3.30 Halaman *login* moodle menggunakan *CAS*

CAS akan mengambil alih semua proses otentikasi dari yang awalnya menggunakan basisdata dari layanan itu sendiri setelah terntegrasi dengan *CAS*, integrasi layanan akan menggunakan basisdata pengguna dari *CAS*. Pada aplikasi moodle dan wordpress basisdata yang yang digunakan masih yang asli untuk mengecek ulang apakah pengguna yang terdaftar dalam server *CAS* mempunyai akun pada aplikasi tersebut, begitu pula pada proses *logout*.

3.1.3. Pengujian Integrasi Secara Manual

Pada tahapan ini dilakukan pengujian integrasi pada salah satu aplikasi web untuk mengetahui apakah database dari aplikasi-aplikasi web sudah terintegrasi atau terpusat. Pengujian dilakukan secara manual untuk melihat apakah aplikasi benar-benar terintegrasi. Salah satu dari ketiga aplikasi yang digunakan yaitu aplikasi moodle.

Gambar 3.31 halaman penambahan user pada LDAPserver

Gambar 3.31 menampilkan proses uji coba pembuatan user atau pengguna untuk pengujian integrasi. Dimana admin dapat melakukan penambahan, perubahan dan penghapusan data didalam server LDAP.

Gambar 5.32 Penambahan user Pada server LDAP

Dari gambar 5.32, menampilkan proses penambahan user atau pengguna dengan user “tes” dan password “123456” pada server ldap. Dimana dengan menambahkan user baru pada server ldap akan terintegrasi dengan seluruh client yang telah di atur sebelumnya. Sehingga ke depannya, jika user akan melalukan login menggunakan moodle, user tidak perlu menambahkan lagi akun baru pada server moodle, cukup dengan menambahkn akun user pada server ldap saja.

Gambar 3.33 hasil penambahan *user* pada server LDAP

Gambar 3.33 adalah tampilan dari hasil penambahan user dengan nama ‘tes’ yang telah selesai didaftarkan pada server LDAP. Dengan ini user ‘tes’ dan passwordnya sudah terintegrasi dan dapat digunakan untuk login pada layanan aplikasi yang dibuat. Setelah penambahan user ‘tes’ pada server LDAP, dilanjutkan dengan melakukan uji coba login pada aplikasi web elearning moodle.

Gambar 3.34 Uji coba login pada elearning moodle

Gambar 3.34 menampilkan proses uji coba login user dengan menggunakan user dan password yang dibuat di server LDAP. Dengan username ‘tes’ dan Password “123456”. Setelah proses login sukses, user dapat masuk kedalam halaman moodle, namu harus mengisi data terlebih dahulu. Seperti pada gambar 4.32 dibawah ini yang merupakan sarat utama bagi seorang user untuk melengkapi data mereka yang akan disimpan pada database moodle dan CAS. Pengisian data ini hanya dilakukan satu kali ketika user baru pertama kali login menggunakan user dan password yang baru didaftarkan pada server CAS.

Gambar 5.35 halaman pengisian data pada moodle

Proses login pada aplikasi elearning moodle telah berhasil. Dengan demikian integrasi dari LDAP server ke aplikasi tersebut telah berhasil dilakukan.

Proses uji coba login juga dilakukan pada CAS server dengan memasukan username ‘tes’ dan password “ 123456” yang dibuat pada server LDAP. Pengujian dilakukan seperti pada gambar 3.33 dibawah ini.

Gambar 3.36 Uji coba login pada CAS server

Gambar 5.37 Uji coba login pada sercer CAS berhasil

Dengan demikain proses dari integrasi pada semua layanan aplikasi pada pembuatan simulasi pada penelitian ini telah berhasil.

4. KESIMPULAN

Dari hasil penelitian pada pembuatan simulasi dengan metode Single Sign On (SSO) berbasis Central Authentication Service (CAS) pada Universitas Bina Darma ini dapat disimpulkan menjadi beberapa poin:

1. Penggunaan Sistem Single Sign On (SSO) dengan penerapan Central Authentication Service (CAS) pada Universitas Bina Darma akan dapat sangat

membantu karena pengguna portal tidak perlu menggunakan banyak account.

2. Penggunaan Lightweight Directory Access Protocol (LDAP) sebagai data store dapat membantu dalam pengorganisasian data user pada Single Sign On (SSO) ini.

3. Sebagai contoh pengguna yaitu mahasiswa Universitas Bina Darma merasa bahwa metode Single Sign On ini dapat bermanfaat bagi mereka. Hal tersebut di indikasikan dari hasil kuisioner yang dibuat.

Guna pengembangan serta mendukung keberhasilan dalam pemanfataan aplikasi ini ini, penulis memberikan saran sebagai berikut:

1. Agar sistem dapat berjalan, yaitu dengan melakukan pengembangan penelitian menggunakan sertifikat ssl yang resmi, diterbitkan oleh penerbit yang terpercaya.

2. Penelitian terhadap keamanan dan kinerja dari Sistem Single Sign On dengan penerapan Central Authetication Service (CAS) dapat dilakukan dengan melakukan analisis dari penelitian yang telah dilakukan.

5. DAFTAR RUJUKAN

Emarketer. (2013). "*Asia-Pasific: Digital Ad Spending Share WorldWide, by Region*, 2010-2016". *https://www.emarketer.com/Coverage/Asia-Pacific.aspx.* Diakses 22 April 2013.

Asosiasi Penyelenggara Jasa Internet Indonesia (APJII).http://www.apjii.or.id/v2/index.php/read/page/halaman-data/9/statistik.html. Diakses 21 April 2013.

Central Authentication Service (CAS). *www.jasig.org/cas*. Diakses 22 April 2013.

Koesnandar. (2008). "Pengembangan Bahan Belajar Bebasis Web". *http://www.teknologipendidikan.net/2008/02/12/pengembangan-bahan-belajar-berbasis-web/*. Diakses 22 April 2013.

Rudy, dan Riechie, Odi Gunadi. (2009). *Integrasi Aplikasi Menggunakan Single Sign On* Berbasiskan Lightwight *Directory Access Protocol (LDAP)* dalam Portal binus@ccess (BEE-PORTAL). Jakarta, Universitas Bina Nusantara.

Saputro, Muhammad Yanuar Ali. (2012). Jurnal: Implementasi Sistem *Single Sign On/Single Sign*

Out Berbasis *Central Authentication Service Protocol* Pada Jaringan Berbasis *Lightweight Directory Access Protoco*l. Universitas Diponegoro.

Dian, Novera. (2013). *Single Sign On (SSO)* dengan Menggunakan *Lightweight Directory Access Protocol*. Palembang, Universitas Bina Darma.

Cartealy, Imam. (2013). *Linux Networking*. Jakarta: Jasakom.

Sugiyono. (1999). Metode Penelitian Bisnis. Bandung: ALFABETA.

Davinson, R.M., Martinsons, M.G., Kock N. (2004). Jurnal: *Information Systems* dan *Principles of Canonical Action Research.*

Nasir, Moh Ph.D. (2003). Metode Penelitian. Jakarta: Ghalia Indonesia.

Tuxkeren. (2012). *Panduan Membuat Email Server dengan Zimbra*. Jakarta: Jasakom

Amiroh, S.Kom. (2012). *Kupas Tuntas Membangun E-learning dengan Learning Management System Moodle*. Jakarta Selatan: Genta Group Production PT Berkah Mandiri Globalindo

E-book. “Single Sign On, Keberos dan LDAP”. Universitas Sumatera Utara. *repository.usu.ac.id/bitstream/.../3/Chapter%20II.pdf*. Diakses tanggal 22 April 2013

http://www.binadarma.ac.id/content/43/5/Visi%20misi.html. Diakses tanggal 2 juli 2013

http://blog.binadarma.ac.id/upt/sample-page. Diakses tanggal 2 juli 2013

http://id.wikipedia.org/wiki/WordPress. Diakses tanggal 2 juli 2013

Anharku.(2009). *Flowchart.* http://ilmukomputer.org/wpcontent/uploads/2009/06/anharku-flowchart.pdf. Diakses tanggal 2 juli 2013