SISTEM OTENTIKASI TERPUSAT BERBASIS *LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL*

R. Fibrian Satya Putera[1)], Adian Fatchur Rochim[2)] Yuli Christiyono [2)]
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia
email : b1214ns@gmail.com

ABSTRACT

Along with the development of the Internet, many Internet service requires authentication process to validate or prove the identity of users who are entitled to use the service. Diponegoro University has a variety of services such as email, weblogs, Internet access and so on but the validation of the user should always be done to register the user on each application, even on the student email service there is no validation process, allowing users outside of the University of Diponegoro have an email address private university. With built a centralized authentication system then all the authentication process of the various services will be governed by a centralized authentication system and will not need to be made difficult by the problems in the validation because the user is registered in the system is a user that has been validated.

The final task was made by building a centralized authentication system at Diponegoro University computer network to be used by services that require authentication system. Designing the system to adjust to Diponegoro University environment. Implementation is done by building an email service, samba-PDC, weblogs and captive portal. Authentication services are integrated with centralized authentication system found on the LDAP server.LDAP server applications built using OpenLDAP.

This final result in a centralized authentication system based on Lightweight Directory Access Protocol used by email service, samba-PDC, weblogs, and a captive portal for authentication purposes. Test results on the success of the login process on all services and replacement of passwords users have to focus on a single application showed a centralized authentication system running well.

***Keyword** : Authentication, Centralized Authentication System , LDAP, OpenLDAP*

I. PENDAHULUAN

Latar Belakang

Perkembangan teknologi komputer dewasa ini sangat pesat, terlebih lagi ketika ditemukannya teknologi jaringan komputer dan *Internet*. Seiring dengan perkembangan *Internet*, banyak dibangun sistem yang bersifat *real-time* dan *online* yang memungkinkan seseorang dapat mengaksesnya dari mana saja dan mendapatkan informasi terkini. Sistem tersebut dibangun sendiri (*stand alone*) dan banyak yang membutuhkan sebuah proses identifikasi terhadap *user* yang berhak untuk mengakses sistem-sistem tersebut. Hal ini dapat terlihat pada permintaan *login* menggunakan *username* serta *password* sehingga hanya *user* yang berhak yang dapat mengakses atau masuk ke dalam sistem.

Sistem otentikasi terpusat yang melayani proses pembuktian identitas seseorang akan sangat berguna apabila terdapat banyak aplikasi yang membutuhkan proses otentikasi. Sistem ini memerlukan satu buah *user identity* yang unik untuk seorang *user* yang disimpan dalam sebuah *credential store*. Otentikasi dari semua aplikasi yang ada diatur oleh sebuah *server* otentikasi (*Central Authentication Services*) yang menangani validasi dan otorisasi *user identity* yang tersimpan dalam *server* yang menyimpan informasi *user* tersebut.

Salah satu metode otentikasi yang sering digunakan saat ini adalah otentikasi berbasis *Lightweight Directory Access Protocol* (LDAP). Dengan LDAP, pengguna bisa dimudahkan dalam pemakain kata sandi, dimana pengguna hanya mengingat satu kata sandi saja untuk banyak aplikasi. Akan tetapi pengguna tetap harus memasukkan nama dan kata sandi pada saat akan melakukan otentikasi pada masing-masing aplikasi.

Tujuan

Tujuan dari pembuatan tugas akhir ini adalah mempelajari, merancang, dan mengimplementasikan sistem otentikasi terpusat berbasis *Lightweight Directory Access Protocol* di jaringan komputer Universitas Diponegoro.

Batasan Masalah

Adapun pembatasan masalah pada makalah ini adalah sebagai berikut :

1. *Server* LDAP menggunakan sistem operasi Linux distro Ubuntu dengan aplikasi openLDAP.
2. Integrasi openLDAP dengan aplikasi *email, weblog, captive portal* dan samba-PDC.
3. Data primer berasal dari basis data Sistem informasi akademik Universitas Diponegoro.
4. Pembuatan layanan *email* menggunakan program *open source Zimbra collaboration suite*.
5. Pembuatan layanan *weblog* menggunakan wordpress-MU
6. Pembuatan *captive portal* menggunakan aplikasi *squid* yang telah terpasang di jaringan komputer Universitas Diponegoro.
7. Menggunakan sistem operasi Windows XP sebagai klien domain samba.

II. DASAR TEORI

Otentikasi dan Sistem Otentikasi

Otentikasi adalah proses usaha pembuktian identitas seorang pengguna sistem komunikasi pada proses *login* ke dalam sebuah sistem. Pengguna yang telah terbukti

1) Mahasiswa Teknik Elektro UNDIP

2) Dosen Teknik Elektro UNDIP

identitasnya adalah pengguna resmi pada sistem, orang yang memiliki otoritas atas sistem, atau mungkin aplikasi yang berjalan pada sistem.

Otentikasi seringkali diasumsikan identik dengan otorisasi, banyak protokol keamanan dan peraturan yang berdasarkan asumsi ini. Akan tetapi, penggunaan istilah otentikasi yang lebih tepat adalah pembuktian sebagai proses validasi identitas seorang pengguna, sedangkan otorisasi adalah proses validasi bahwa pengguna yang dikenal memiliki kekuasaan untuk melakukan tindakan tertentu

Sistem otentikasi adalah sistem yang melayani proses pembuktian identitas. Sistem otentikasi yang sering ditemui adalah sistem *login* suatu *website* atau pada sebuah komputer.

Direktori

Direktori adalah suatu tempat penyimpanan data yang dapat digunakan untuk memberikan informasi-informasi yang berkaitan dengan objeknya. Direktori berbeda dengan *database*, perbedaannya adalah direktori dibuat untuk dibaca lebih banyak dari pada ditulis. Sedangkan untuk *database* diasumsikan untuk operasi baca dan tulis memiliki frekuensi yang sama. Adapun karakteristik yang dimiliki oleh direktori adalah :

1. Masing-masing objek memiliki independensi dan terhubung dalam sebuah hirarki.
2. Memiliki skema penamaan sesuai dengan tipe dari objek yang pengaturannya dilakukan dalam sebuah hirarki.
3. Penggunaanya lebih sering dipakai untuk kepentingan baca (*read*) atau pencarian (*search*) dari pada untuk kepentingan *update* atau pengubahan
4. Lebih tepat dipakai untuk penyimpanan informasi yang bersifat statik karena tidak bisa melakukan perubahan informasi secara cepat.
5. Secara umum tidak mendukung transaksional data, misalnya transaksional data reservasi tiket. Hal ini diperlukan untuk menjaga kesederhanaan struktur dan kecepatan proses karena proses utama dalam direktori adalah pencarian.

Contoh umum dari direktori adalah *white pages* dan *yellow pages.* Dalam *yellow pages* seperi buku telepon, informasi dapat ditelusuri dengan menentukan kategori yang mungkin terkait.

X.500

X.500 merupakan salah satu sistem direktori yang pertama kali ditujukan untuk umum dan dirancang dari semula untuk dapat diperluas agar dapat melayani kebutuhan berbagai macam aplikasi. X.500 menyediakan operasi pencarian yang didukung berbagai macam *query*. X.500 merupakan standar terbuka yang tidak dikendalikan oleh satu vendor atau terikat pada sistem operasi, teknologi jaringan dan aplikasi tertentu.

Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) adalah sebuah protokol yang mengatur mekanisme pengaksesan layanan direktori yang dapat digunakan untuk mendeskripsikan banyak informasi seperti informasi tentang seseorang, organisasi, komputer dan banyak entitas lainnya. LDAP menggunakan model *client-server*, dimana klien mengirimkan *identifier* data kepada *server* menggunakan protokol TCP/IP dan *server* mencoba mencarinya pada DIT (*Directory Information Tree*) yang tersimpan di *server*. Bila di temukan maka hasilnya akan dikirimkan ke klien tersebut namun bila tidak maka hasilnya berupa *pointer* ke *server* lain yang menyimpan data yang di cari.

LDAP dikatakan ringan jika dibandingkan dengan layanan direktori X.500. Hal ini dikarenakan pada mulanya dasar LDAP sangat dekat terkait dengan X.500. LDAP pada mulanya dibuat sebagai protokol desktop yang lebih mudah dan digunakan sebagai *gateway* untuk X.500 *server*. Dengan demikian LDAP dikatakan ringan (*lightweight*) karena menggunakan pesan sedikit di dalam jaringan yang dipetakan secara langsung pada *TCP layer* (biasanya port 389) dari protokol TCP/IP. Karena X.500 merupakan protokol lapisan aplikasi (dalam OSI *layer*) maka ini akan membawa lebih banyak data, seperti *network header* yang dipasang pada paket pada setiap *layer* sebelum akhirnya dikirimkan ke jaringan.

Otentikasi LDAP

Pada otentikasi menggunakan metode otentikasi LDAP terdapat mekanisme *binding*. LDAPv3 mendukung 3 jenis otentikasi yaitu [10]:

1. *Anonymous* : klien yang mengirimkan suatu permintaan ke *server* LDAP tanpa melakukan mekanisme *bind* maka akan diperlakukan sebagai klien anonim dimana klien tersebut hanya dapat melihat tampilan sebagai tamu.
2. *Simple authentication* : mekanisme ini melakukan otentikasi ke *server* LDAP dengan mengirimkan identitas dirinya yang valid berupa *username* dan *password* pada saat *login*. Kemudian melakukan *bind* atau membuka sesi pada koneksi *server* LDAP.
3. *Simple authentication and Security Layer* (SASL) : untuk mengatasi masalah keamanan pada *simple authentiucation*, maka digunakan mekanisme ini dengan cara melakukan enkripsi pada saluran transmisi. Hal ini didukung dengan adanya *Secure Socket Layer* (SSL).

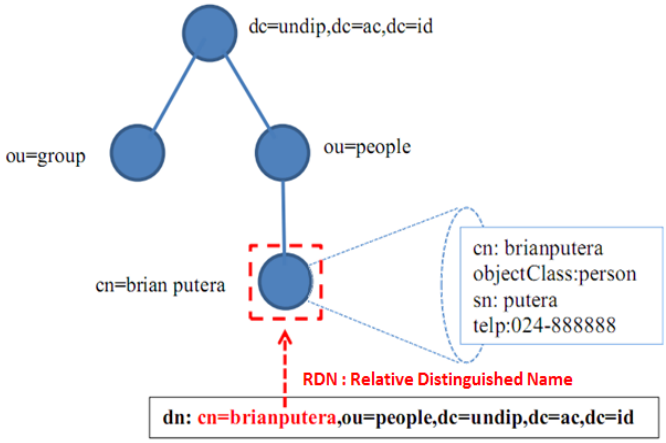
Ruang Lingkup LDAP

Terdapat empat hal yang harus diperhatikan untuk mempelajari LDAP serta dapat mengimplementasikan suatu layanan direktori menggunakan LDAP, empat hal yang dimaksud adalah sebagai berikut [1]:

1. *LDAP Namespace* : Menjelaskan tentang pemberian nama-nama pada layanan LDAP.
2. *LDAP Operation* : Menjelaskan tentang operasi-operasi yang dapat dilakukan dan yang terjadi pada protokol LDAP.
3. *LDAP Schema* : Menjelaskan tentang penggunaan dan penentuan tipe dari informasi yang akan dimasukkan pada sebuah objek.
4. *LDAP Management* : Menjelaskan tentang pembangunan layanan LDAP yang handal seperti kemampuan manajemen entri, replikasi, *referral* dan sebagainya.

Model Penamaan Entri

Model penamaan menentukan bagaimana entri dan data dalam DIT memiliki alamat yang unik. Setiap entri memiliki sebuah atribut yang unik diantara semua atribut yang lain. Keunikan atribut ini dinamakan *Relative Distinguished Name* (RDN). Keunikan suatu atribut dapat dicari dengan mengenali entri dalam sebuah direktori dengan mengikuti RDN dari semua entri dalam jalur dari node yang diinginkan sampai *root* dari *tree*. *String* ini dibuat dengan mengkombinasikan RDN ke dalam bentuk nama yang unik yang disebut node *Distinguished Name (*DN).



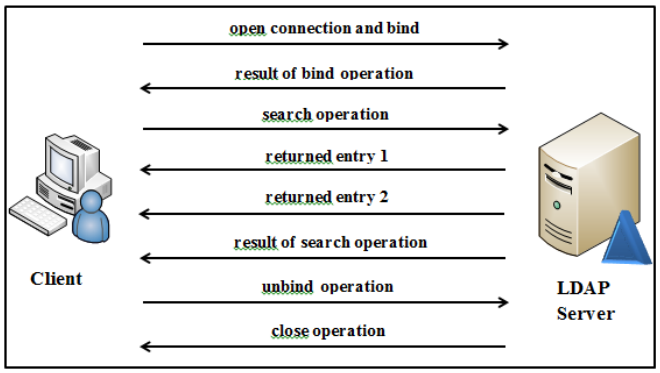
Gambar 2.1 Contoh *tree* dan RDN sebuah entri direktori LDAP

Garis besar entri direktori pada gambar 2.1 diatas mempunyai sebuah RDN yaitu cn=brianputera. Sebagai catatan bahwa nama atribut akan sama dengan nilai yang dimasukkan dalam RDN. DN untuk node ini akan menjadi cn=brianputera,ou=people,dc=undip,dc=ac, dc=id.

Operasi Protokol LDAP

Pada protokol LDAP, terdapat operasi yang dibagi menjadi 3 kategori yaitu:

1. Operasi interogasi
   a. *search* : Merupakan operasi pencarian entri dalam direktori yang sesuai dengan kriteria yang diberikan pada filter pencarian [14]
   b. *compare* : Operasi membandingkan informasi yang diberikan oleh klien dengan informasi yang disimpan di dalam direktori
2. Operasi *update*
   a. *add* : Operasi membuat entri atau atribut baru.
   b. *delete* : Operasi menghapus entri atau atribut.
   c. *modify* : Operasi mengubah atribut dalam sebuah entri.
3. Operasi otentikasi dan kontrol
   a. *bind* : Operasi validasi sebuah entri di dalam direktori
   b. *unbind* : Operasi mengizinkan klien untuk mengakhiri sesi.
   c. *abadon* : Operasi menggagalkan operasi yang sebelumnya dilakukan oleh klien.



Gambar 2.2 Ciri khas pembicaraan protokol LDAP

Hubungan *client-server* biasanya mengikuti pola seperti gambar 2.2, penjelasan dari gambar tersebut adalah sebagai berikut:

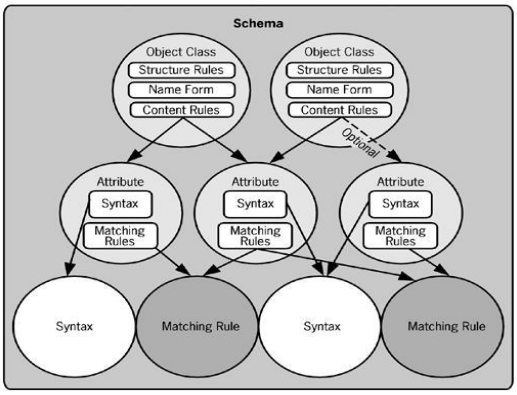
1. Klien berhubungan dengan *server* dan meminta sebuah *bind operation*.
2. *Server* mengembalikan *bind operation* berupa pengembalian kode (sukses atau proses berakhir disini)
3. Klien meminta sebuah *search operation* (atau beberapa operasi yang lain)
4. *Server* mengembalikan pesan dengan menempatkan satu entri atau banyak entri dari *search operation*. Jika tidak ada entri yang ditemukan, tidak ada pesan entri yang dikirimkan.
5. *Server* mengirimkan kode hasil *search operation* pada klien
6. Klien meminta sebuah *unbind operation*
7. *Server* mengirimkan kode hasil *unbind* dan menutup hubungan

Skema LDAP

Skema menentukan aturan yang menguasai sebagian besar dari hal-hal yang dapat dilakukan oleh suatu layanan LDAP. Skema menentukan jenis dari entri yang dapat dibuat dalam direktori. Ini menentukan informasi yang dapat disimpan.

Pengubahan skema untuk mengizinkan sebuah tipe baru dari objek atau sebuah tipe atribut baru. Ini berdampak pada pembuatan tipe baru dari sebuah entri yang dapat ditambah lebih banyak pada fungsi dari direktori itu sendiri.

Skema terdiri dari beberapa komponen. Pada gambar 2.3 ditunjukkan bagaimana setiap elemen dari skema berhubungan dalam konteks dari skema.



Gambar 2.3 Diagram konseptual dari skema

Pada gambar 2.3 menunjukan komponen-komponen yang membangun sebuah skema, komponen-komponen tersebut diantaranya adalah *object class* dan atribut yang menempati tingkat tertinggi dari komponen skema dan selanjutnya adalah sintaks dan *matching rule*.

LDAP Data Interchange Format

LDAP Data Interchange Format (LDIF) merupakan format bahasa untuk merepresentasikan entri pada layanan direktori yang dapat dibaca oleh manusia [2]. LDIF merupakan format teks dan *binary* dan dapat digunakan untuk impor dan ekspor entri pada layanan direktori. Oleh karena format ini berbasis teks maka dapat dengan mudah dibuat dengan menggunakan bermacam-macam editor teks.

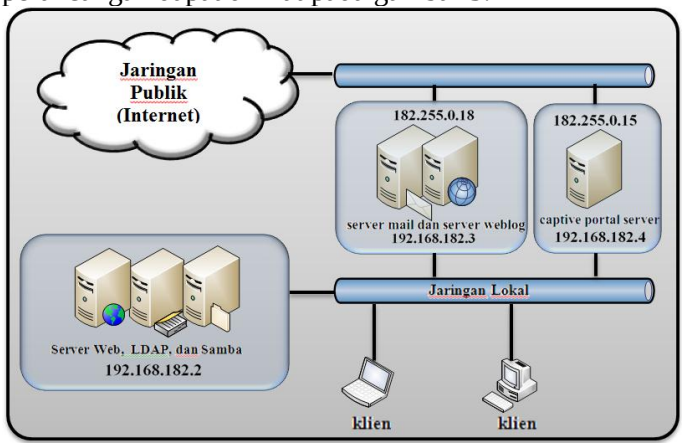
OpenLDAP

OpenLDAP adalah software *open source* yang mengimplementasikan protokol LDAP yang tersedia secara gratis dan terbuka. Terdapat dua buah *server* yang ada dalam paket openLDAP-servers yaitu : *Standalone LDAP Daemon* (slapd) dan *Standalone LDAP Update Replication Daemon* (slurpd). Slapd adalah *standalone LDAP server* sedangkan slurpd. daemon digunakan untuk sinkronisasi perubahan-perubahan dari satu *server* LDAP ke *server* LDAP lainnya dalam suatu jaringan.

III. PERANCANGAN SISTEM

Perancangan ini menggunakan tiga *server* untuk membangun lima layanan yaitu layanan LDAP, samba-PDC, *email*, *weblog*, dan *captive portal*. Layanan LDAP dan samba-PDC akan berada di dalam satu *server* dan layanan *email* dan *weblog* juga akan disatukan dalam satu *server*.

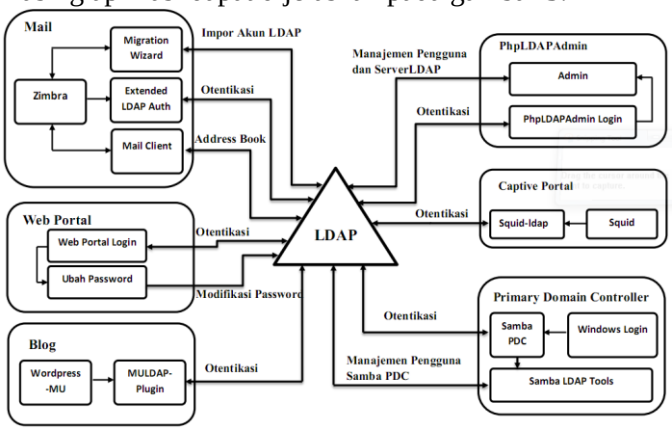
Perancangan ini ditujukan untuk implementasi di dalam jaringan komputer Universitas Diponegoro oleh karena itu penggunaan alamat IP juga akan disesuaikan dengan alokasi alamat IP jaringan komputer UNDIP. *Server email* dan *weblog* akan memiliki dua kartu jaringan yang masing-masing dihubungkan ke jaringan lokal dan jaringan publik. *Server captive portal* juga memiliki dua kartu jaringan yang terhubung ke jaringan lokal dan jaringan publik sedangkan *server* LDAP dan samba-PDC hanya terhubung ke jaringan lokal. Skema perancangan dapat dilihat pada gambar 3.1



Gambar 3.1 Skema perancangan jaringan secara penuh

Perancangan kali ini menempatkan layanan LDAP dan samba-PDC di dalam satu *server*. Hal ini dapat dilakukan karena tidak ada penggunaan layanan yang saling mengganggu satu sama lain. *Server LDAP* diletakan pada jaringan lokal, hal ini dikarenakan layanan samba-PDC dirancang hanya dapat diakses oleh pengguna jaringan lokal. Meskipun demikian, layanan LDAP tetap dapat dimanfaatkan oleh pengguna jaringan *Internet* publik karena akses yang dilakukan ke layanan *email* dan *weblog* dimana kedua *server* tersebut memanfaatkan layanan LDAP yang terdapat pada jaringan lokal. Layanan *email* dan *weblog* memerlukan alamat IP yang terhubung ke jaringan *Internet* publik karena merupakan layanan yang harus dapat diakses dari berbagai tempat.

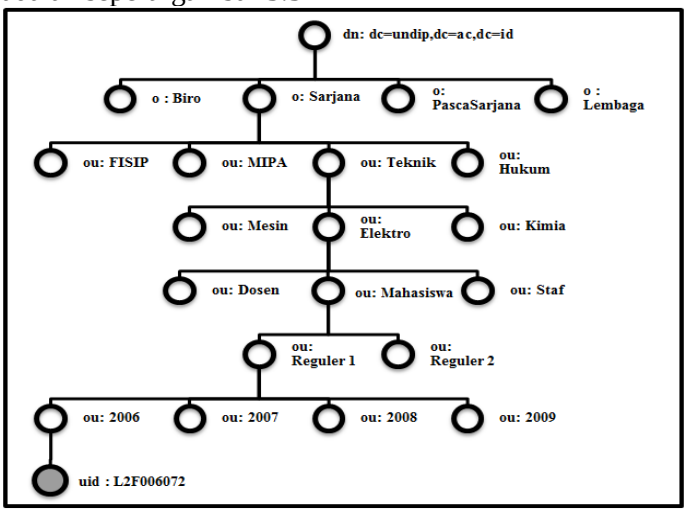
Masing-masing *server* mempunyai layanan dan aplikasi-aplikasi yang berbeda sehingga memungkinkan adanya operasi-operasi yang berbeda. Otentikasi merupakan operasi yang secara umum dilakukan oleh masing-masing aplikasi, akan tetapi terdapat operasi-operasi lain yang dapat dilakukan seperti pencarian *address book*, modifikasi *password*, manajemen *user* dan *server LDAP* dan sebagainya. Skema operasi dari masing-masing aplikasi dapat dijelaskan pada gambar 3.2



Gambar 3.2 Skema operasi terhadap *server* LDAP

Namespace Design

Perancangan *namespace* harus menghasilkan tingkatan yang berbentuk hirarki. Dengan satu *Domain Component* yaitu dc=undip,dc=ac,dc=id dan tingkatan selanjutnya sesuai dengan klasifikasi seperti yang ditemukan di dalam Universitas yaitu dimulai dengan Fakultas, Program Studi, Jurusan dan seterusnya. Gambaran dari *Directory Information Tree* untuk *namespace* yang dirancang adalah seperti gambar 3.3



Gambar 3.3 Skema DIT perancangan *namespace*

Pada gambar 3.3 hanya merepresentasikan letak *user* yang berada pada hirarki tertentu. Dalam hal ini *user* yang berperan sebagai mahasiswa angkatan 2006 program Reguler1 Teknik Elektro Universitas Diponegoro. Dengan demikian dapat dibuat format penulisan *Distinguished Name* yaitu sebagai berikut:

```
uid=UserID,ou=Angkatan,ou=Program-
Studi,ou=Jabatan,ou=Jurusan,ou=Fakultas,o=Str
uktur,dc=undip,dc=ac,dc=id
```

Format pelunisan di atas dapat dijelaskan pada tabel 3.1 berikut:

Tabel 3.1 Tabel format penulisan *Distinguished Name*

Tingkatan DN	**Keterangan**	**Contoh**
uid=UserID	Indentitas Pengguna yang bersifat unik (NIM/NIP)	uid=L2F006072
ou=Angkatan	Tahun Masuk Mahasiswa	ou=2006
ou=Program-Studi	Jenis program studi pengguna	ou=Reguler1
ou=Jabatan	Posisi pengguna dalam Universitas	ou=Mahasiswa
ou=Jurusan	Jurusan pengguna	ou=Elektro
ou=Fakultas	Fakultas pengguna	ou=Teknik
o=Organisasi	Jenis Organisasi	o=Sarjana
dc=undip,dc=ac, dc=id	*Domain Component*	-

Schema Design

Pada peracangan skema, akan menententukan informasi yang dapat disimpan oleh suatu entri. Hal ini berkaitan dengan penggunaan *ObjectClass* dan pemilihan atribut yang disediakan oleh *objectClass* tersebut. Atribut seperti alamat *email* disediakan oleh *objectclass inetOrgPerson*, oleh karena itu *objectClass* tersebut harus digunakan di dalam *server* LDAP.

Selain *objectClass* yang berada pada skema yang sudah dimiliki oleh openLDAP, beberapa *objectClass* seperti *sambaSamAccount* dibutuhkan untuk mendukung layanan samba-PDC. Oleh karena itu perlu ditambahkan skema yang disediakan oleh samba di dalam openLDAP karena openLDAP tidak mempunyai skema yang terdapat *objectClass* tersebut. Berikut ini merupakan daftar *objectClass* dan atribut yang digunakan untuk masing-masing layanan.

Tabel 3.2 Tabel penggunaan *objectclass*

Nama ObjectClass	**Nama Layanan**
inetOrgPerson	Email Weblog Samba-PDC Web portal Captive portal
shadowAccount	Samba-PDC
posixAccount	Samba-PDC
sambaSamAccount	Samba-PDC
undipAccount	Web Portal

IV. IMPLEMENTASI DAN PENGUJIAN

Implementasi dan Pengujian *server* LDAP

Implementasi dilakukan dengan membuat layanan-layanan seperti pada perancangan. Dimulai dengan penanaman sistem operasi pada masing-masing *server* sampai dengan instalasi aplikasi-aplikasi yang diperlukan oleh masing-masing *server*. Setelah semua paket telah terpasang, maka hal terpenting yang harus dilakukan adalah melakukan konfigurasi agar semua proses otentikasi dari semua layanan menggunakan otentikasi yang disediakan oleh *server* LDAP. Tabel 4.1 berikut merupakan konfigurasi-konfigurasi yang dilakukan pada masing-masing layanan khususnya untuk melakukan integrasi otentikasi dengan *server* LDAP.

Tabel 4.1 Tabel konfigurasi pada masing-masing aplikasi

Layanan	**Aplikasi/paket**	**Konfigurasi**
LDAP	OpenLDAP	/slapd.conf /openldap/schema/ /etc/init.d/slapd
	BerkeleyDB	/usr/local/lib/openldap-data/
	Apache	/etc/apache2/
	Php5-LDAP	-
	PhpLDAPadmin	/phpldapadmin/config
	OpenSSL	/sites-avaliable/ssl
Samba-PDC	Samba-server	/smb.conf /home/samba/ /logon.bat
	Samba-client	-
	SambaLDAPtools	/etc/smbldaptools/ /usr/sbin/
	PAM-LDAP	/etc/ldap.conf /profile.d/open_ldap
Email	Zimbra Collaboration Suite	External GAL Migration Wizard External LDAP auth
Weblog	Apache	/httpd.conf
	Php5-LDAP	-
	MySQL	database studentsblog
	Wordpress-MU	wp-config.php /.httaccess
	WPMU-LDAP	LDAP Option
Captive portal	Squid	/squid.conf
	LDAP-Squid	-
	Group-LDAP-Squid	-
Web portal	Apache2	/etc/apache2/
	Php5-LDAP	ldap_auth.php password.php
	OpenSSL	/sites-avaliable/ssl

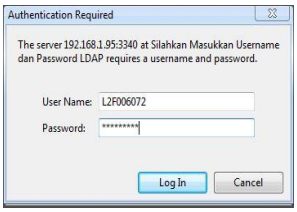
Pengujian otentikasi dilakukan pada halaman/jendela *login* yang disediakan oleh setiap layanan. Terdapat enam buah halaman/jendela *login* pada pengujian ini dimana akan meminta informasi berupa *username* dan *password* seperti pada gambar-gambar dibawah ini.

(a)

(b)

(c)

(d)



(e)

(f)

Gambar 4.1 (a) Halaman login phpLDAPadmin
(b) Jendela login Windows XP
(c) Halaman login Zimbra email client
(d) Halaman login Wordpress-MU
(e) Jendela login web browser
(f) Halaman login web portal

Masing-masing halaman/jendela *login* seperti yang terlihat pada gambar-gambar di atas memerlukan masukan informasi *username* dan *password* kecuali pada gambar 4.1(a) dimana memerlukan informasi *distinguished name*. Otentikasi LDAP seharusnya membutuhkan informasi *distinguished name* dan password pada operasi *bind*. Dengan kata lain aplikasi phpLDAPadmin tidak memerlukan *filter* pencarian pada proses otentikasinya. Berikut ini merupakan tabel dari informasi yang dibutuhkan dan atribut yang diperlukan atau didapatkan pada saat proses *login* berhasil.

Tabel 4.2 Filter dan pemetaan atribut masing-masing aplikasi

Aplikasi	**Filter/atribut**	***Attribute mapping***
PhpLDAPadmin	dn userPassword	Seluruh Attribute dalam LDAP
Windows XP Login	uid Sambantpassword	Homedirectory Gecos
Zimbra email client	uid userPassword	cn - Nama mail – Alamat Email ou - grup
Wordpress Login	uid userPassword	uid - username cn – Nama mail – Alamat Email
Web browser	uid userPassword	ou – grup
Web Portal	uid userPassword	NIM/NIP cn - nama dn - dn mail – alamat email Jurusan – Nama Jurusan Jabatan – Jabatan user ou - grup

Pembuatan Entri LDAP

Terdapat dua cara untuk membuat sebuah akun atau entri di dalam direktori, kedua cara tersebut yaitu menggunakan menggunakan fitur pembuatan entri pada aplikasi phpLDAPadmin dan eksekusi berkas LDIF. Menggunakan berkas LDIF merupakan cara terbaik untuk membuat akun secara massal. Universitas Diponegoro menyimpan informasi *user* khususnya mahasiswa di dalam basis data Sistem Informasi Akademik yang menggunakan MySQL sebagai aplikasi basis data. Oleh karena itu dibuatlah aplikasi MySQL-LDIF pada tugas akhir ini.

Aplikasi MySQL-LDIF dikembangkan menggunakan bahasa pemrograman php yang ditujukan untuk mensiasati pembuatan akun dari basisdata MySQL ke dalam direktori *server* LDAP. Karena skema dan format penyimpanan pada kedua aplikasi berbeda maka untuk membuat akun pada *server* LDAP maka harus dibuat berkas LDIF dengan berdasarkan pada data yang ada di dalam MySQL. Hasil keluaran dari aplikasi ini akan disalin ke aplikasi phpLDAPadmin untuk dieksekusi.

Pada gambar 4.2 merupakan contoh tabel basis data Sistem Informasi Akademik yang menyimpan informasi nama, nim dan *password*. Tabel ini kemudian akan diakses oleh aplikasi MySQL-LDIF untuk diubah menjadi berkas teks format LDIF.

NAMA	NIM	PASS
Rian Aditia	L2F006077	95e7ff88cb7fd02ec7dc3837daeac62e
Haryo Punto Susilo	L2F006048	87c5470ea32fe7d0b1a59bf1e3570855
Anton Ratrianto	L2F006010	d1a6f5b3d9f67ae4977d8024c443c00e
Primanda Adhi Putera	L2F006070	c3a432ab73fd60d721a19d64dc43efa7
Darmawan Surya	L2F006027	d6ab036968ba5788ab7883d830e78c6c
Chandra Efendi	L2F006025	056fb91e4cf6540e264afc2beebe51d4
Dwi Ardhanto	L2F006032	df93d68002e63f79f60e3f0a3976e8a2
Causa Prima	L2F006030	8912f2704407226566fa1182cd7f773d
Joko	L2F006055	3a59f42c0807183f6d9f2df28a7009b2
Ardian	L2F006004	841e711445a0d6447f3ffcf973992e8b

Gambar 4.2 Tabel basis data *user*

Aplikasi ini akan menjadikan NIM sebagai *username* dan bagian dari *distinguished name* karena NIM merupakan suatu atribut yang bersifat unik yang dapat mewakili sebuah *user*. *Password* yang ada di dalam basisdata mempunyai format enkripsi MD5 akan disandikan dengan *base64encode* agar dapat dikenali oleh *server* LDAP. Berikut ini merupakan contoh dari satu buah entri yang dihasilkan oleh aplikasi ini dengan format LDIF.

```
dn:uid=L2F006010,ou=2006,ou=Reguler1,ou=Mahasis
wa,ou=Elektro,ou=Teknik,o=Sarjana,dc=undip,
dc=ac,dc=id
cn: Anton Ratrianto
sn: L2F006010
uid: L2F006010
NIM: L2F006010
mail: L2F006010@students-mail.undip.ac.id
objectClass: InetOrgPerson
objectClass: undipAccount
ou: Elektro
ou: Mahasiswa
Jurusan : Elektro
Jabatan : Mahasiswa
userPassword: {MD5}0ab1s9n2euSXfYAkxEPADg==
```

Hasil dari keluaran berformat LDIF akan di salin ke halaman impor aplikasi phpLDAPadmin. Perlu dipastikan bahwa sebuah entri pada format LDIF harus dipisahkan dengan satu baris, apabila ada dua atau lebih entri yang tidak dipisahkan oleh satu baris maka akan dianggap satu entri dan akan terjadi kesalahan pada proses impor ini.

```
Adding uid=L2F006077,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006048,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006010,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006070,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006027,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006025,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006032,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006030,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006055,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
Adding uid=L2F006004,ou=Mahasiswa,ou=Elektro,ou=Reguler1,o=Teknik,dc=undip,dc=ac,dc=id Success
```

Gambar 4.3 Hasil pembuatan entri phpLDAPadmin

Berdasarkan gambar 4.3 dapat dilihat bahwa pembuatan entri berhasil dilakukan. Pada layanan *email* perlu digunakan fitur *migration wizard* untuk membuat *mailbox* berdasarkan entri atau akun yang ada di dalam direktori. Sedangkan untuk akun samba-PDC diperlukan aplikasi berbasis *command line* yaitu *smbldap-tools*. Hal ini disebabkan berkas LDIF hanya dapat membuat sebuah entri di dalam direktori akan tetapi tidak dapat membuat sebuah direktori di dalam sistem operasi Linux.

PENUTUP

Kesimpulan

Dari hasil impelentasi dan pengujian dapat disimpulkan bahwa:

1. Sistem otentikasi terpusat yang menggunakan *server* LDAP dapat berjalan dengan baik, ditandai dengan keberhasilan otentikasi pada layanan samba-PDC, *captive portal*, *weblog*, *email*, dan web portal.
2. OpenLDAP membutuhkan aplikasi berbasis GUI (*Graphical User Interface*) untuk memudahkan dalam hal manajemen entri oleh pihak administrator.
3. Proses otentikasi menggunakan *server* LDAP dilakukan dengan operasi *bind* dan *search*. Operasi *bind* dilakukan pada saat pemanggilan dan validasi identitas dan operasi *search* dilakukan pada saat penggambilan nilai atribut yang diinginkan dari sebuah entri.
4. Rancangan skema dan *namespace* harus disesuaikan dengan lingkungan implementasi dan informasi yang dibutuhkan. Pada tugas akhir ini rancangan skema dan *namespace* disesuaikan dengan lingkungan Universitas Diponegoro.
5. Otentikasi pada layanan *email* membutuhkan informasi alamat *email*, *username*, *password* yang berada di *server* LDAP serta *mailbox* yang terdaftar di *server email*.
6. Otentikasi pada layanan *weblog* dan *captive portal* membutuhkan informasi *username*, *password* yang terdapat di dalam *server* LDAP.
7. Otentikasi pada layanan samba-PDC tidak membutuhkan informasi dari atribut *password* melainkan dari atribut *sambantpassword*, oleh karena itu diperlukan sinkronikasi antara kedua atribut tersebut.
8. Penggantian *password* oleh *user* sebagai atribut yang digunakan dalam proses otentikasi dilakukan disatu aplikasi yaitu web portal untuk memastikan pengelolaan akun secara terpusat.

Saran

Adapun saran yang dapat diberikan untuk menjadi masukan pada penelitian lebih lanjut adalah:

1. Layanan LDAP yang telah dibangun dapat ditambahakan fungsi replikasi untuk sinkronisasi data dan menjaga ketersediaan data.
2. Pengembang perangkat lunak dapat mengembangkan aplikasi yang memanfaatkan sistem otentikasi yang disediakan oleh *server* LDAP ini.
3. Sistem otentikasi terpusat dapat dikembangkan menjadi *sistem single sign on*.
4. Penelitian terhadap keamanan dan kinerja sistem otentikasi terpusat berbasis LDAP dapat dilakukan dengan memanfaatkan implementasi yang telah dilakukan.

DAFTAR PUSTAKA

[1] Arkills, B., *LDAP Directories Explained: An Introduction and Analysis*, Addison Wesley, Boston, MA 02116, U.S.A., 2003.

[2] Carter, G., *LDAP System Administration*, O'Reilly, 1005 Gravenstein Highway North Sebastopol, CA 95472, U.S.A., 2003.

[3] Donley, C., *LDAP Programming Managemen*. Manning, Greenwich, CT 06830, U.K., 2003.

[4] Jackiewicz, T., *Deploying OpenLDAP*, Apress, New York, U.S.A., 2003

[5] Jatnika, D., *Layanan Sistem Direktori Dengan Menggunakan LDAP*, Makalah Kemanan Sistem Lanjut, Istitut Teknologi Bandung, 2004.

[6] Liang, J., V. Vaishnavi, A. Vandenberg., "Clustering of LDAP Directory Schema fo Facilitate Information Resource Interoperability Across Organizations", IEEE Computer Society, July 2006.

[7] Nurdeni, D.A., *Implementasi Teknologi Single Sign On di Lingkungan Teknik Informatika ITS*, Skripsi S-1, Institut Teknologi Sepuluh November, 2009.

[8] Pradnyana, I. W., *Analisis dan Perancangan Arsitektur Sistem Otentikasi Terintegrasi Antara Platform Linux, Windows 2000, dan Novell Netware : Studi Kasus Jurusan Teknik Informatika FTIF ITS*, Skripsi S-1, Institut Teknologi Sepuluh November, 2004.

[9] Rudy, Riechie, dan O. Gunadi., *Integrasi Aplikasi Menggunakan Single Sign On Berbasis Lightweight Directory Access Protocol (LDAP) Dalam Portal BINUS@CCESS(BEE-PORTAL)*, Skripsi S-1, Universitas Bina Nusantara, 2009.

[10] Sari, R.F. dan H. Syarif, "Integrasi Mekanisme Autentikasi Aplikasi Web Server Dengan Metode LDAP : Studi Kasus Aplikasi SIPEG UI" , Jurnal Teknologi FTUI, 2006.

[11] Shengli, L., W. Wenbing and Z. Yuefei. "A New-Style Domain Integrating Management of Windows and UNIX" IEEE Computer Society. August 2008.

[12] T. Howes, M. Smith, and G. S. Good, *Understanding and Deploying LDAP Directory Services*, 2nd ed., Addison Wesley Professional, 2003

[13] Voglmaier, R., *The ABCs of LDAP: How to Install, Run, and Administer LDAP Services*, Auerbach Publications, Boca Raton, Florida, 2003.

[14] Xin, W., S. Henning, K. Dilip and V. Dinesh. "Measurement and Analysis of LDAP Performance" IEEE Computer Society, February 2008

[15] Yang, C.S., C.Y., J.H Chen and C.Y. Sung. "Design and Implementation of Secure Web-based LDAP Managemenent System", IEEE Computer Society, August 2002.

[16] ---, *OpenLDAP Software 2.4 Administator's Guide*, http://openldap.org, Maret 2011.

BIODATA

R Fibrian Satya Putera, lahir di Manado tanggal 12 Februari 1989. Menempuh pendidikan dasar di SD Negeri 11 Manado dan SD Angkasa 9 Jakarta. Melanjutkan ke SLTP N 81 Jakarta dan SLTP N 6 Makassar, Dan Pendidikan tigkat atas di SMU N 7 Manado dan SMU N 91 Jakarta, lulus tahun 2006. Dari tahun 2006 sampai saat ini masih menempuh studi Strata-1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, konsentrasi Komputer dan Informatika.

Menyetujui,

Dosen Pembimbing I

Adian Fatchur Rochim, S.T.,M.T.

NIP. 197302261998021001

Dosen Pembimbing II

Yuli Christyono, S.T.,M.T.

NIP. 196807111997021001