

# **LEMBAR PENGESAHAN**

Batam, 27 Juli 2010

**Pembimbing,**

**Evaliata Sembiring, S.Kom**

**NIK 106042**

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310701027

Nama : Rini Irmanti

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

### IMPLEMENTASI SINGLE SIGN ON

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 27 Juli 2010

**Rini Irmanti**  
3310701027

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310701083

Nama : Nofriani

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

### IMPLEMENTASI SINGLE SIGN ON

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 27 Juli 2010

**Nofriani**  
3310701083

## KATA PENGANTAR

Puji dan syukur kehadiran Allah swt atas berkat rahmat, hidayat serta bimbingan-Nya, penyusun dapat menyelesaikan Tugas Akhir yang berjudul Implementasi Single Sign On. Aplikasi ini bertujuan untuk mempermudah dalam melakukan menyelesaikan masalah atau pekerjaan dalam mengelola koperasi polibatam dengan baik dan mudah.

Dalam kesempatan ini, penyusun mengucapkan terima kasih kepada:

1. Bapak Uuf Brajawidagda ST, Msc selaku koordinator tugas akhir.
2. Ibu Evaliata Sembiring S.Kom, selaku dosen pembimbing tugas akhir.
3. Bapak/Ibu dosen program studi Teknik Informatika atas bimbingannya.
4. Kedua orang tua tercinta yang memberikan dukungan moral, doa dan semangat.
5. Teman-teman seperjuangan IF 2007 yang memberikan dukungan semangat dan doa.

Dalam penulisan ini, penyusun mengakui bahwa masih terdapat kekurangan-kekurangan dalam penyusunannya. Oleh karena itu, penyusun sangat mengharapkan bantuan dari berbagai pihak berupa kritik ataupun saran guna penyempurnaan selanjutnya. Akhir kata penyusun ucapkan terima kasih, semoga laporan ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah aplikasi yang serupa.

Batam, Juli 2010

Penulis

## ABSTRAK

### IMPLEMENTASI SINGLE SIGN ON

Tugas Akhir ini membahas tentang pengimplementasian *metode Single Sign On (SSO)* dengan menggunakan PHP OpenLDAP dan *Lightweight Data Access Protocol (LDAP)* di dalam sebuah web portal. Tujuan utama dari pengimplementasian SSO ini adalah untuk menggabungkan aplikasi yang ada ke dalam sebuah *site* sehingga terbentuk integrasi aplikasi, khususnya dalam bentuk web yang biasa disebut dengan Web Portal. Dengan adanya Web Portal yang menggunakan *metode Single Sign On (SSO)* ini, setiap *user* hanya perlu memiliki satu *username*, satu *password*. Dan bila ingin mendapatkan layanan atau fasilitas di Web Portal, *user* ini hanya perlu *login* satu kali saja bisa dapat menggunakan semua fasilitas atau layanan aplikasi yang ada di dalam Web Portal tersebut. Hal ini dapat mempermudah *user* dalam menggunakan aplikasi yang ada. *User* tidak perlu menghafal banyak *account*, hanya satu *account* dan tidak perlu berulang kali *login*, cukup dengan sekali *login*. Hal ini juga dapat mempermudah dalam pengorganisasian data *user* yang ada, sehingga keamanan data *user* lebih terjamin, karena menggunakan tempat penyimpanan data *user* yang terpusat. PHP OpenLDAP digunakan untuk menangani masalah komunikasi antara aplikasi web yang berbeda, sehingga semua aplikasi dapat diintegrasikan ke dalam sebuah Web Portal. LDAP digunakan sebagai sebuah protokol direktori servis, dimana semua data *user* disimpan di dalam LDAP.

Kata Kunci: aplikasi web, portal, SSO, PHP openLDAP, LDAP.

## **ABSTRACT**

### **SINGLE SIGN ON IMPLEMENTATION**

This final assignments discuss about implementation of single sign on use method PHP OpenLDAP and Lightweight Data Access Protocol (LDAP) in a portal web. The purpose of SSO implementation is to bundle some application web in a site, so was built application integration. Especially a web is known as portal web. With portal web use single sign on method, every user only need a username and a password. and if user want get services or facilities in portal web, user only once in a login to use all of the facilities or services in that portal web. This case can facilitate the user to use the application. User doesn't need to remember some account, only one account and doesn't need login repeatedly, sufficient with once login. This case can easier for data user organizing at hand. So the data user security more safety, because this application use centrally user database. PHP OpenLDAP is used to handle communication problem among differently web application, so all of the application can be integrated in a portal web. LDAP is used like a protocol of service directory, where all of user data are saved in the LDAP.

Key words: web application, portal, SSO, PHP OpenLDAP, LDAP.

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN .....	ii
KATA PENGANTAR .....	iv
ABSTRAK .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	ix
DAFTAR ISI LAMPIRAN .....	x
Bab I Pendahuluan .....	1
I.1 Latar Belakang .....	1
I.2 Rumusan Masalah.....	2
I.3 Batasan Masalah.....	2
I.4 Tujuan.....	2
I.5 Sistematika Penulisan.....	3
Bab II Landasan Teori.....	4
II.1 Internet.....	4
II.1.1 Pengertian Internet .....	4
II.1.2 World Wide Web (WWW).....	4
II.1.3 Web.....	4
II.1.4 Web Portal.....	5
II.2 Lightweight Directory Access Protocol (LDAP) .....	5
II.2.1 Pengertian LDAP.....	5
II.2.2 Organisasi Data LDAP .....	7
II.2.3 Directory security.....	9
II.3 Single Sign On (SSO).....	9
II.3.1 Pengertian Single Sign On (SSO) .....	9

II.3.2	PHP OpenLDAP.....	10
II.3.3	Implementasi Single Sign On (SSO) .....	11
II.3.4	Implementasi Single Sign On (SSO) Universitas Bina Nusantara.....	11
II.4	Web Lembaga Amil Zakat (LAZ) Masjid Raya Batam .....	12
II.5	Web IF Politeknik Batam.....	13
Bab III	Analisis.....	14
III.1	Deskripsi Umum Proses Single Sign On (SSO).....	14
III.2	Proses Sistem Single Sign On (SSO).....	15
III.3	Kebutuhan Data.....	17
III.4	Activity Diagram.....	19
III.5	Use Case Diagram.....	20
III.5.1	Skenario Use Case Diagram .....	21
Bab IV	Perancangan.....	24
IV.1	Perancangan Proses Sistem Setelah SSO .....	24
IV.1.1	Perancangan Proses Registrasi user .....	25
IV.1.2	Perancangan Proses User Login .....	25
IV.1.3	Proses pada Web Lembaga Amil Zakat Masjid Raya Batam.....	26
IV.1.4	Proses pada Web IF Politeknik Batam.....	28
IV.1.5	Perancangan Proses Edit Registration Info .....	29
IV.1.6	Perancangan Antarmuka .....	30
Bab V	Implementasi dan Pengujian.....	39
V.1	Spesifikasi Perangkat Keras dan Perangkat Lunak.....	39
V.2	Instalasi dan Konfigurasi.....	39
V.2.1	Instalasi dan Konfigurasi Server LDAP.....	39
V.2.2	Instalasi dan Konfigurasi PHP OpenLDAP .....	40
Bab VI	Kesimpulan dan Saran .....	41
VI.1	Kesimpulan.....	41

VI.2 Saran.....	41
DAFTAR PUSTAKA .....	40

## DAFTAR GAMBAR

Gambar 1 X.500 dengan OSI Vs LDAP dengan TCP/IP .....	6
Gambar 2 Hubungan antara LDAP client, server, dan data storage .....	6
Gambar 3 Contoh Hirarki Pohon LDAP .....	8
Gambar 4 Perbedaan sebelum dan sesudah SSO .....	10
Gambar 5 Deskripsi Umum Sistem Proses SSO.....	14
Gambar 6 Proses Sebelum Single Sign On (SSO) .....	16
Gambar 7 Proses Setelah Single Sign On (SSO) .....	17
Gambar 8 Activity Diagram .....	19
Gambar 9 Use Case untuk Web Portal.....	20
Gambar 10 Perancangan Proses Sistem Setelah SSO.....	24
Gambar 11 Diagram Alir Proses Registrasi User .....	25
Gambar 12 Diagram Alir untuk Login.....	25
Gambar 13 Diagram alir untuk web LAZ Masjid Raya Batam .....	26
Gambar 14 Diagram alir untuk web IF Politeknik Batam .....	28
Gambar 15 Diagram alir untuk Edit Registration Info .....	29
Gambar 16 Tampilan Utama Web Portal .....	30
Gambar 17 Tampilan Registrasi User Baru.....	31
Gambar 18 Tampilan Halaman Login Single Sign On (SSO).....	34
Gambar 19 Tampilan Edit Registration Info .....	36

## DAFTAR TABEL

Tabel 1 Data User pada direktori LDAP.....	17
--	----

# Bab I Pendahuluan

## I.1 Latar Belakang

Dewasa ini perkembangan dunia informasi semakin pesat. Hal ini sangat berpengaruh pada perkembangan internet. Internet berkembang menjadi media yang kuat untuk komunikasi secara *global*. Web portal merupakan sebuah teknologi yang akan berkembang pada teknologi web di masa depan. Informasi web portal dapat menggabungkan berbagai aplikasi web menjadi satu kesatuan. Fasilitas integrasi semua *server* dengan satu *username* dan satu *password* dinamakan *Single Sign On (SSO)*.

*Single Sign On (SSO)* memungkinkan seorang *user* hanya membutuhkan satu *username* dan satu *password* untuk masuk ke beberapa aplikasi. Penggunaan metode SSO ini dimaksudkan untuk membuat suatu integrasi layanan yang ada. Sebagai contoh pengimplementasian metode SSO adalah penggunaan satu *login* untuk mengakses web portal Universitas Bina Nusantara yang berisi beberapa aplikasi diantaranya, Blog, Forum, IPTV, *Messenger*, *SoftPhone* dan *Sharing Content*.

Metode SSO ini tidak lepas dari penggunaan PHP OpenLDAP dan LDAP. OpenLDAP adalah salah satu *framework* yang menyediakan layanan SSO. Sedangkan LDAP menyediakan berbagai layanan seperti menyediakan data untuk *client* contohnya *address book*, *email*, *account login*, dan masih banyak data lainnya.

Tugas Akhir ini akan membahas tentang usaha penerapan dan implementasi metode *Single Sign On (SSO)* dalam beberapa aplikasi web dengan menggunakan PHP OpenLDAP dan *Lightweight Data Access Protocol (LDAP)*. Dalam hal ini contoh yang digunakan adalah aplikasi web lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam, karena kedua aplikasi ini telah dibangun sebelumnya oleh mahasiswa yang magang di Politeknik Batam dan merupakan aplikasi berbasis web.

## **I.2 Rumusan Masalah**

Pokok permasalahan mengacu pada penerapan *Single Sign On (SSO)* pada aplikasi web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam, agar program atau aplikasi tersebut dapat diakses hanya dengan menggunakan satu akun saja.

## **I.3 Batasan Masalah**

Batasan masalah dalam penelitian ini adalah:

1. Pembuatan aplikasi web portal dengan contoh aplikasi web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam.
2. Penerapan metode *Single Sign On (SSO)* dalam melakukan *login* dengan menggunakan *Lightweight Directory Access Protokol (LDAP)* dan PHP OpenLDAP.

## **I.4 Tujuan**

Tujuan penelitian ini adalah

1. Mengimplementasikan metode *Single Sign On (SSO)* dengan menggunakan *Lightweight Directory Access Protokol (LDAP)* dan *PHP OpenLDAP* sehingga terbentuk suatu integrasi *login* seorang *user* agar dapat mengakses beberapa situs web dalam suatu web portal menggunakan satu *username* dan satu *password*.
2. Mengetahui jenis aplikasi dan pengimplementasian metode *Single Sign On (SSO)*.

## **I.5 Sistematika Penulisan**

**Bab 1** Pendahuluan berisi penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan buku.

**Bab 2** Landasan Teori berisi penjelasan mengenai teori-teori yang berhubungan dengan *Single Sign On (SSO)*.

**Bab 3** Analisis yang berisi tentang proses sistem berjalan, kebutuhan data, deskripsi umum proses SSO, *activity diagram*, dan *use case diagram*.

**Bab 4** Perancangan yang berisi tentang perancangan proses sistem.

**Bab 5** Implementasi dan Pengujian yang berisi tentang proses instalasi dan konfigurasi aplikasi.

**Bab 6** Kesimpulan dan Saran yang berisi tentang kesimpulan dari hasil pembuatan aplikasi dan saran untuk perbaikan dan pengembangan.

## **Bab II      Landasan Teori**

### **II.1    Internet**

#### **II.1.1    Pengertian Internet**

Secara umum internet adalah sebuah kumpulan jaringan yang terdiri dari banyak manusia, mesin, *program software*, dan data, yang memiliki jumlah sangat besar, tersebar di seluruh dunia, dan terus-menerus berhubungan.

#### **II.1.2    World Wide Web (WWW)**

Menurut Turban et al (2005), menjelaskan bahwa *WWW* tidak serupa dengan *Internet*. Fungsi *Internet* adalah sebagai mekanisme *transport*. Sedangkan *WWW* adalah sebuah aplikasi yang menggunakan fungsi *transport* itu. Aplikasi lain juga dapat berjalan dalam *Internet*.

#### **II.1.3    Web**

*Web* merupakan sebuah sistem dengan standar universal yang dapat diterima untuk penyimpanan, pengambilan, pembuatan/pembentukan, dan menampilkan informasi lewat arsitektur *client/server*. *Web* menangani semua tipe informasi digital, termasuk teks, hypermedia, grafik, dan suara. *Web* menggunakan *user interface* berupa grafik. *Web* yang berdasarkan standar bahasa *hypertext* dinamakan *Hypertext Markup Language (HTML)*, dimana format dokumen dan penggabungan *link hypertext* yang dinamik untuk dokumen lain disimpan pada komputer yang sama atau berbeda.

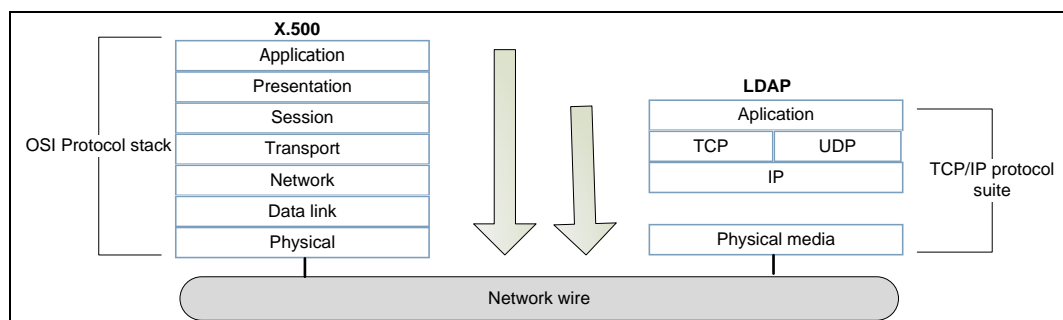
## II.1.4 Web Portal

Web Portal merupakan sebuah teknologi yang akan berkembang pada teknologi web di masa depan. Satu halaman portal terdiri dari berbagai macam *portlets* yang dapat mengirimkan informasi dari banyak. Selain informasi web portal juga dapat menggabungkan berbagai aplikasi web menjadi satu kesatuan, sebagai contoh adalah *sharing content, digital cinema, e-learning, dll* (Mannaert et al, 2003).

## II.2 Lightweight Directory Access Protocol (LDAP)

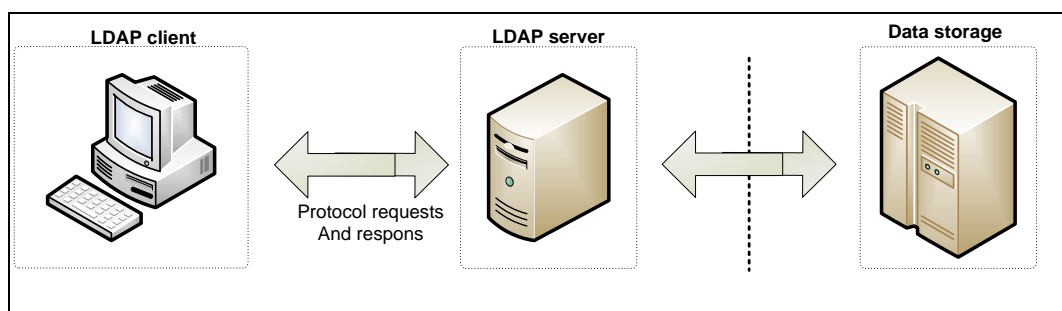
### II.2.1 Pengertian LDAP

LDAP merupakan singkatan dari *Lightweight Directory Access Protocol* (Protokol Akses Direktori Ringan) yang artinya adalah sebuah protokol kelas ringan untuk mengakses *Directory Service* dalam jaringan, baik internet ataupun intranet berdasarkan pada protokol servis direktori X.500. LDAP berjalan melalui protokol *TCP/IP*. LDAP dikatakan ringan jika dibandingkan dengan X.500 servis direktori. X.500 mendapatkan gelar "*Heavyweight*" karena membutuhkan *client* dan *server* untuk berkomunikasi menggunakan *Open System Interface (OSI)* protokol. Tujuh layer OSI ini bagus dalam mengaplikasikan jaringan *protocol suite*, tetapi ketika dibandingkan dengan *TCP/IP protocol suite*, ini serupa dengan bepergian jauh dengan dengan barang bawaan yang sangat berat.



**Gambar 1 X.500 dengan OSI Vs LDAP dengan TCP/IP**

Gambar 1 merupakan perbandingan antara servis direktori X.500 dengan LDAP. LDAP dibuat berdasarkan X.500, LDAP memiliki hampir semua fungsi-fungsi utama X.500. X.500 adalah model untuk Layanan Direktori dari konsep OSI. X.500 berisi definisi nama dan protokol untuk menanyakan (*query*) dan memperbarui direktori. Tetapi ditemukan bahwa solusi X.500 terlalu berlebihan dalam banyak kasus. Sekarang LDAP Seperti X.500 juga yang berisi model data/nama untuk direktori dan protokol juga. Tetapi LDAP didisain untuk berjalan langsung pada lingkungan TCP/IP. Jadi LDAP merupakan versi ramping dari X.500.



**Gambar 2 Hubungan antara LDAP client, server, dan data storage**

Gambar 2 merupakan gambaran LDAP. LDAP menggunakan model *client-server*, dimana *client* 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 *client*. LDAP *server* dapat digunakan sebagai *backend storage* untuk web *server*. Semua HTML dan file grafik dapat disimpan dalam direktori dan dapat dipertanyakan (*query*) oleh banyak web *server*. Disamping semuanya, sebuah web *server* biasanya hanya membaca *file* dan mengirimkannya ke *client*, *file* itu sendiri tidak akan sering diubah.

LDAP tersebut memiliki bentuk struktur yang berhirarki, bukannya berformat kolom dan baris, seperti halnya *database* normal, sehingga memudahkan untuk memasukkan

sejumlah besar detail yang mirip dalam bentuk yang terorganisir. LDAP *client-server* memperkecil proses pengeluaran dari *client*.

## II.2.2 Organisasi Data LDAP

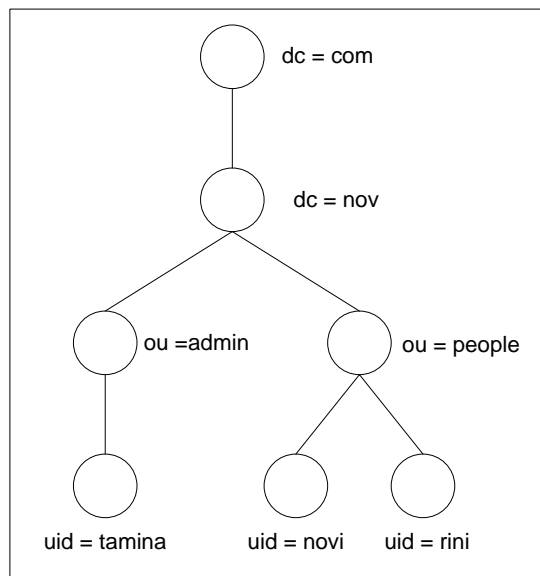
Istilah-istilah yang terdapat pada organisasi LDAP:

1. *Entry*: Input yang harus dimasukkan kedalam sebuah *database* direktori. Titik yang menyimpan nama seseorang, berikut informasi lain yang akan disimpan pada direktori. Seperti alamat *email*, posisi pekerjaan, dll. Disebut juga *Distigued Service Entry (DSE)*.
2. *Attribute*: *Field* yang diisikan pada sebuah direktori *database*. Contoh *common name (CN)* adalah atribut menandakan nama dari sebuah entri.
3. *Object*: jika dianalogikan pada sebuah *database*, *object* dapat dikatakan sebagai tabel.
4. *Distinguished Name (DN)*: Nama unik bersifat global yang digunakan untuk mengidentifikasi sebuah entri.
5. *Relative Distinguished Name (RDN)*: Setiap level dalam komponen direktori menghasilkan komponen DN, komponen-komponen DN inilah yang disebut RDN.
6. *Directory Information Tree (DIT)*: Informasi yang termuat pada pohon direktori.
7. *Schema*: Memuat suatu aturan yang menjelaskan bagaimana data dikelompokkan sesuai dengan bagan direktori LDAP.

Dalam sistem LDAP, *entries* disusun berdasar tertib hirarki pohon struktur. Untuk sebuah titik *root*, secara umum ditunjukkan dengan suatu atribut *dc (domain component)*, namun dapat juga langsung ditunjuk dengan atribut *cn (common name)*, atau *o (organization)*, ataupun *ou (organization unit)*. Berikut beberapa atribut untuk sebuah titik pada direktori servis.

1. *uid = user id*

2. cn = common name
3. sn = sure name
4. l = location
5. ou = organization unit
6. o = organization
7. dc = domain component
8. st = state
9. c = country



**Gambar 3 Contoh Hirarki Pohon LDAP**

Gambar 3 merupakan contoh hirarki pohon pada LDAP yang terdiri dari beberapa atribut yang menggambarkan sebuah tabel *database* penyimpanan data *user* terstruktur dengan atribut *dc* (*domain component*) sebagai *root*. Kemudian dilanjutkan dengan

atribut *ou* (*organization unit*) sebagai *field* yang membedakan unit organisasi, dan *user-user* yang berada didalam masing-masing unit.

### **II.2.3 Directory security**

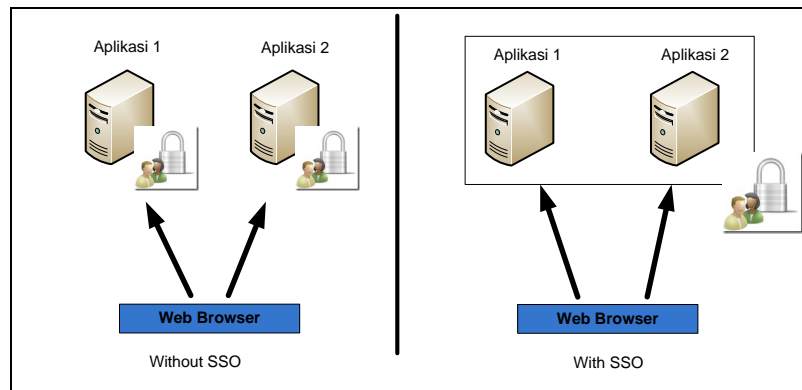
Menurut Arkills (2003), *Authentication* adalah sebuah proses yang menyatakan dan menegaskan siapa sebenarnya *user*, dengan kata lain ini membangun identitas dari *client*. Secara praktek identitas ini biasanya adalah sebuah *username*, *user identity* (*uid*), tiket, atau sertifikat.

Menurut Arkills (2003), *Authorization* adalah sebuah proses pembangunan dimana sebuah *client* diotorisasi untuk mengakses *resource*. Proses ini dapat ditentukan dengan kombinasi dari faktor *access control* seperti *authentication identity*, *source IP*, kekuatan enkripsi, metode autentikasi, operasi *request*, dan sumber *request*.

## **II.3 Single Sign On (SSO)**

### **II.3.1 Pengertian Single Sign On (SSO)**

*Single Sign On (SSO)* merupakan sebuah sistem yang memfasilitasi penanganan *user account* untuk beberapa *server* dengan hanya menggunakan satu *username* dan satu *password*. Teknologi ini sangat diminati, khususnya dalam jaringan yang sangat besar dan bersifat heterogen. Dengan menggunakan SSO, seorang pengguna hanya cukup melakukan proses autentikasi sekali saja untuk mendapatkan izin akses terhadap semua layanan yang terdapat di dalam jaringan.



**Gambar 4 Perbedaan sebelum dan sesudah SSO**

Gambar 4 menunjukkan perbedaan antara aplikasi yang tidak menerapkan *Single Sign On (SSO)* dan aplikasi yang menerapkan *Single Sign On (SSO)*, jika seorang *user* ingin masuk kedalam sebuah aplikasi maka *user* tersebut harus mempunyai beberapa *username* dan *password* dan melakukan proses *login* beberapa kali. Sedangkan jika sebuah web portal menggunakan SSO, maka *user* tersebut hanya memasukkan satu *username* dan satu *password* untuk melakukan proses *login*, dan *user* dapat mengakses aplikasi yang tersedia dalam web portal.

*Single sign on (SSO)* adalah sebuah *session* atau proses autentikasi *user* yang memungkinkan *user* untuk menyediakan sebuah *credential* sekali dengan maksud untuk mengakses banyak aplikasi. *Single Sign On (SSO)* mengautentikasi *user* untuk mengakses semua aplikasi yang telah di-*authorized* untuk diakses. Ini menghilangkan permintaan *authentication* lagi ketika *user* mengganti aplikasi selama *session* berlaku. SSO dapat menghemat biaya untuk memelihara *password* yang rumit yang dapat mencapai ratusan dolar setiap pengguna tiap tahun (wikipedia).

### **II.3.2 PHP OpenLDAP**

PHP OpenLDAP adalah protokol *Single Sign On* untuk web. digunakan untuk menangani masalah komunikasi antara aplikasi web yang berbeda, sehingga semua aplikasi dapat diintegrasikan ke dalam sebuah web portal. PHP OpenLDAP adalah

merupakan sebuah sistem autentikasi yang menyediakan sebuah jalan yang aman untuk sebuah aplikasi untuk meng-autentikasi seorang *user*.

### **II.3.3 Implementasi Single Sign On (SSO)**

Aplikasi yang mengimplementasi metode *Single Sign On (SSO)* harus mempunyai suatu terminologi yang sama, misalnya di suatu instansi memiliki beberapa layanan aplikasi yang berbeda-beda tapi dapat diakses melalui satu web portal yang sama. Sedangkan untuk mengimplementasikan metode *Single Sign On (SSO)* ada beberapa syarat utama yang harus dipenuhi antara lain:

1. Membuat sebuah data user terpusat dengan OpenLDAP.
2. Menginstal PHP OpenLDAP sebagai fitur *single* autentikasi atau penyatuan *login* untuk beberapa aplikasi.
3. Merubah *session* pada masing-masing aplikasi menjadi *session* PHP OpenLDAP.

### **II.3.4 Implementasi Single Sign On (SSO) Universitas Bina Nusantara**

Universitas Bina Nusantara merupakan salah satu contoh instansi yang telah menerapkan metode *Single Sign On (SSO)* dalam mengintegrasikan beberapa aplikasi pada satu web portal. Aplikasi-aplikasi tersebut adalah Blog, Forum, IPTV, *Messenger*, *SoftPhone* dan *Sharing Content*. Sebelum menerapkan metode *Single Sign On (SSO)*, aplikasi-aplikasi tersebut diakses secara terpisah dengan melakukan beberapa kali login untuk masuk ke dalam masing-masing aplikasi. Saat ini aplikasi-aplikasi tersebut dapat diakses dalam satu web portal dengan sekali login untuk mengakses semua aplikasi yang tersedia pada web portal tersebut. Fasilitas yang digunakan untuk membangun web portal *Single Sign On (SSO)* tersebut, universitas bina nusantara menggunakan *framework* berupa *Central Authentication Service (CAS)* sebagai *single* autentikasi dan *Lightweight Directory Access Protocol (LDAP)* sebagai *directory service*.

Pada penelitian ini, aplikasi yang disediakan pada web portal adalah: web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam. Dan fasilitas yang digunakan berupa *framework* PHP OpenLDAP sebagai *single* autentikasi dan *Lightweight Directory Access Protocol (LDAP)* sebagai *directory service*.

#### **II.4 Web Lembaga Amil Zakat (LAZ) Masjid Raya Batam**

Web LAZ Masjid Raya Batam merupakan sebuah website yang digunakan untuk memberi informasi kepada pengguna serta mempermudah pengguna untuk mengetahui lebih jauh tentang zakat dan perhitungan zakat yang ada di Masjid Raya Batam. Dalam *website* ini terdapat fasilitas perhitungan zakat, forum diskusi, artikel, galeri, dan berita. Dalam *website* ini terdapat beberapa kategori *user*, yaitu:

1. *Administrator* adalah *user* yang dapat mengelola secara keseluruhan *website* Lembaga Amil Zakat Masjid Raya Batam. Pengelolaan yang dilakukan *user* yaitu menambah, mengubah, menghapus data atau informasi yang berkaitan dengan web.
2. *User Login* adalah *user* biasa atau masyarakat umum yang sudah melakukan registrasi sehingga dapat *login* ke *website* dan mempunyai hak akses untuk menggunakan layanan yang tersedia, seperti menggunakan fasilitas layanan forum dan artikel.
3. *User Public*, *user* atau masyarakat umum yang belum atau tidak melakukan registrasi sehingga hanya bisa melihat halaman *website* tanpa bisa menggunakan layanan yang tersedia.

## II.5 Web IF Politeknik Batam

Web IF Politeknik Batam adalah sebuah *website* yang digunakan untuk memberi informasi kepada pengguna tentang semua hal yang ada di Prodi Teknik Informatika serta informasi tentang data alumni Prodi Teknik Informatika Politeknik Batam. Dalam *website* ini terdapat beberapa kategori user, yaitu:

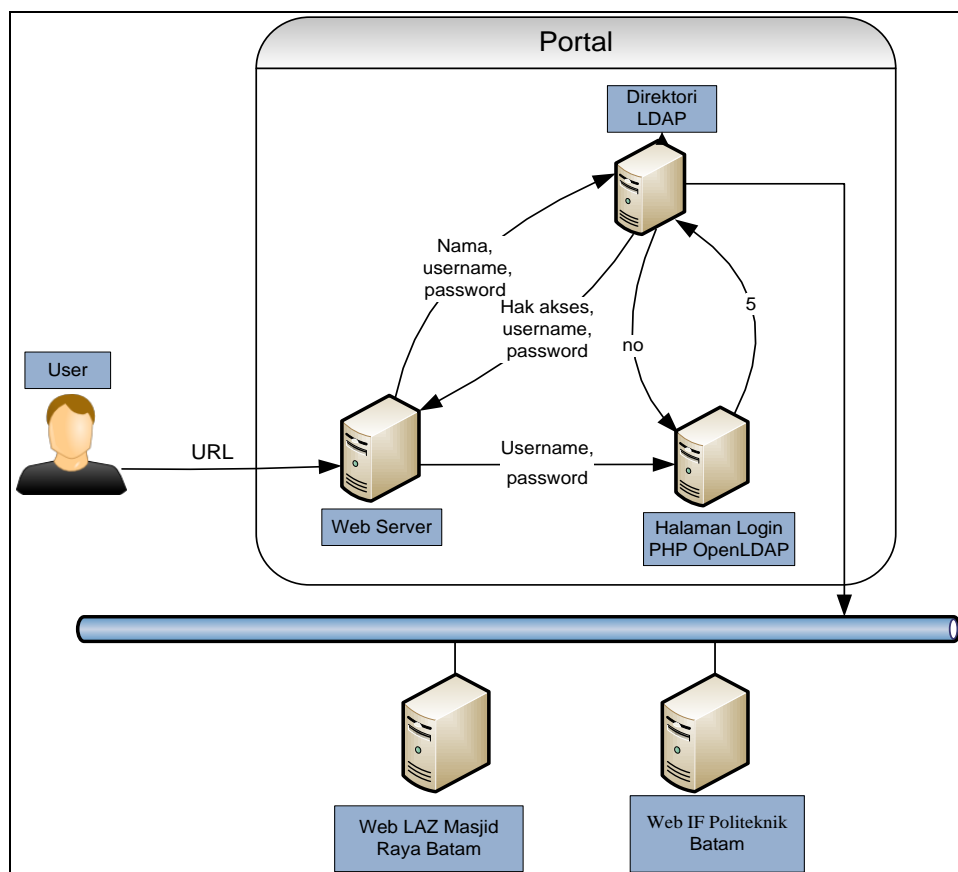
1. Administrator adalah *user* yang dapat mengelola secara keseluruhan dari *website*. Pengelolaan yang dilakukan *user* yaitu menambah, mengubah, menghapus data atau informasi yang berkaitan dengan web.
2. *User Public* adalah *user* atau masyarakat umum yang hanya bisa melihat informasi yang tersedia pada halaman *website*.

## Bab III Analisis

Analisis dari perancangan penelitian ini meliputi deskripsi umum proses *Single Sign On* (SSO), proses sistem *Single Sign On* (SSO), kebutuhan data, *activity diagram* dan *use case diagram*.

### III.1 Deskripsi Umum Proses Single Sign On (SSO)

Deskripsi umum proses *Single Sign On* (SSO) pada gambar 5, menjelaskan secara umum bagaimana sistem dapat berjalan.



Gambar 5 Deskripsi Umum Sistem Proses SSO

Gambar 5 merupakan gambaran umum bagaimana jalannya *service* ketika ada *user* masuk dalam portal. Seorang *user* baru yang masuk dalam *web server* tidak dapat

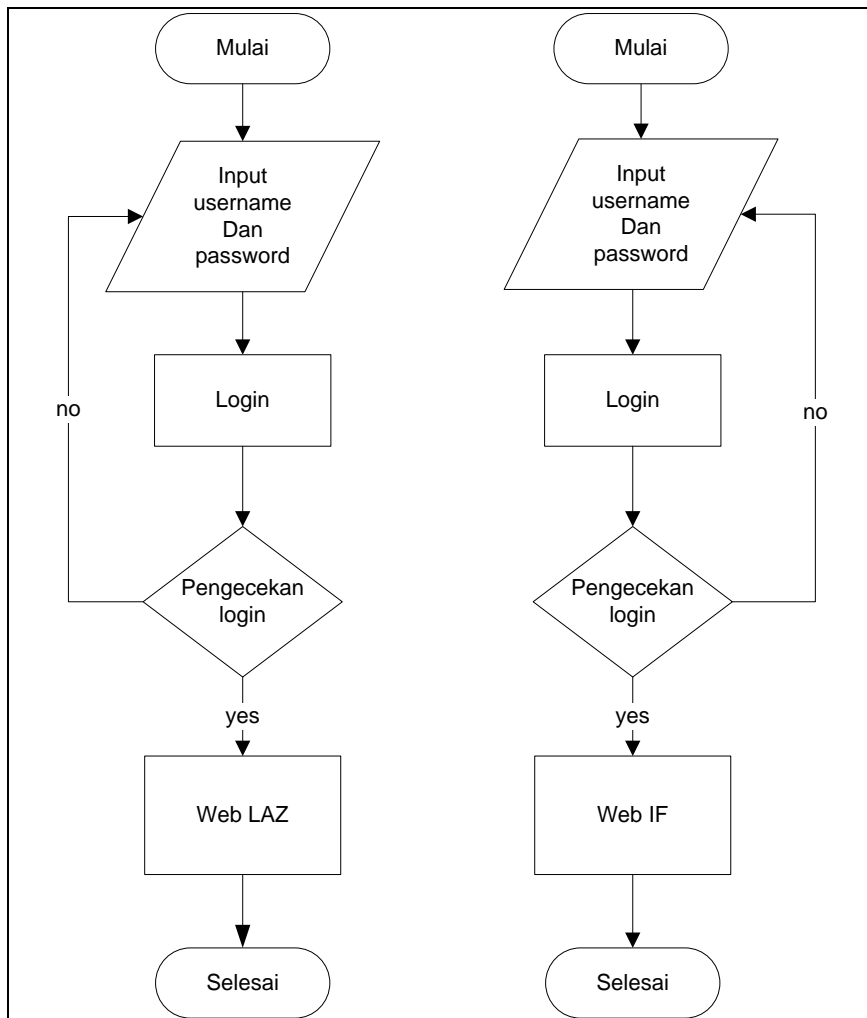
melakukan *login*, karena *user* tersebut belum terdaftar. Ketika *user* berhasil melakukan registrasi, data *user* dikirimkan ke direktori *LDAP*. Kemudian baru *user* dapat melakukan *login*. Pada saat melakukan *login* maka *user* melakukan autentikasi dengan menggunakan PHP OpenLDAP pada *web server*. Saat mengautentikasi maka data *user* tersebut akan di-*binding* dengan data yang ada pada direktori *LDAP*. Bila sukses maka PHP OpenLDAP akan membuat suatu *session* untuk *user* tersebut yang dapat dipakai untuk mengakses setiap layanan yang terdapat pada web portal.

Setelah itu baru *user* dapat memilih layanan yang ada di dalam web portal. Bila *user* memilih suatu layanan, maka server dari layanan tersebut akan mengecek apakah *user* tersebut diijinkan untuk mengakses layanan yang ada, dengan jalan mengecek *session* yang ada pada PHP OpenLDAP. Bila *session user* tersebut ada maka *user* bisa menikmati layanan yang ada, bila tidak maka akan dikembalikan ke halaman portal untuk melakukan *login* ulang. *Session* pada PHP OpenLDAP tidak akan dihancurkan sampai *user* melakukan *logout* pada web portal. Oleh karena itu penting bagi *user* untuk melakukan *logout*.

Penggunaan metode SSO ini dimaksudkan untuk membuat suatu integrasi aplikasi web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam. Metode SSO ini tidak lepas dari penggunaan PHP OpenLDAP dan LDAP. Penggunaan PHP OpenLDAP dikarenakan PHP OpenLDAP adalah salah satu framework yang menyediakan fasilitas SSO. Dalam mengaplikasikan SSO dibutuhkan suatu *credential* yang disimpan pada sebuah server terpusat atau digunakan sebuah direktori. *Lightweight Directory Access Protocol (LDAP)* didesain untuk meng-*update* dan mencari direktori yang berjalan lewat jaringan *TCP/IP*.

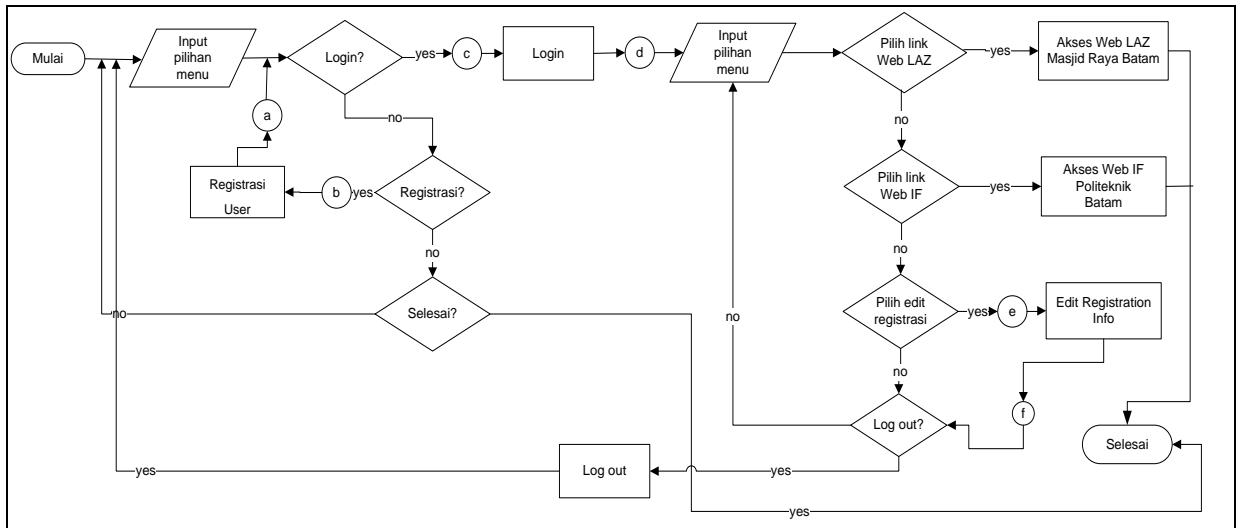
### **III.2 Proses Sistem Single Sign On (SSO)**

Proses sistem SSO ini menjelaskan perbedaan antara sistem sebelum penerapan metode SSO yang dijelaskan pada gambar 6 dan sistem setelah penerapan metode SSO yang dijelaskan pada gambar 7.



**Gambar 6** Proses Sebelum Single Sign On (SSO)

Gambar 6 merupakan gambaran proses sebelum menggunakan *Single Sign On (SSO)*. Pada gambar diatas dijelaskan bahwa untuk masuk ke aplikasi web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam, seorang *user* harus melakukan *login* disatu aplikasi dan melakukan *login* lagi ketika ingin masuk ke aplikasi lainnya.



**Gambar 7 Proses Setelah Single Sign On (SSO)**

Gambar 7 merupakan gambaran proses setelah menggunakan *Single Sign On (SSO)*. Pada gambar 7 dijelaskan bahwa untuk masuk ke aplikasi web, *user* hanya perlu melakukan *login* satu kali saja kemudian dapat mengakses kedua layanan yang tersedia, yaitu web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam. Selain itu, *user* hanya cukup melakukan *logout* sekali saja jika ingin keluar dari kedua aplikasi tersebut.

### III.3 Kebutuhan Data

Data yang dibutuhkan dalam aplikasi SSO ini berupa data user yang disimpan dalam direktori LDAP. Data-data tersebut, dijelaskan pada tabel 1.

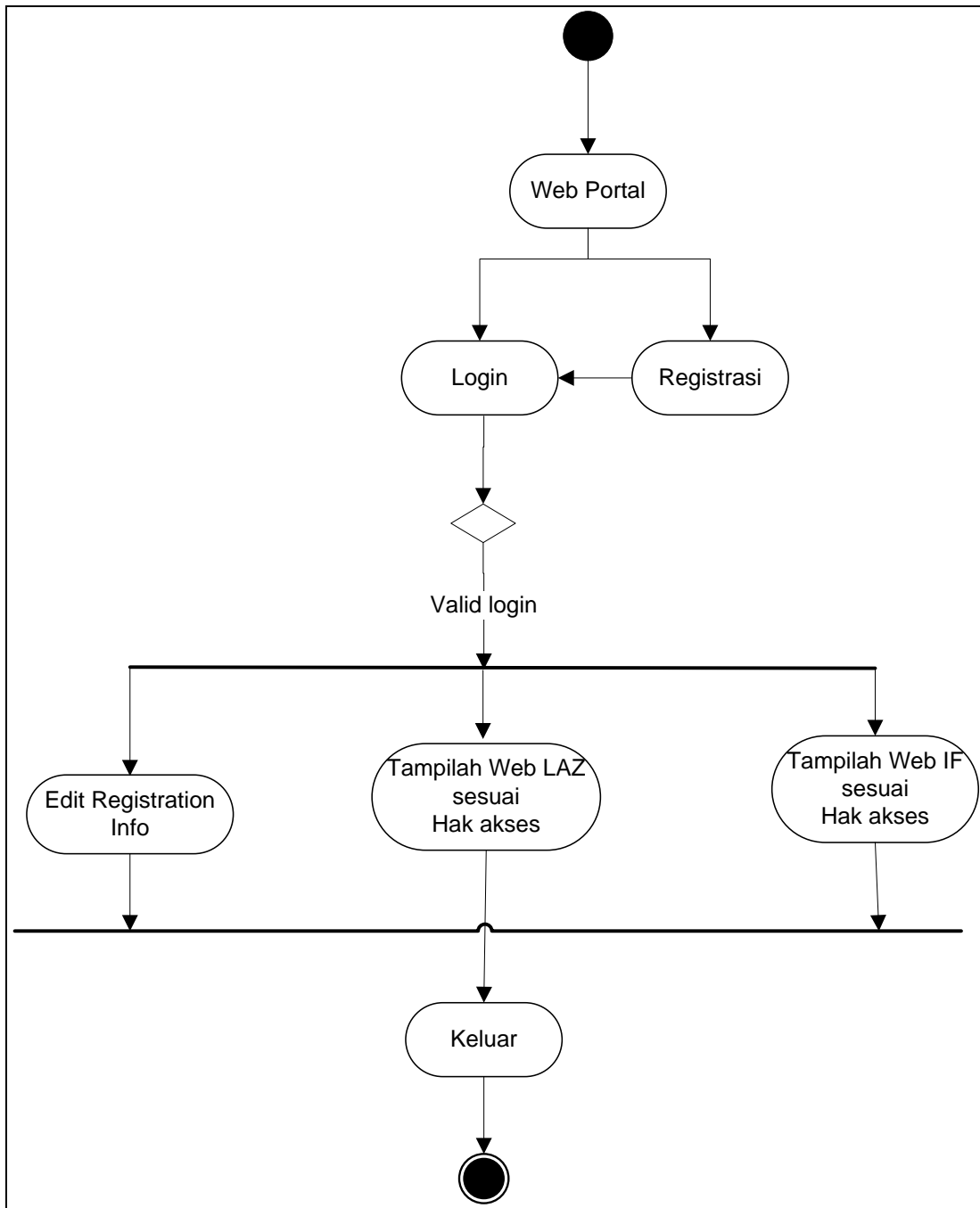
**Tabel 1 Data User pada direktori LDAP**

Nama field	Deskripsi Isi	Type & Length	Boleh NULL	Default	Keterangan lain
uid	Username	Varchar	Tidak	-	Primary Key
userPassword	Pam nssword	Varchar	Tidak	-	-
displayName	Full name	Varchar	Tidak	-	-
initials	Inisial	Varchar	Tidak	-	-
mail	Email	Varchar	Tidak	-	-

<b>Nama field</b>	<b>Deskripsi Isi</b>	<b>Type &amp; Length</b>	<b>Boleh NULL</b>	<b>Default</b>	<b>Keterangan lain</b>
o	Organisasi	Varchar	Tidak	-	-
title	Jabatan	Varchar	Tidak	-	-
postalAddresses	Alamat	Varchar	Tidak	-	-
l	Kota	Varchar	Tidak	-	-
postalCode	Kode pos	Varchar	Tidak	-	-
mobile	No hp	Varchar	Tidak	-	-
homePhone	No telp	Varchar	Tidak	-	-

### III.4 Activity Diagram

Activity diagram pada gambar 8, menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang.

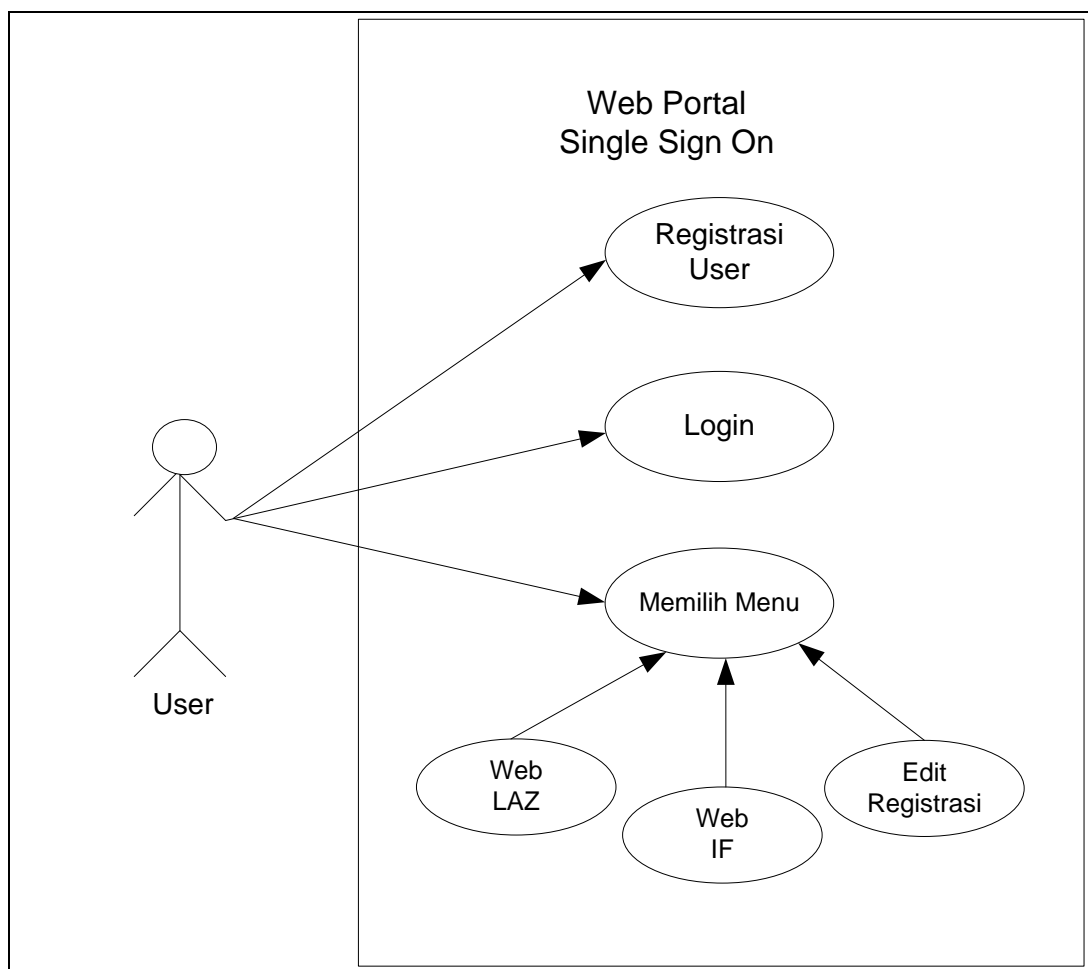


Gambar 8 Activity Diagram

Pada gambar 8 menggambarkan proses yang terjadi di sistem. Diketahui bahwa *user* masuk ke Web portal untuk dapat melihat web Lembaga Amil Zakat dan web IF Politeknik Batam serta dapat melakukan *login* dengan *metode Single Sign On (SSO)* untuk mengakses kedua aplikasi hanya dengan satu akun pengguna.

### III.5 Use Case Diagram

Use Case Diagram pada gambar 9, menggambarkan fungsionalitas yang diharapkan dari sistem. Melalui Use Case, dapat diketahui bagaimana interaksi antara aktor (*user*) dengan sistem.



Gambar 9 Use Case untuk Web Portal

Dari gambar 9 diatas terdapat satu aktor yaitu *user* dan terdapat lingkaran *elips* yang menunjukkan sebuah *Use Case*. Yaitu *use case* registrasi *user*, *login*, web Lembaga Amil Zakat Masjid Raya Batam, *use case* web IF Politeknik Batam, dan *Edit Registration Info*. Dari gambar diatas juga dapat dilihat garis yang menghubungkan antara *Use Case* dengan aktor yang disebut asosiasi. Seorang *user* melakukan registrasi terlebih dahulu dan kemudian melakukan *login* pada halaman *login* untuk kemudian dapat mengakses dan menggunakan layanan yang tersedia pada aplikasi web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam sesuai dengan hak aksesnya.

### **III.5.1 Skenario Use Case Diagram**

Deskripsi *Use Case* menceritakan secara rinci bagaimana *Use Case* berjalan pada sistem dan proses berjalannya sistem.

#### **III.5.1.1 Skenario Use Case: Registrasi User**

*Use Case* ini menggambarkan proses *user* melakukan registrasi data *user* untuk dapat masuk ke dalam aplikasi web portal.

Aktor : *user*

Skenario : *User* memilih menu registrasi, kemudian sistem menampilkan layar registrasi. *User* memasukkan *username*, *password*, *full name*, *initial*, *email*, *organization*, *title*, *address*, *city*, *ZIP*, *mobile*, *home phone*. Selanjutnya, data *user* akan masuk kedalam direktori LDAP dan *user* dapat *login* pada web portal.

*Precondition* : *User* belum registrasi.

*Postcondition* : *User* telah registrasi dan dapat melakukan *login* pada web portal.

### **III.5.1.2 Skenario Use Case: Login**

*Use Case* ini menggambarkan proses *user* melakukan *login* untuk dapat masuk ke dalam aplikasi web portal.

Aktor : *User*

Skenario : *User* memilih menu *login*, kemudian sistem menampilkan layar *login*. *User* memasukkan *username* dan *password*. Setelah itu sistem akan mengecek apakah *username* dan *password* yang dimasukkan oleh *user* sudah benar dengan cara mencocokkan dengan data yang tersedia dalam tempat penyimpanan data *user*. Jika *username* dan *password* benar, maka sistem akan menampilkan halaman utama. Dan jika *username* dan *password* *user* salah, maka akan tetap pada halaman *login*.

*Precondition* : *User* belum *login*.

*Postcondition* : *User* telah *login* dan sistem akan menampilkan menu utama.

### **III.5.1.3 Skenario Use Case: Web Lembaga Amil Zakat Masjid Raya Batam dan Web IF Politeknik Batam**

*Use Case* ini menggambarkan proses *user* untuk masuk ke dalam menu web Lembaga Amil Zakat Masjid Raya Batam dan web IF Politeknik Batam.

Aktor : *User*

Skenario : *User* memilih menu web Lembaga Amil Zakat Masjid Raya Batam atau menu web IF Politeknik Batam pada web portal. Sistem akan melakukan pengecekan apakah *user* sudah melakukan *login* atau belum. Jika *user* sudah dinyatakan melakukan *login*, maka sistem akan memunculkan menu aplikasi web sesuai pilihan menu yang dipilih oleh *user*. Namun jika *user* belum melakukan *login*, maka *user* akan dialihkan ke halaman *login* PHP OpenLDAP.

*Precondition* : *User* memilih menu aplikasi web pada web portal.

*Postcondition* : Tampil halaman *website* sesuai dengan pilihan menu yang dipilih oleh *user*.

#### **III.5.1.4 Use Case: Edit Registration Info**

*Use Case* ini menggambarkan tentang proses perubahan tentang data *user*.

Aktor : *User*

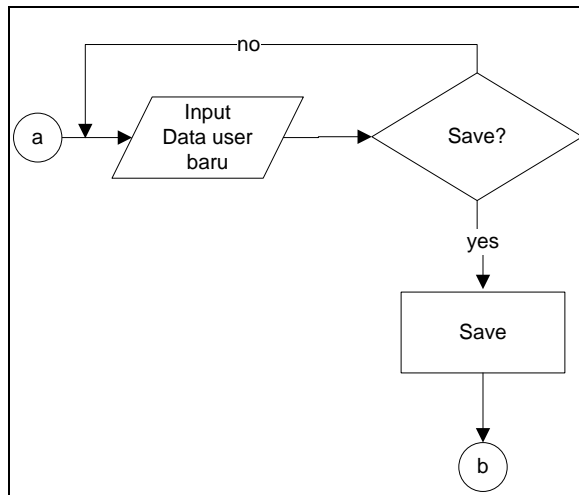
Skenario : *User* memilih menu *Edit Registration Info*. Kemudian sistem akan menampilkan halaman *Edit Registration Info*. *User* mengedit data yang lama dengan yang baru seperti: *password, full name, initial, email, organization, title, address, city, ZIP, mobile, home phone*. Sistem melakukan pengecekan apakah data *user* ada di tempat penyimpanan data *user*. Jika data *user* ada, maka sistem akan melakukan peng-update-an data *user* kemudian sistem akan menampilkan halaman keberhasilan perubahan data *user*.

*Precondition* : *User* memilih menu *Edit Registration Info*.

*Postcondition* : Data registrasi *user* berhasil diubah.



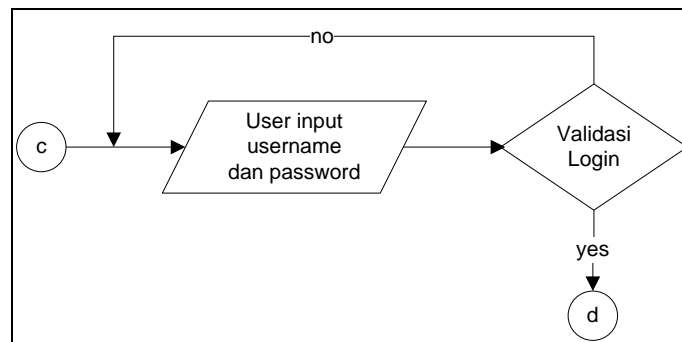
#### IV.1.1 Perancangan Proses Registrasi user



Gambar 11 Diagram Alir Proses Registrasi User

Pada gambar 11 menjelaskan proses registrasi *user* baru pada aplikasi. Untuk membuat sebuah akun, *user* harus mendaftarkan *username*, *password*, *full name*, *initial*, *email*, *organization*, *title*, *address*, *city*, *ZIP*, *mobile*, *home phone* halaman Registrasi *User*. Selanjutnya, data akan tersimpan dalam *direktori* LDAP dan *user* dapat menggunakan layanan yang tersedia.

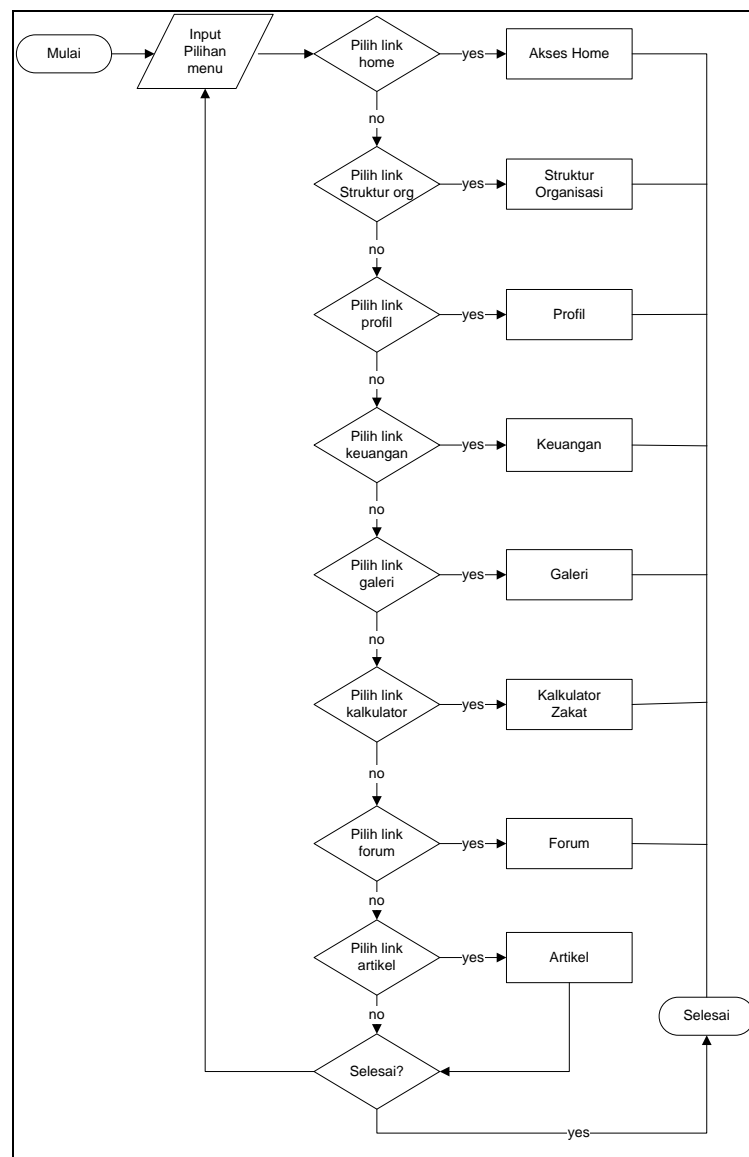
#### IV.1.2 Perancangan Proses User Login



Gambar 12 Diagram Alir untuk Login

Pada gambar 12 menjelaskan bahwa *user* melakukan proses *login* kedalam aplikasi. Ketika *user* memilih *button login*, maka selanjutnya *user* akan masuk ke dalam halaman *login* PHP OpenLDAP dimana *user* meng-*input* *username* dan *password*. setelah itu data yang di-*input* akan dicek ke LDAP, jika data cocok maka *user* akan masuk ke halaman utama aplikasi. Namun jika data *user* tidak maka akan dikembalikan lagi pada halaman *login* PHP OpenLDAP.

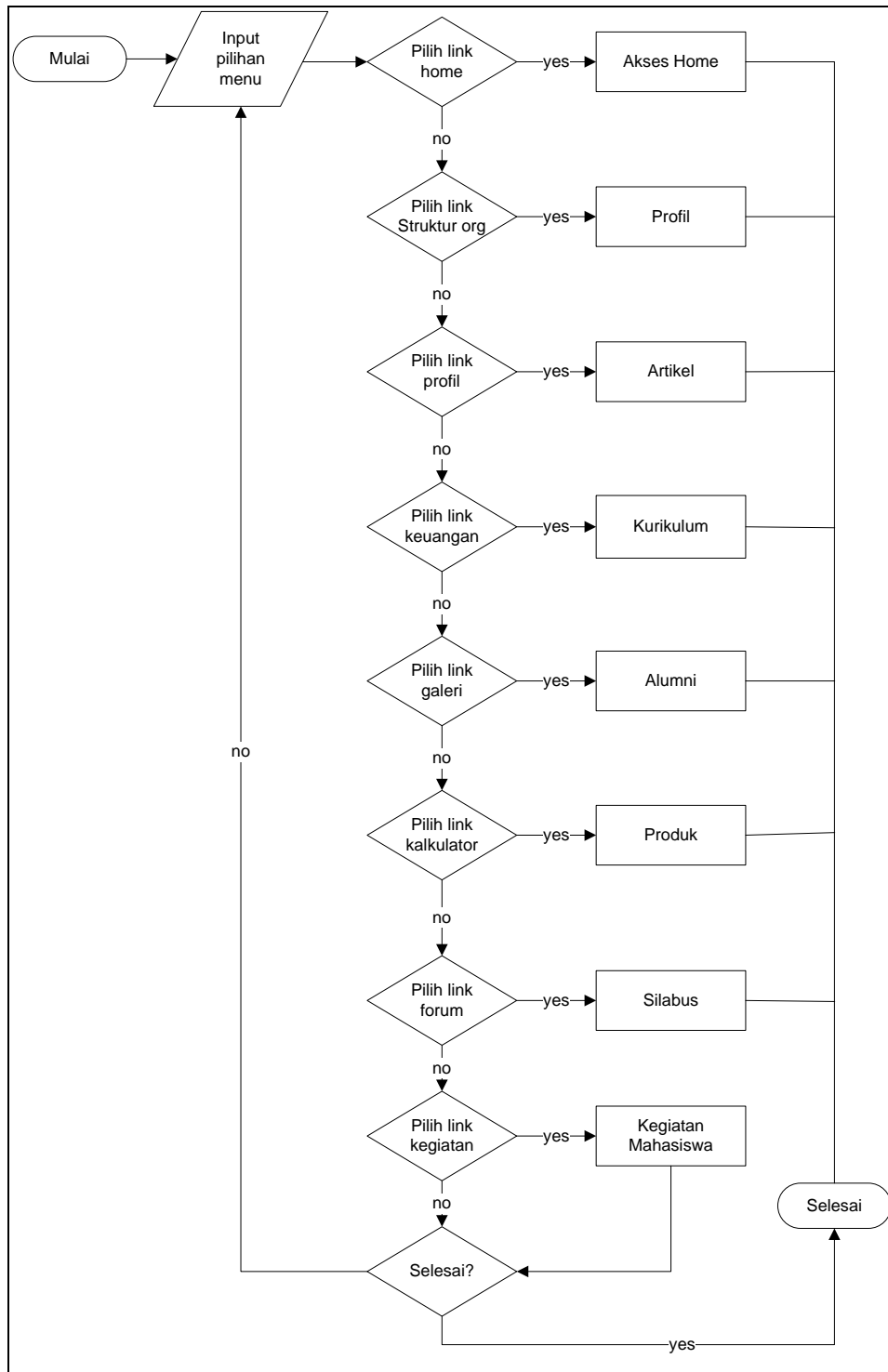
#### IV.1.3 Proses pada Web Lembaga Amil Zakat Masjid Raya Batam



Gambar 13 Diagram alir untuk web LAZ Masjid Raya Batam

Pada gambar 13 menjelaskan proses-proses yang ada pada web LAZ Masjid Raya Batam. *User* yang dapat mengakses aplikasi ini terdiri dari: *Administrator* dan *User* biasa. *Administrator* dapat mengakses semua fasilitas yang ada, seperti: menu *home*, profil, struktur organisasi, keuangan, galeri, kalkulator zakat, forum, artikel. Selain itu dapat melakukan pengolahan data seperti : menambah, merubah, dan menghapus. Sedangkan *user* biasa dapat melihat dan melakukan penambahan pada beberapa fasilitas yang tersedia seperti forum dan artikel.

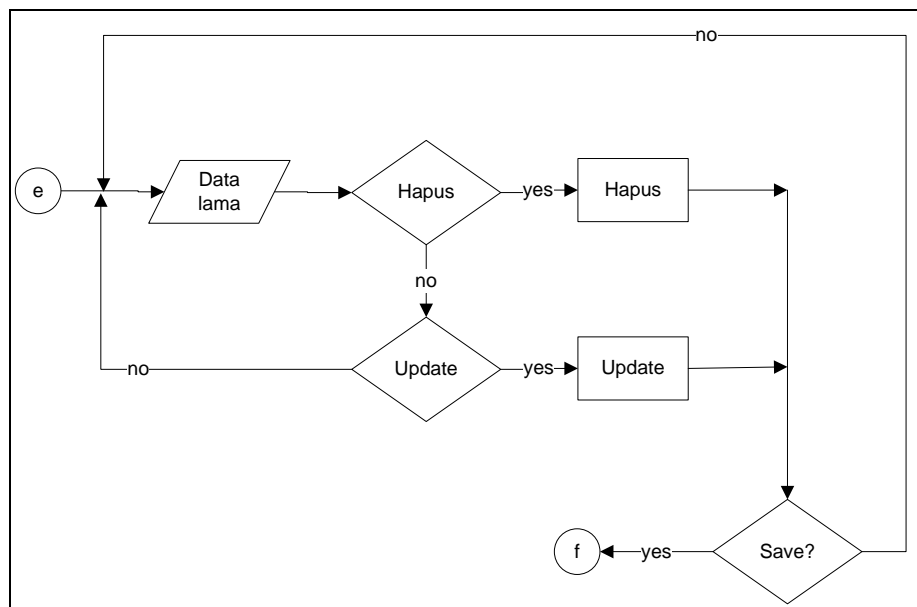
#### IV.1.4 Proses pada Web IF Politeknik Batam



Gambar 14 Diagram alir untuk web IF Politeknik Batam

Pada gambar 14 menjelaskan proses-proses yang ada pada web IF Politeknik Batam. *User* yang dapat mengakses aplikasi ini yaitu *Administrator*. *Administrator* dapat mengakses semua fasilitas yang ada, seperti: menu *home*, profil, artikel, kurikulum, alumni, produk, silabus dan kegiatan mahasiswa.

#### IV.1.5 Perancangan Proses Edit Registration Info



**Gambar 15 Diagram alir untuk Edit Registration Info**

Pada gambar 15 menjelaskan proses-proses *edit registration info* pada aplikasi dimulai dengan mengedit *password, full name, initial, email, organization, title, address, city, ZIP, mobile, home phone*. Jika data *user* yang di-input lengkap, maka data akan tersimpan pada direktori LDAP. Namun jika tidak, maka akan tetap berada pada halaman *edit registrastion info*.

## IV.1.6 Perancangan Antarmuka

### IV.1.6.1 Nama Objek: Layar Utama

#### IV.1.6.1.1 Rancangan Tampilan



**Gambar 16 Tampilan Utama Web Portal**

Gambar 16 menggambarkan layar utama web portal aplikasi *Single Sign On (SSO)* yang terdiri dari menu web Lembaga Amil Zakat Masjid Raya Batam, web IF Politeknik Batam, *Login here*, *Register here*, dan *Contact*.

## IV.1.6.2 Nama Objek : Layar Registrasi User Baru

### IV.1.6.2.1 Rancangan Tampilan

### IMPLEMENTASI SINGLE SIGN ON (SSO)

---

#### Registrasi

Username	<input type="text"/>
Password	<input type="text"/>
Full Name	<input type="text"/>
Initial	<input type="text"/>
Email	<input type="text"/>
Organization	<input type="text"/>
Title	<input type="text"/>
Adress	<input type="text"/>
City	<input type="text"/>
ZIP	<input type="text"/>
Mobile	<input type="text"/>
Home Phone	<input type="text"/>

Page 1

Gambar 17 Tampilan Registrasi User Baru

Gambar 17 menggambarkan layar *user* baru yang ingin membuat *account* baru pada web portal. Setelah melakukan registrasi, *user* dapat melakukan *login* untuk masuk kedalam web portal dan dapat menggunakan layanan yang tersedia dalam aplikasi web portal.

### IV.1.6.2.2 Algoritma/Query

Nama Layar : Layar Registrasi User Baru

Nama Operasi : Registrasi User Baru

```
Query1      :  $info['uid']=$uid;
              info['userPassword']="{MD5}".base64_encode(pack("H*",m
              d5($_POST['userPassword'])));
              $info['displayName']=$_POST['displayName'];
              $info['initials']=strtoupper($_POST['initials']);
              $info['mail']=$_POST['mail'];
              $info['o']=$_POST['o'];
              $info['title']=$_POST['title'];
              $info['postalAddress']=$_POST['postalAddress'];
              $info['postalCode']=$_POST['postalCode'];
              $info['l']=$_POST['l'];
              $info['mobile']=$_POST['mobile'];
              $info['homePhone']=$_POST['homePhone'];
              $r = ldap_add($ds,$userdn,$info);
```

Algoritma :

```
Insert (username, password, full name, initial, email,
organization, title, address, city, zip, mobile, dan home phone)

If pilih = register then

Open direktori LDAP
```

```
Jalankan Query1

If (username=null and password=null and full name=null and
initial=null and email=null and organization=null and
title=null and address=null and city=null and zip=null and
mobile=null and home phone=null) then

    Tetap pada layar registrasi user baru

Else

    Masuk ke menu login

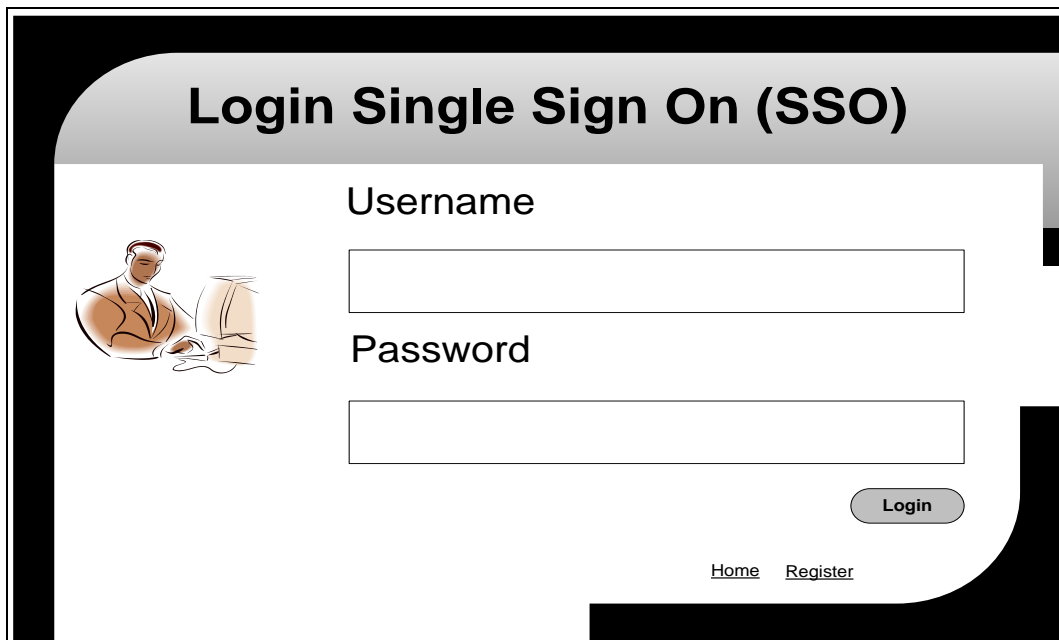
End if

End if

Close direktori
```

#### **IV.1.6.3 Nama Objek: Layar Login single Sign On (SSO)**

##### **IV.1.6.3.1 Rancangan Tampilan**



**Gambar 18 Tampilan Halaman Login Single Sign On (SSO)**

Gambar 18 merupakan tampilan halaman *login Single Sign On (SSO)* yang digunakan oleh *user* registrasi untuk masuk ke dalam aplikasi web portal.

#### **IV.1.6.3.2 Algoritma/Query**

Nama Layar :Layar Login SSO

Nama Operasi : Login SSO

Query2 : `:$r = ldap_bind($ds,$userdn,$userpassword);`

Algoritma :

```
Insert (username dan password)
```

```
If pilih = login then
```

```
Open direktori LDAP

    Jalankan Query2

    If username = null and password = null then

        Tetap pada layar login

    Else

        Masuk ke menu utama

    End if

End if

Close direktori
```

#### **IV.1.6.4 Nama Objek : Edit Registration Info**

##### **IV.1.6.4.1 Rancangan Tampilan**

**IMPLEMENTASI SINGLE SIGN ON (SSO)**

---

**Edit Registration Info**

Username	<input type="text" value="irmanti"/>
Password	<input type="text"/>
Full Name	<input type="text" value="Rini Irmanti"/>
Initial	<input type="text" value="RI"/>
Email	<input type="text" value="Rinirmanti@gmail.com"/>
Organization	<input type="text" value="poltek"/>
Title	<input type="text" value="princess"/>
Adress	<input type="text" value="Bengkong indah"/>
City	<input type="text" value="batam"/>
ZIP	<input type="text" value="29432"/>
Mobile	<input type="text" value="085668223844"/>
Home Phone	<input type="text" value="452042"/>

[Home](#)   [Logout](#)   Page 1

**Gambar 19 Tampilan Edit Registration Info**

Gambar 19 adalah gambar tampilan *edit registration Info* bagi *user* registrasi (*administrator* dan *user login*) yang ingin mengganti data *user* lama dengan data *user* baru.

#### **IV.1.6.4.2 Algoritma/Query**

Nama Layar : Layar Edit Registration Info

Nama Operasi : Edit Registration Info

```

Query3      : $info['uid']=$uid;
              info['userPassword']="{MD5}".base64_encode(pack("H*",md5($_POST['userPassword'])));

              $info['displayName']=$_POST['displayName'];

              $info['initials']=strtoupper($_POST['initials']);

              $info['mail']=$_POST['mail'];

              $info['o']=$_POST['o'];

              $info['title']=$_POST['title'];

              $info['postalAddress']=$_POST['postalAddress'];

              $info['postalCode']=$_POST['postalCode'];

              $info['l']=$_POST['l'];

              $info['mobile']=$_POST['mobile'];

              $info['homePhone']=$_POST['homePhone'];

              $r = ldap_modify($ds,$userdn,$info);

```

**Algoritma** :

```

Insert (password, full name, initial, email, organization, title,
address, city, zip, mobile, home phone)

If pilih = save then

Open direktori LDAP

    Jalankan query3

    If (full name=null and initial=null and email=null and
organization=null and title=null and address=null and

```

```
        city=null and zip=null and mobile=null and home phone=null)
    then

        Tetap pada layar edit registration info

    Else

        Kembali ke menu utama

    End if

End if

Close direktori
```

#### **IV.1.6.5 Nama Objek: Logout**

##### **IV.1.6.5.1 Algoritma/Query**

Nama Operasi : Logout

Query4 : session\_destroy ();

Algoritma :

```
If pilih = logout
    Jalankan Query4
Then
Kembali ke halaman utama
End if
```

## **Bab V Implementasi dan Pengujian**

Pada proses implementasi dan pengujian, akan dijelaskan mengenai spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam membuat aplikasi web portal Single Sign On (SSO) ini. Serta proses instalasi dan konfigurasi perangkat lunak yang digunakan.

### **V.1 Spesifikasi Perangkat Keras dan Perangkat Lunak**

Pada bagian ini akan dijelaskan tentang proses pembuatan aplikasi web portal SSO. Pembuatan aplikasi ini memerlukan tahap-tahap yang harus dilalui terlebih dahulu sebelum dilakukan evaluasi terhadap sistem yang berjalan. Perangkat komputer yang digunakan dalam mengimplementasikan aplikasi ini, yaitu:

1. Perangkat Keras
  - a. Processor : Intel Core 2 Duo
  - b. Memory : 1.5 GB
2. Perangkat Lunak
  - a. Linux Mint 8 “Helena”
  - b. OpenLDAP
  - c. PHP OpenLDAP

### **V.2 Instalasi dan Konfigurasi**

#### **V.2.1 Instalasi dan Konfigurasi Server LDAP**

Bagian ini merupakan tahap awal dalam pembuatan aplikasi web portal *single sign on*. Pembuatan LDAP *server* dilakukan sebagai tempat penyimpanan data tentang *user* yang akan menggunakan aplikasi web portal ini. OpenLDAP digunakan karena *Operating system* yang digunakan adalah berbasis Linux, yaitu Linux Mint 8 “Helena”. Proses pembuatan LDAP *server* yaitu :

- Install Linux Mint 8 “Helena”

- Install OpenLDAP (Proses instalasi dan konfigurasi *server* LDAP mengacu pada Lampiran A)

### V.2.2 Instalasi dan Konfigurasi PHP OpenLDAP

PHP OpenLDAP merupakan sebuah aplikasi SSO *web based* yang digunakan untuk menghubungkan banyak aplikasi dalam banyak *server* sehingga membuat suatu aplikasi menjadi satu seperti sebuah portal. Proses instalasi dan konfigurasi PHP OpenLDAP ini yaitu :

1. Ekstrak aplikasi PHPOpenLDAP ke web direktori. Misalkan di '/var/www'  

```
#cp phpOpenLDAP.tar.bz2 /var/www  
#tar -xjf phpOpenLDAP.tar.bz2
```
2. Edit file `phpopenldap_config.php` dan sesuaikan dengan setting server LDAP
3. Buka file `php.ini`  

```
#vi /etc/php5/apache2filter.php.ini
```
4. Di dalam file `php.ini`, cari entry 'auto\_prepend' dan ubah menjadi  

```
auto_prepend = phpOpenLDAP/phpopenldap_check.php
```
5. Cari entri 'include\_path' dalam file `php`
6. Simpan file `php.ini`
7. Buat softlink dari direktori aplikasi `phpOpenLDAP` ke direktori dengan perintah :  

```
'include_path' di file php.ini tadi  
#ln -s /var/www/phpOpenLDAP/ /usr/share/php/
```
8. Restart web server dengan perintah :  

```
#etc/init.d/apache2 restart.
```

## Bab VI Kesimpulan dan Saran

### VI.1 Kesimpulan

Pada proses pengimplementasian dan pengujian metode *Single Sign On (SSO)* dengan menggunakan *PHP OpenLDAP* dan *Lightweight Data Access Protocol (LDAP)* dapat disimpulkan:

1. Melalui contoh yang digunakan yaitu aplikasi web LAZ Masjid Raya Batam dan web IF Politeknik Batam dapat diakses dengan baik menggunakan teknologi yang disebut di atas.
2. Aplikasi yang akan menerapkan metode *Single Sign On (SSO)* harus dapat dimodifikasi *session*-nya sehingga *user* yang ingin mengimplementasi kan metode ini harus memiliki *source code* aplikasi.

### VI.2 Saran

Untuk pengembangan selanjutnya, saran yang dapat diberikan adalah: Mengimplementasikan metode *Single Sign On (SSO)* menggunakan *framework* yang berbeda, untuk jenis aplikasi yang berbeda.

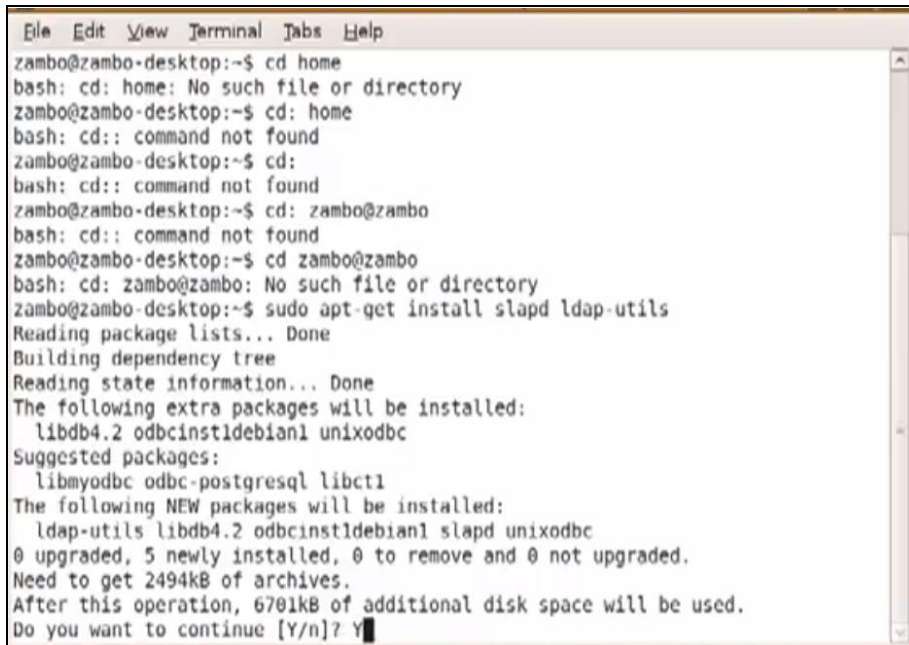
## DAFTAR PUSTAKA

1. <http://purpalacious.arieflatu.net/2009/08/penggunaan-sistem-single-sign-on-dengan-ldap/> (diakses pada tanggal 12 maret 2010)
2. <http://www.binus-access.com/> (diakses pada tanggal 18 maret 2010)
3. <http://www.kandar.info/index.php/php/integrasi-single-sign-on-openid-pada-website-berbasis-php/> (diakses pada tanggal 5 maret 2010)
4. [www.infolinux.web.id](http://www.infolinux.web.id) (diakses pada tanggal 21 april 2010)
5. <http://lecturer.ukdw.ac.id/budsus/sister/ldap.pdf> (diakses pada tanggal 22 april 2010)
6. <http://kelasjarkom.wordpress.com/category/pengenalan-ldap-by-oni-brian/> (diakses pada tanggal 8 april 2010)
7. <http://willmen46.wordpress.com/2007/09/13/pengenalan-ldap-lightweight-directory-access-protocol/> (diakses pada tanggal 10 maret 2010)
8. <http://www.openldap.org/> (diakses pada tanggal 10 maret 2010)
9. <http://budi.insan.co.id/courses/e17010/2004/dika-proposal.pdf> (diakses pada tanggal 22 april 2010)
10. <http://id2.php.net/manual/en/function.ldap-add.php/> (diakses pada tanggal 19 mei 2010)
11. [http://www.batan.go.id/sjk/tutorial/web\\_multimedia/Otentifikasi%20Menggunakan%20Directory%20Service%20OpenLDAP.pdf](http://www.batan.go.id/sjk/tutorial/web_multimedia/Otentifikasi%20Menggunakan%20Directory%20Service%20OpenLDAP.pdf) (diakses pada tanggal 19 april 2010)

## LAMPIRAN A PROSES INSTALASI LDAP

### A.1 Instal paket OpenLDAP

```
#apt-get install slapd ldap-utils
```

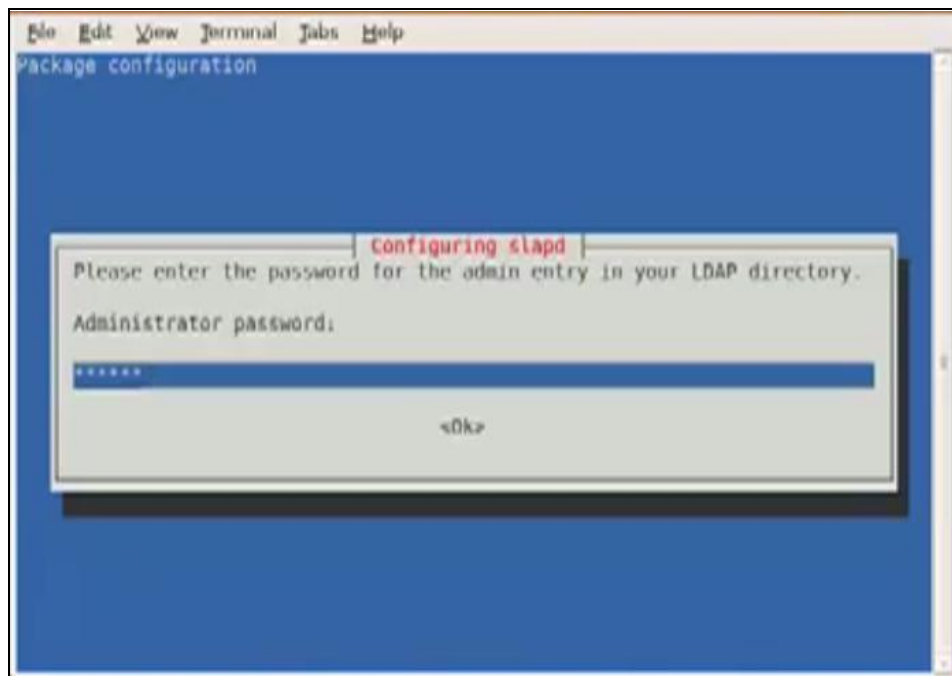


```
File Edit View Terminal Tabs Help
zambo@zambo-desktop:~$ cd home
bash: cd: home: No such file or directory
zambo@zambo-desktop:~$ cd: home
bash: cd:: command not found
zambo@zambo-desktop:~$ cd:
bash: cd:: command not found
zambo@zambo-desktop:~$ cd: zambo@zambo
bash: cd:: command not found
zambo@zambo-desktop:~$ cd zambo@zambo
bash: cd: zambo@zambo: No such file or directory
zambo@zambo-desktop:~$ sudo apt-get install slapd ldap-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libdb4.2 odbcinstdebiana1 unixodbc
Suggested packages:
  libmyodbc odbc-postgresql libct1
The following NEW packages will be installed:
  ldap-utils libdb4.2 odbcinstdebiana1 slapd unixodbc
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 2494kB of archives.
After this operation, 6701kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

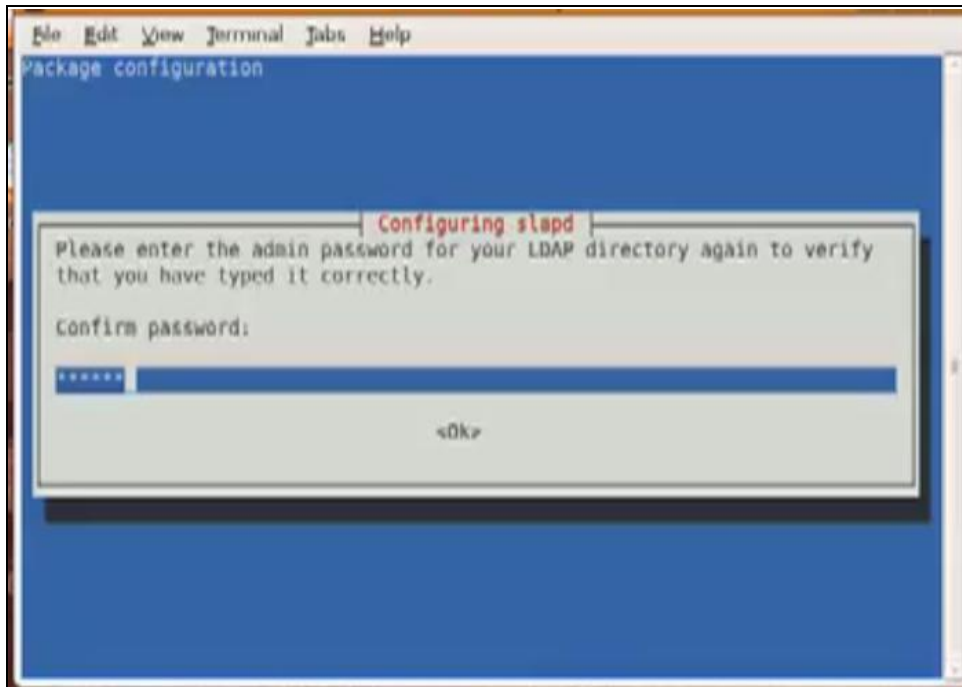
Pilih huruf “Y” untuk melanjutkan instalasai

```
File Edit View Terminal Tabs Help
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libdb4.2 odbcinstdebi1 unixodbc
Suggested packages:
  libmyodbc odbc-postgresql libct1
The following NEW packages will be installed:
  ldap-utils libdb4.2 odbcinstdebi1 slapd unixodbc
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 2494kB of archives.
After this operation, 6701kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Get:1 http://us.archive.ubuntu.com intrepid/main libdb4.2 4.2.52+dfsg-4 [400kB]
Get:2 http://us.archive.ubuntu.com intrepid/main odbcinstdebi1 2.2.11-16build
2 [66.3kB]
Get:3 http://us.archive.ubuntu.com intrepid/main unixodbc 2.2.11-16build2 [295kB
]
Get:4 http://us.archive.ubuntu.com intrepid-updates/main slapd 2.4.11-0ubuntu6.1
[1467kB]
Get:5 http://us.archive.ubuntu.com intrepid-updates/main ldap-utils 2.4.11-0ubun
tu6.1 [266kB]
Fetched 2494kB in 6s (381kB/s)
```

Selanjutnya, akan muncul layar *package configuration*. Isikan *password administrator LDAP*.



Setelah memasukkan *password admin*, akan muncul layar untuk memasukkan konfirmasi *password*.



Setelah memasukkan konfirmasi *password*, LDAP akan melanjutkan proses instalasi.

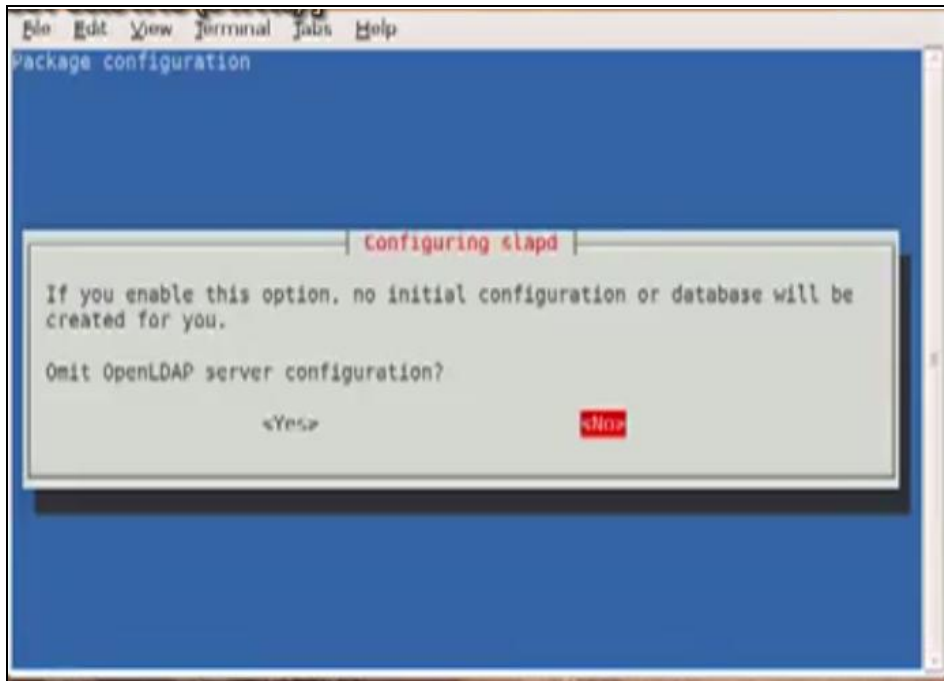
```
File Edit View Terminal Tabs Help
Selecting previously deselected package unixodbc.
Unpacking unixodbc (from .../unixodbc_2.2.11-16build2_i386.deb) ...
Selecting previously deselected package slapd.
Unpacking slapd (from .../slapd_2.4.11-0ubuntu6.1_i386.deb) ...
Selecting previously deselected package ldap-utils.
Unpacking ldap-utils (from .../ldap-utils_2.4.11-0ubuntu6.1_i386.deb) ...
Processing triggers for man-db ...
Setting up libdb4.2 (4.2.52+dfsg-4) ...
Setting up odbcinst1debian1 (2.2.11-16build2) ...

Setting up unixodbc (2.2.11-16build2) ...

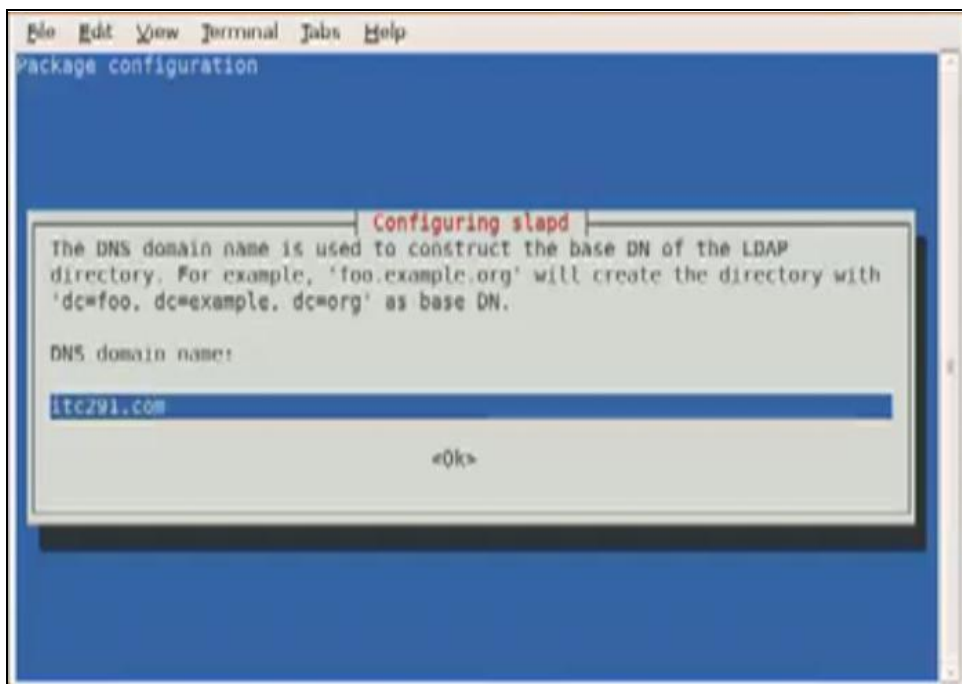
Setting up slapd (2.4.11-0ubuntu6.1) ...
  Creating new user openldap... done.
  Creating initial slapd configuration... done.
  Creating initial LDAP directory... done.
Reloading AppArmor profiles : done.
Starting OpenLDAP: slapd.

Setting up ldap-utils (2.4.11-0ubuntu6.1) ...
Processing triggers for libc6 ...
ldconfig deferred processing now taking place
zambo@zambo-desktop:~$ sudo dpkg-reconfigure slapd
```

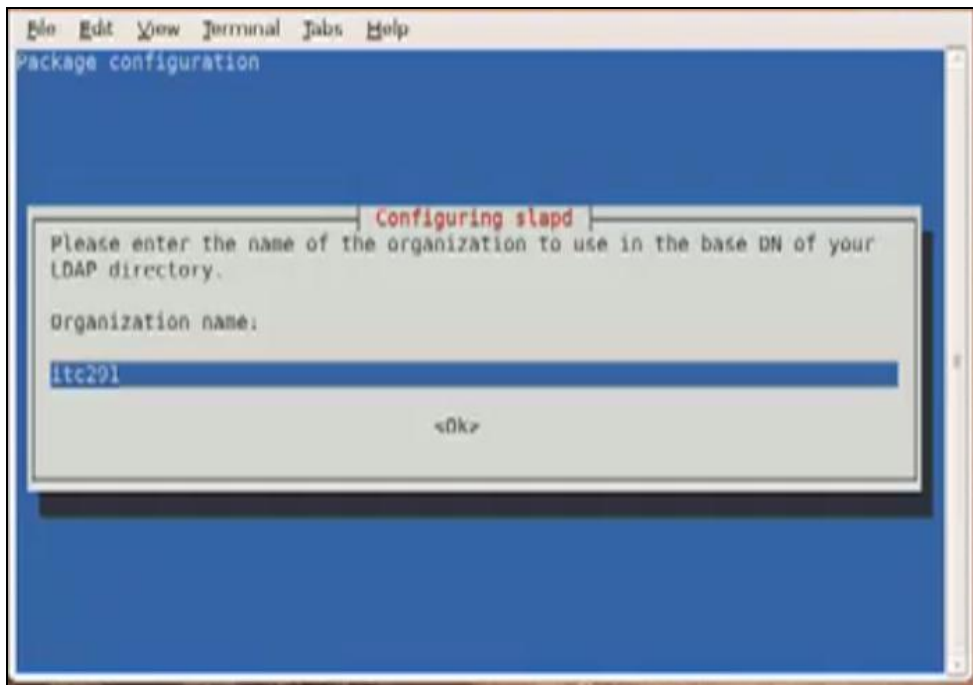
Pada proses selanjutnya, jika muncul pertanyaan “Omit OpenLDAP server configuration?” maka pilihlah menu pilihan “<No>”.



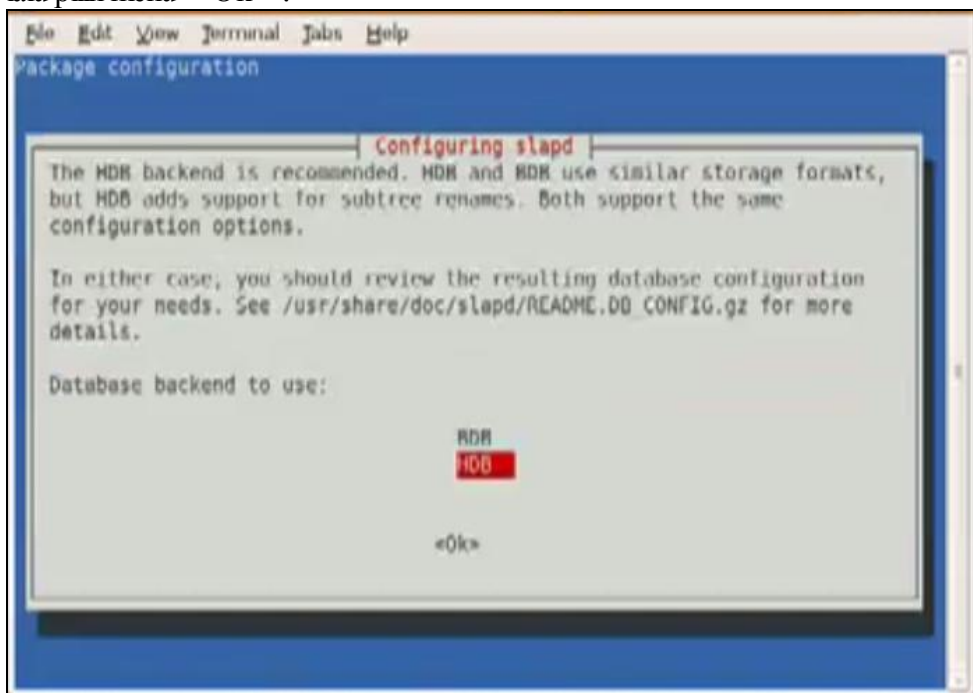
Kemudian masukkan nama domain yang diinginkan lalu pilih menu "<Ok>".



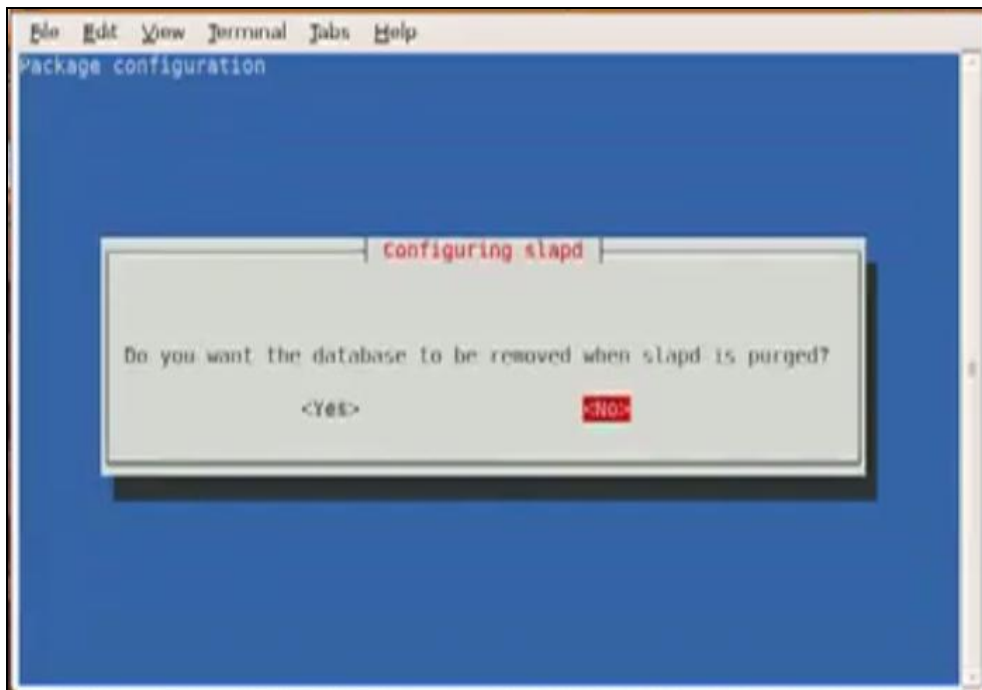
Masukkan nama organisasi lalu pilih menu "<Ok>".



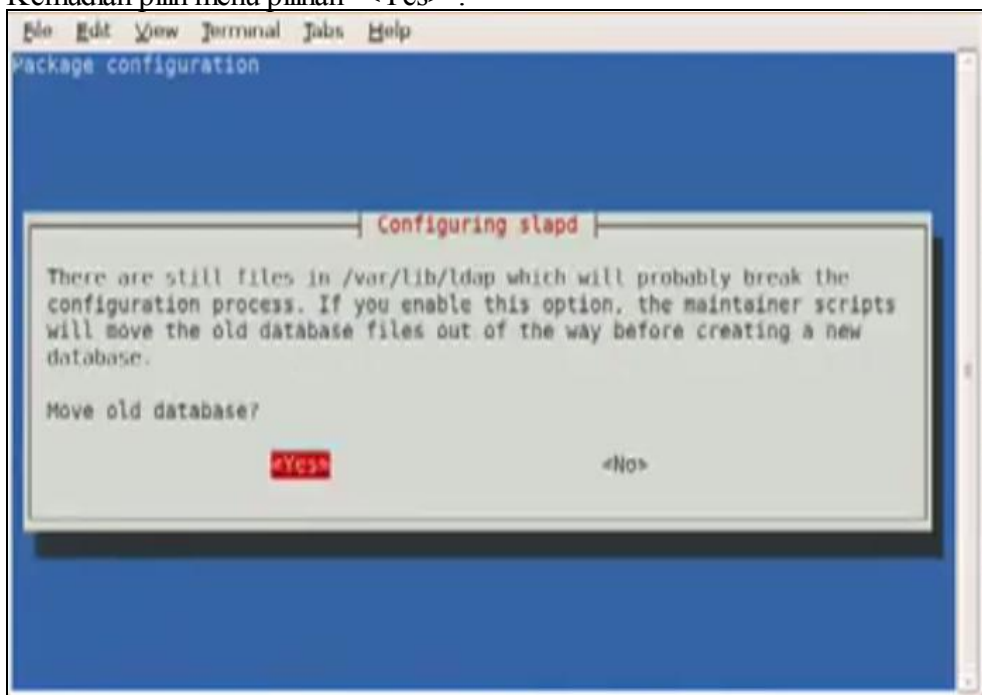
Jika muncul pertanyaan “Database backend to use:” maka pilihlah menu pilihan “HDB” lalu pilih menu “<Ok>”.



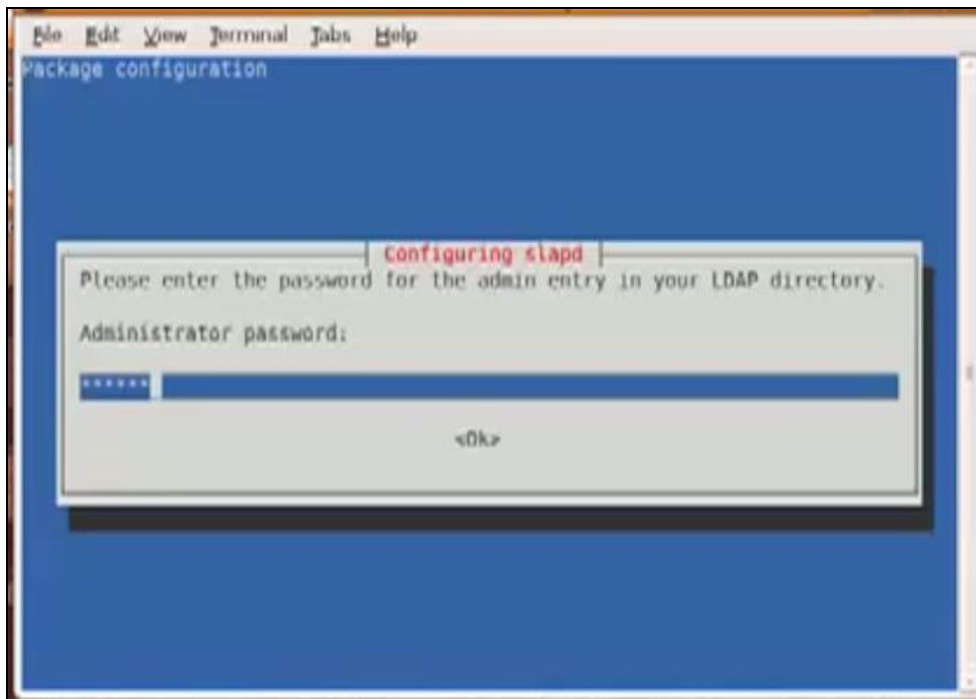
Kemudian ketika muncul pertanyaan “Do you want the database to be removed when slapd is purged?” maka pilihlah menu pilihan “<No>”.



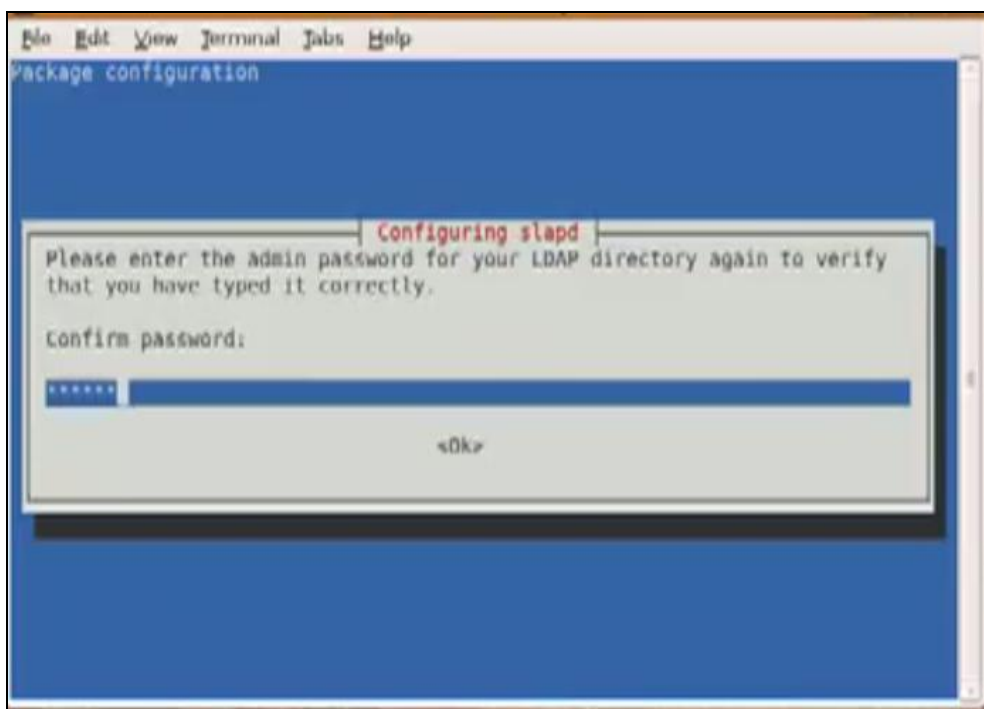
Kemudian pilih menu pilihan "<Yes>".



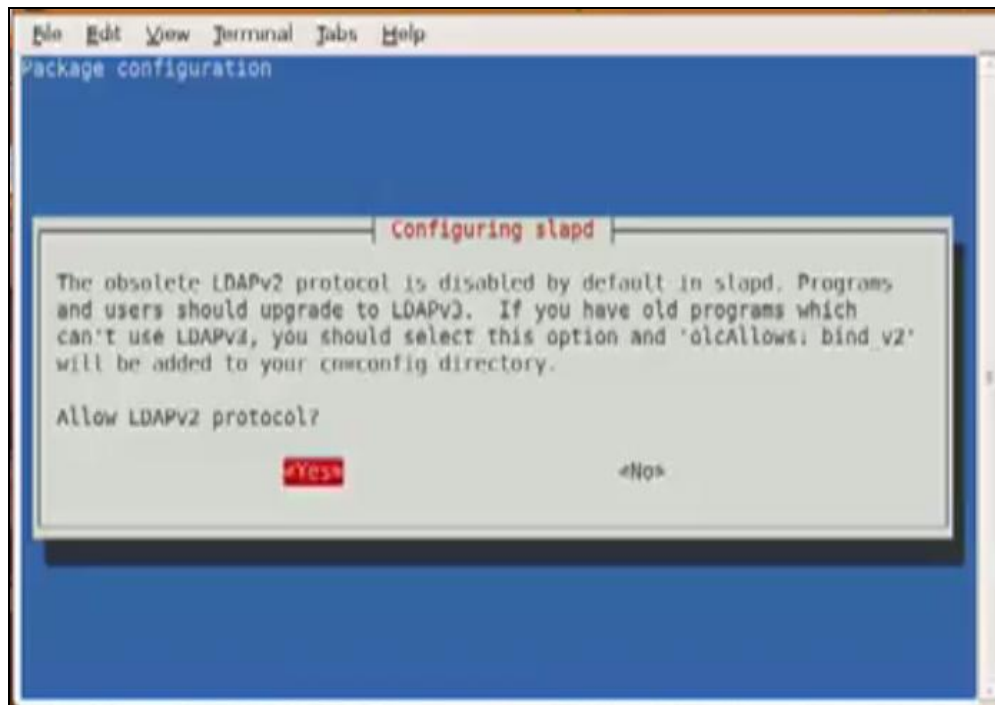
Masukkan password admin, kemudian pilih menu pilihan "<Ok>".



Masukkan konfirmasi *password* admin, lalu pilih menu pilihan "<Ok>".



Kemudian pilih menu pilihan "<Yes>" jika muncul pertanyaan "Allow LDAPV2 protocol?".



## A.2 Menambahkan Skema default pada direktori '/etc/ldap'

```
# cd /etc/ldap

#ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/ldap/schema/cosine.ldif

#ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/ldap/schema/inetorgperson.ldif

#ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/ldap/schema/nis.ldif
```

```
Terminal
File Edit View Terminal Help
nov nov # cd /etc/ldap
nov ldap # ldapadd -Y EXTERNAL_H ldap:/// -f /etc/ld
ldap/ ld.so.cache ld.so.conf ld.so.conf.d/
nov ldap # ldapadd -Y EXTERNAL_H ldap:/// -f /etc/ldap/schema/cosine.ldif
Add or modify entries from an LDAP server
usage: ldapadd [options]
The list of desired operations are read from stdin or from the file
specified by "-f file".
Add or modify options:
-a add values (default)
-c continuous operation mode (do not stop on errors)
-E [!]ext=extparam modify extensions (! indicate s criticality)
-f file read operations from 'file'
-M enable Manage DSA IT control (-MM to make critical)
-P version protocol version (default: 3)
-S file write skipped modifications to 'file'
Common options:
-d level set LDAP debugging level to 'level'
-D binddn bind DN
-e [!]ext=[=extparam] general extensions (! indicates criticality)
[!]assert=<filter> (RFC 4528; a RFC 4515 Filter string)
[!]authzid=<authzid> (RFC 4370; "dn:<dn>" or "u:<user>")
[!]chaining=[=resolveBehavior[/<continuationBehavior>]]
one of "chainingPreferred", "chainingRequired",
"referralsPreferred", "referralsRequired"
[!]managedSAIt (RFC 3296)
[!]noop
ppolicy
[!]postread=[<attrs>] (RFC 4527; comma-separated attr list)
[!]preread=[<attrs>] (RFC 4527; comma-separated attr list)
[!]relax
abandon, cancel, ignore (SIGINT sends abandon/cancel,
or ignores response; if critical, doesn't wait for SIGINT.
not really controls)
-h host LDAP server
-H URI LDAP Uniform Resource Identifier(s)
-I use SASL Interactive mode
-n show what would be done but don't actually do it
-N do not use reverse DNS to canonicalize SASL host name
-O props SASL security properties
```

```
Terminal
File Edit View Terminal Help
-n show what would be done but don't actually do it
-N do not use reverse DNS to canonicalize SASL host name
-O props SASL security properties
-o <opt>[=<optparam>] general options
nettimeout=<timeout> (in seconds, or "none" or "max")
-p port port on LDAP server
-Q use SASL Quiet mode
-R realm SASL realm
-U authcid SASL authentication identity
-v run in verbose mode (diagnostics to standard output)
-V print version info (-VV only)
-w passwd bind password (for simple authentication)
-W prompt for bind password
-X Simple authentication
-X authzid SASL authorization identity ("dn:<dn>" or "u:<user>")
-y file Read password from file
-Y mech SASL mechanism
-Z Start TLS request (-ZZ to require successful response)
nov ldap #
```

### A.3 Buat password terenkripsi

```
#slappasswd -h {MD5}
```

Kemudian masukkan password yang akan dienkrpsi.

```

Terminal
File Edit View Terminal Help
      (oo)\_____)\
      ( _)\_____)\
      ||-----w |
      ||         ||

nov@nov ~ $ su
Password:

( Wagner's music is better than it )
( sounds.                            )
(                                     )
( -- Mark Twain                       )
-----
o
o

{~.~.~}
( Y )
()~*~()
( )-( )

nov nov # slappasswd -h {MD5}
New password:
Re-enter new password:
{MD5}snhiAF90MKMZEViP7LOBzg==
nov nov #

```

## A.4 Buat file db.ldif

```
# vim.tiny db.ldif
```

```

Terminal
File Edit View Terminal Help
# Load dynamic backend modules
dn: cn=module{0},cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: {0}back_hdb

# Create the database
dn: olcDatabase={1}hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=nov,dc=com
olcRootDN: cn=admin,dc=nov,dc=com
olcRootPW: {MD5}snhiAF90MKMZEViP7LOBzg==
olcDbConfig: {0}set_cachesize 0 2097152 0
olcDbConfig: {1}set_lk_max_objects 1500
olcDbConfig: {2}set_lk_max_locks 1500
olcDbConfig: {3}set_lk_max_lockers 1500
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcDbIndex: uid pres,eq
olcDbIndex: cn,sn,mail pres,eq,approx,sub
olcDbIndex: objectClass eq

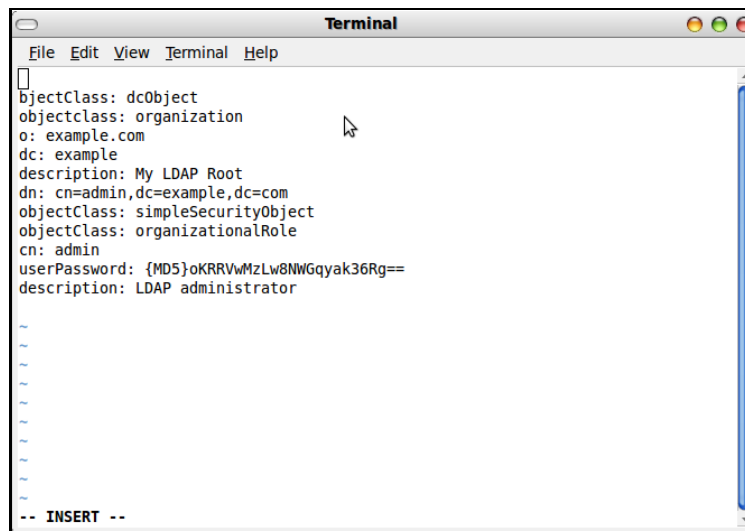
```

## A.5 Membuat dn dan account untuk admin

```
# vim.tiny base.ldif
```

Kemudian jalankan perintah

```
#ldapadd -Y EXTERNAL -H ldapi:/// -f base.ldif
```

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Help). The window displays the output of the ldapadd command. The output shows LDAP entry details for an organization and an administrator user. The text is as follows:

```
objectClass: dcObject
objectClass: organization
o: example.com
dc: example
description: My LDAP Root
dn: cn=admin,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
userPassword: {MD5}oKRRVwMzLw8NWGqyak36Rg==
description: LDAP administrator
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

## A.6 Buat file config.ldif

```
# vim.tiny config.ldif
```

Kemudian jalankan perintah

```
#ldapadd -Y EXTERNAL -H ldapi:/// -f config.ldif
```

```
dif x |
dn: cn=config
changetype: modify
delete: olcAuthzRegexp

dn: olcDatabase={-1}frontend,cn=config
changetype: modify
delete: olcAccess

dn: olcDatabase={0}config,cn=config
changetype: modify
delete: olcRootDN

dn: olcDatabase={0}config,cn=config
changetype: modify
add: olcRootDN
olcRootDN: cn=admin,cn=config

dn: olcDatabase={0}config,cn=config
changetype: modify
add: olcRootPW
olcRootPW: {MD5}snhiAF90MKMZEViP7LOBzg==

dn: olcDatabase={0}config,cn=config
changetype: modify
delete: olcAccess
```

## A.7 Buat acl untuk direktori LDAP

```
#vim.tiny acl.ldif
```

Kemudian jalankan perintah

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f config.ldif
```

```
acl.ldif x |
dn: olcDatabase={1}hdb,cn=config
add: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange by dn="cn=admin,dc=nov,dc=com" write by anonymous auth by self write by * none
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=nov,dc=com" write by * read
```

Server LDAP telah terinstall.

## LAMPIRAN B DOKUMEN RINCI TESTING

### Tim Penguji

1. Evaliata Sembiring, S.Kom (EV)

**Tabel 2 Hasil Rinci Testing**

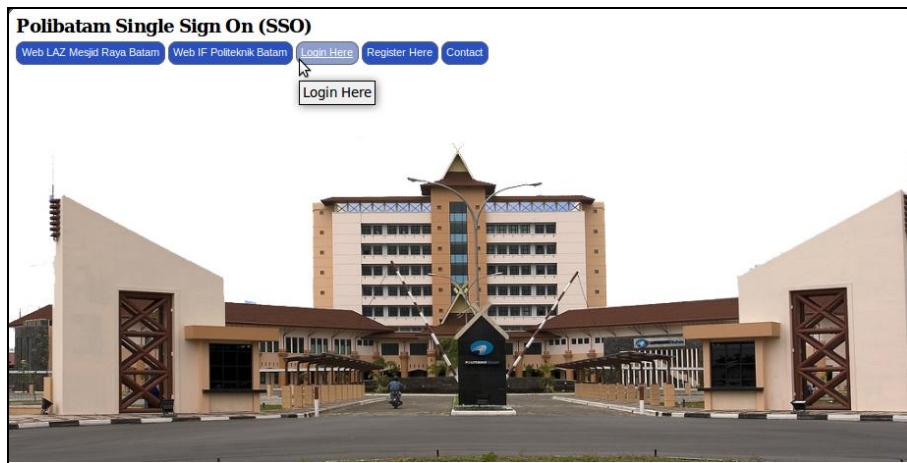
No	User	Use Case	Kelompok Uji	Prosedur& Kasus uji	Hasil yang diharap	Hasil Test	Tester	Tgl Testing	Ket
1	Guest	Registrasi	Normal	Input data user (username, password, full name, initial, email, organization, title, address, city, zip, mobile, dan home phone)	Data user tersimpan dalam direktori LDAP dan kemudian menampilkan halaman login.	Diterima	EV	21 Juni 2010	-
			Data Salah	Input data user tidak lengkap.	Tetap pada halaman registrasi user baru	Diterima	EV	21 Juni 2010	-
2	User Registrasi	Login	Normal	Input username dan password.	Tampil menu utama	Diterima	EV	21 Juni 2010	-
			Data Salah	Input username dan password yang salah	Tetap pada menu login	Diterima	EV	21 Juni 2010	-
		Web LAZ	Normal	Pilih menu Web LAZ	Tampil aplikasi Web	Diterima	EV	21 Juni 2010	Menu aplikasi

No	User	Use Case	Kelompok Uji	Prosedur & Kasus uji	Hasil yang diharapkan	Hasil Test	Tester	Tgl Testing	Ket
		Masjid Raya Batam		Masjid Raya Batam	LAZ Masjid Raya Batam dan dapat menggunakan layanan yang tersedia				tampil sesuai dengan hak akses user pada aplikasi.
			Data Salah	Tidak memilih menu Web LAZ Masjid Raya Batam	Tetap pada halaman menu utama	Diterima	EV	21 Juni 2010	-
		Web IF Politeknik Batam	Normal	Pilih menu Web IF Politeknik Batam	Tampil aplikasi Web IF Politeknik Batam dan dapat menggunakan layanan yang tersedia	Diterima	EV	21 Juni 2010	Menu aplikasi tampil sesuai dengan hak akses user pada aplikasi.
			Data Salah	Tidak memilih menu Web IF Politeknik Batam	Tetap pada halaman menu utama	Diterima	EV	21 Juni 2010	-
		Edit Registration Info	Normal	Input data user ( password, full name, initial, email, organization, title, address, city, zip, mobile, dan home phone)	Data user pada direktori LDAP akan ter-update dan masuk pada halaman menu utama.	Diterima	EV	21 Juni 2010	-

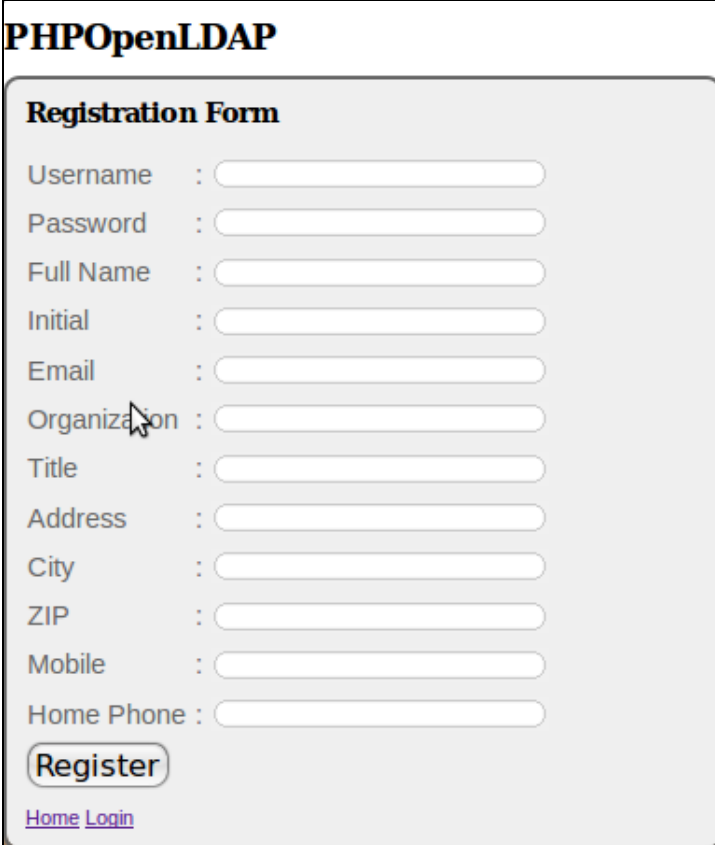
No	User	Use Case	Kelompok Uji	Prosedur& Kasus uji	Hasil yang diharap	Hasil Test	Tester	Tgl Testing	Ket
			Data Salah	Input data user tidak lengkap.	Tetap pada halaman edit registration info.	Diterima	EV	21 Juni 2010	-

## LAMPIRAN C MANUAL APLIKASI

1. Penggunaan aplikasi
  - a) Install LDAP
  - b) Install PHPOpenLDAP
2. Berikut tampilan aplikasi Single Sign On (SSO), klik menu Register Here untuk membuat username aplikasi.



3. Untuk melakukan registrasi, isikan data user pada halaman registrasi



**PHPOpenLDAP**

**Registration Form**

Username :

Password :

Full Name :

Initial :

Email :

Organization :

Title :

Address :

City :

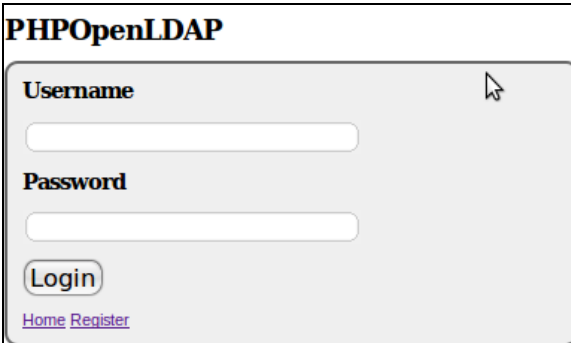
ZIP :

Mobile :

Home Phone :

[Home Login](#)

4. Pilih menu Login here untuk masuk ke dalam aplikasi. Kemudian masukkan username dan password pada halaman login.



**PHPOpenLDAP**

**Username**

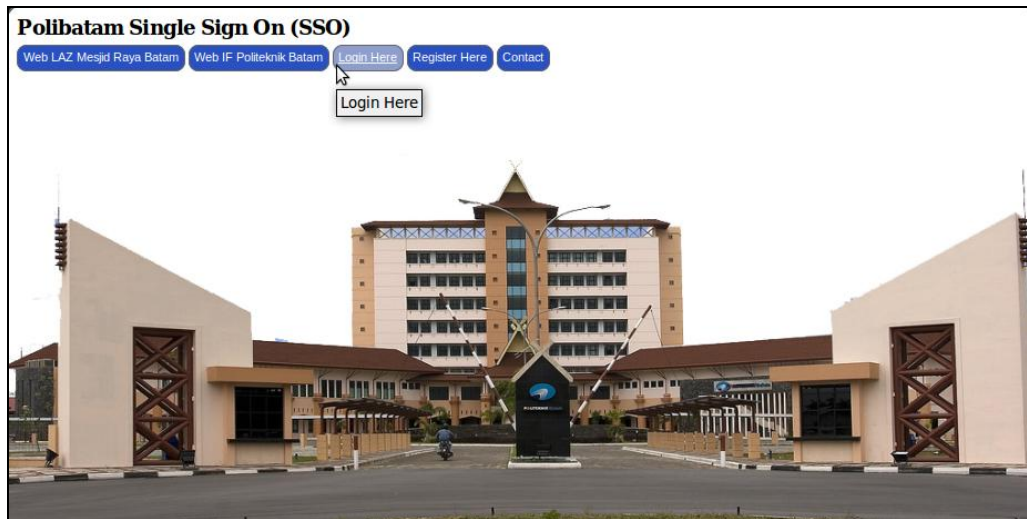
**Password**

[Home Register](#)

- Setelah login berhasil, akan tampil menu utama. Pilih link home pada menu utama untuk melihat pilihan menu aplikasi.



- Selanjutnya akan muncul halaman pilihan menu.



- Pilih menu Web LAZ Masjid Raya Batam untuk masuk ke aplikasi tersebut. Kemudian aplikasi akan terbuka pada halaman baru.



8. Pilih menu Web IF Politeknik Batam untuk masuk ke aplikasi tersebut. Kemudian aplikasi akan terbuka pada halaman baru.



9. Untuk melakukan perubahan data user, pilih link Edit Registration Info. Maka akan muncul halaman Edit Registration info yang berisi data lama. Lakukan perubahan pada *field* yang tersedia.

**PHPOpenLDAP**

**Modify Registration Form**

Username :

Password :

Full Name :

Initial :

Email :

Organization :

Title :

Address :

City :

ZIP :

Mobile :

Home Phone :

[Home](#) [Logout](#)