Modul Drupal untuk Single-Sign-On Link Eksternal

Rinta Kridalukmana

Abstract – Recently enterprises tend to develop web-based application to deal with their business process. With numerous applications, enterprise must manage user for each application and how to access those application in a good manner. Application portal with single-sign-on approach is one alternative in managing user and enterprise applications. Single-sign-on approach can reduce the necessaries to remember login information to each application by user. As a solution in web-based content management system, Drupal is one alternative to manage both issues above. Combined with LDAP (Lightweight Directory Access Protocol) to support single-sign-on approach, LDAP integration module for Drupal is needed. This module will authenticate user information from LDAP server. After LDAP authentication, the next problem is how to use the authenticated user information to sign on enterprise application. Other module needs to be developed solve the problem.

Index Terms : single-sign-on, LDAP, Drupal module

I. PENDAHULUAN

Sebagai salah satu solusi *content management system* berbasis web, Drupal memiliki potensi untuk dapat dimanfaatkan dalam mengelola *application farm* pada organisasi/enterprise. Dengen dukungan berbagai modul yang telah tersedia, Drupal saat ini telah mendukung proses *single-sign-on* yang akan mempermudah user dalam mengelola informasi login ke berbagai aplikasi yang ada pada perusahaan.

Melalui penelitian ini, akan dicoba untuk menggali lebih jauh bagaimana Drupal melalui pengembangan modul yang ada dapat digunakan untuk mengelola proses *single-sign-on* pada portal perusahaan dan sekaligus menjadi jembatan untuk proses otentikasi ke aplikasi yang ada.

II. RUMUSAN & BATASAN MASALAH

Pengembangan modul drupal untuk mendukung pengelolaan *single-sign-on* multi aplikasi merupakan rumusan masalah yang akan dibahas pada penelitian ini. Perumusan permasalahan ini sekaligus memberikan batasan bahwa *content management system* yang dipilih untuk implementasi pengelolaan aplikasi perusahaan adalah drupal versi 6. Dan dengan banyaknya alternatif yang dapat dilakukan untuk melakukan *single-sign-on*, maka pada penelitian ini digunakan LDAP sebagai alternatif pengelolaan login pengguna aplikasi.

III. LITERATUR

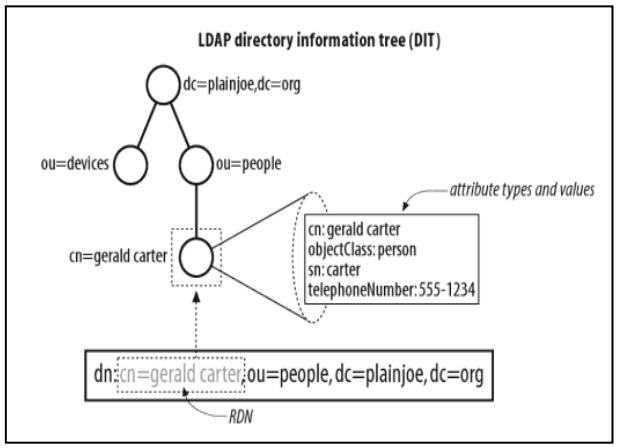
A. Lightweight Directory Access Protocol

Secara prinsip, LDAP adalah suatu protokol yang secara esensial merupakan sekumpulan pesan yang digunakan untuk mengakses data tertentu [1]. Namun istilah LDAP sendiri merupakan merupakan *extensible* dari protokol internet yang dipergunakan untuk mengakses layanan direktori [2]

RFC 2251 membagi direktori LDAP ke dalam 2 komponen, yaitu model protokol dan model data. Sedangkan Timothy A. Howes, Mark C. Smith, dan Gordon S. Good [2] mendefinisikan 4 model dalam LDAP, yaitu :

- *Information model*, yang menyediakan struktur dan tipe data yang dibutuhkan untuk menyusun *LDAP Directory Tree*
- *Naming Model*, yang mendefinisikan bagaimana *entry* dan data secara unik dapat diacu. Atribut unik ini selanjutnya disebut sebagai *Relative Distinguished Name* (RDN)
- *Functional Model*, merupakan protokol yang menyediakan layanan akses data pada pohon direktori. Akses ini diimplementasikan dengan operasi autentikasi (*binding*), operasi *query* (*searches and read*), dan operasi *update* (*writes*)
- *Security Model*, menyediakan mekanisme bagi klien untuk autentikasi.

Contoh *directory information tree* (DIT) pada LDAP dapat dilihan pada Gambar-1 berikut ini.



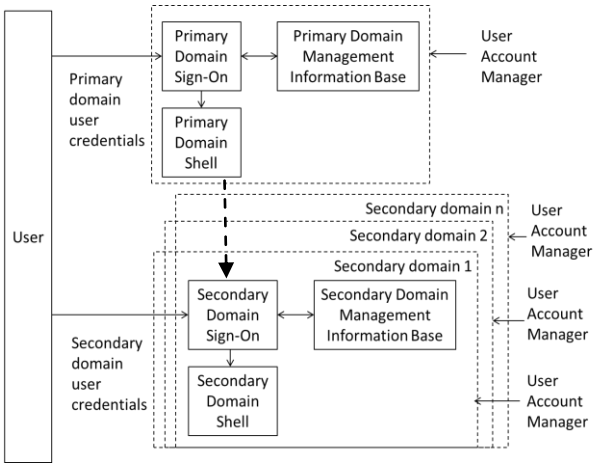
Gambar-1. Contoh LDAP DIT [1]

Rinta Kridalukmana, Program Studi Teknik Sistem Komputer, Fakultas Teknik Universitas Diponegoro Jln. Prof. Sudharto, S.H., Tembalang, semarang 50275 indonesia. Email : rintakrida@undip.ac.id

B. Manajemen User dengan Single Sign On

Oleh *opengroup*[3], *single sign on* didefinisikan sebagai mekanisme di mana hanya dengan satu aksi autentikasi seorang *user* dapat diijinkan untuk mengakses ke seluruh komputer dan sistem sesuai dengan hak aksesnya, tanpa perlu memasukkan password berulang kali.

Gambar-2 di bawah ini menunjukkan proses di mana seorang *user* memiliki beberapa *credentials* yang dipergunakan untuk dapat mengakses domain data atau aplikasi pada suatu sistem.



Gambar-2. *User Sign-On to Multiple System* [3]

Ilustrasi pada Gambar-2 menunjukkan bahwa secara historis, sistem terdistribusi memiliki komponen yang bertindak sebagai *independent security domain*. Komponen-komponen ini memiliki platform masing-masing yang berhubungan dengan sistem operasi dan aplikasi, dan bertindak sebagai domain independen dalam arti bahwa seorang *end-user* harus teridentifikasi dan terautentikasi secara independen pula ke setiap domain yang akan diakses.

Pertama-tama user akan berinteraksi dengan *primary domain* untuk membuat suatu *session* dalam *primary domain*. Untuk itu, user harus memberikan *credential* yang sesuai untuk *primary domain* tersebut. Selanjutnya untuk mendapatkan layanan di *secondary domain*, user juga diminta untuk melakukan login ulang dengan menggunakan *credential* yang sesuai untuk *primary domain* tersebut.

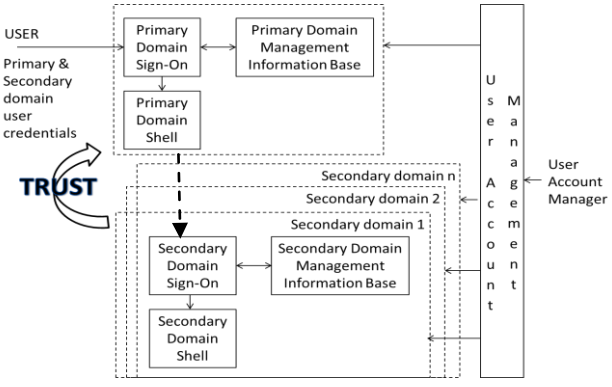
Dengan pertimbangan *usability* dan *security* menimbulkan kebutuhan untuk melakukan koordinasi yang memungkinkan untuk integrasi fungsi *user sign on* dan *user account management* mengingat semakin berkembangnya domain dalam suatu organisasi/ perusahaan. Layanan yang menyediakan integrasi dan koordinasi akan memberikan dampak *real cost benefits* pada perusahaan melalui :

- Berkurangnya waktu yang dibutuhkan user untuk operasi *sign-on* ke masing-masing domain, termasuk mengurangi kemungkinan gagal *sign-on* ke suatu domain
- Meningkatkan *security* melalui mengurangi kebutuhan user untuk mengingat sekumpulan informasi autentikasi
- Mengurangi waktu yang dibutuhkan administrator untuk menambahkan atau mengurangi user ke sistem atau memodifikasi hak akses
- Memperbaiki *security* dengan memberi kemudahan ke administrator sistem untuk memelihara integritas konfigurasi *user account* termasuk kemudahan untuk mencabut akses user individu ke seluruh sumber daya sistem dalam suatu langkah yang terkoordinasi dan konsisten.

Dengan model *single user sign-on* seperti yang terlihat pada Gambar-3, informasi yang diberikan oleh *user* sebagai bagian dari prosedur *sign-on* di *primary domain* dapat dipergunakan pada *secondary domain* melalui beberapa cara, di antaranya adalah :

- *Directly*, informasi yang diberikan *user* akan langsung dikirimkan ke *secondary domain* sebagai bagian dari *secondary domain*.
- *Indirectly*, informasi yang diberikan user dipergunakan untuk melakukan *retrieve* identifikasi user dan informasi *user credentials* yang disimpan dalam *single-sign on management information base*. Hasil dari *retrieval information* selanjutnya dipergunakan sebagai dasar untuk operasi *sign-on* di *secondary domain*
- *Immediately*, yaitu dengan membuat *session* dalam *secondary domain* ketika dilakukan *initial session* saat *sign on* ke *primary domain*.
- Informasi *user sign-on* disimpan sementara atau disimpan dalam *cache* dan dipergunakan pada saat terjadi *request* ke *secondary domain* oleh *user*.

Aspek *security* yang perlu digarisbawahi dalam model *single sign-on* adalah bahwa s*econdary domain* harus mempercayakan pada *primary domain* untuk mengidentifikasikan *user credentials* dan melakukan otentikasi serta melindungi informasi *user credentials* dari penyalahgunaan.



Gambar-3. *Single User Sign On to Multiple Services* [3]

C. Drupal, LDAP, Single-Sign-On, & External Link

Drupal merupakan salah satu *open source content management platform* yang banyak dipergunakan untuk mendukung berbagai macam aplikasi berbasis web dari blog pribadi sampai aplikasi organisasi/perusahaan. Sebagai *content management*, Drupal memberikan dukungan untuk melakukan koneksi dengan LDAP server. Dukungan ini diimplementasikan melalui penambahan modul LDAP *Integration* pada Drupal.

Modul ini memiliki fungsi untuk mengatur [4]:

- Otentikasi yang berfungsi untuk melakukan otentikasi ke server LDAP
- Data, untuk sinkronisasi profil LDAP dan Drupal
- Groups, yang mengatur *role mapping* Drupal ke LDAP *Group*.
- Single Sign-On, untuk melakukan single sign-on ldap

Selain itu, Drupal juga memiliki modul untuk membuat halaman (*page*) pada situs drupal di mana isi dari page tersebut bertipe iframe, sehingga dapat digunakan untuk menampilkan link eksternal ke dalam halaman drupal. Modul ini dapat dimanfaatkan untuk mengelola aplikasi-aplikasi dalam perusahaan dari berbagai server untuk disajikan secara terintegrasi pada halaman drupal.

Dengan modul-modul tersebut, Drupal dapat dimanfaatkan untuk mendukung *single-sign-on* berbasis ldap dan sekaligus dipergunakan untuk mengelola login aplikasi web lainnya.

III. Deskripsi Fungsional Modul Drupal Yang Akan Dikembangkan

Dari permasalahan yang telah diuraikan di atas, maka dapat dijabarkan bahwa modul drupal yang akan dikembangkan harus dapat memenuhi fungsi-fungsi sebagai berikut :

1) Modul harus dapat melakukan otentikasi ke server LDAP
2) Menampilkan *external-link* yang diasumsikan sebagai aplikasi-aplikasi yang akan dikelola dalam portal (*secondary domain*) dalam bentuk content *iframe*
3) Modul harus dapat menyimpan informasi *sign-on* pada *primary domain* dan nantinya akan dipergunakan ketika terjadi *login request* ke *secondary domain*.
4) Selain berfungsi sebagai single sign on, modul sekaligus berfungsi sebagai *single log out* dari tiap *external link secondary domain*.

Dengan melihat fungsional yang dibutuhkan, maka setidaknya terdapat 2 kebutuhan dasar yang telah dapat diakomodasi oleh modul yang telah ada. Yang pertama adalah kebutuhan terhadap otentikasi pada server LDAP, dan ini telah dapat diakomodasi pada modul *LDAP Integration*. Sedangkan untuk menampilkan *external link* sebagai *secondary domain* dalam bentuk iframe dapat dipergunakan modul *iframe page*.

Namun dengan kedua modul ini, masih belum terdapat komunikasi ketika dilakukan *sign-on* pada *primary domain* sehingga dapat dilanjutkan untuk otentikasi pada *secondary domain*. Dari sini terdapat 2 alternatif yang dapat dilakukan, yaitu membuat modul baru untuk memenuhi fungsi-fungsi yang dibutuhkan, atau memanfaatkan kedua modul.

Metode untuk terkoneksi dengan *secondary domain* pun dapat digunakan metode *directly*, *indirectly*, *immediately*, atau dengan memanfaatkan cache. Namun bila menggunakan metode *directly*, dapat terjadi ketidakefektifan dalam proses *sign-on* ke *secondary domain*. Hal terjadi ketika terdapat misalnya 5 *external link* yang dikelola dalam portal, namun pada saat tertentu mungkin hanya 2 *external link* saja yang akan digunakan. Maka dengan metode *directly*, di mana informasi *sign-on* pada *primary link* langsung dikirim ke seluruh *external link* yang merupakan *secondary domain*, akan tidak efektif karena pada saat *sign-on* pertama kali sekaligus *sign-on* ke semuanya, walaupun yang digunakan hanya 2 *external link* pada *secondary domain* saja. Demikian juga pada metode immediately, hanya saja pada metode immediately komunikasi ke seluruh *external link* pada *secondary domain* dilakukan dengan membangun session antara *primary domain* dengan *secondary domain*.

Berbeda dengan bila menggunakan metode lainnya, informasi *sign-on* pada saat pertama kali di *primary link* akan disimpan dan digunakan saat dibutuhkan yaitu ketika terjadi request ke *external link secondary domain*. Pada metode *indirectly*, informasi disimpan secara permanen untuk dapat dimanfaatkan ketika terjadi *request* oleh seluruh *external link* pada *secondary domain* yang ada, sedangkan metode lainnya adalah dengan menyimpan sementara informasi dalam cache.

Dengan melihat berbagai aspek tersebut, maka dapat dilihat bahwa implementasi termudah untuk prosedur *sign-on* pada *secondary domain* adalah dengan menyimpan sementara informasi *sign-on* pada *primary domain*. Dan bila dikaitkan dengan modul LDAP *integration* dan *iframe page* yang telah tersedia pada Drupal, implementasi dengan menyimpan sementara informasi *sign-on* akan lebih mudah dilakukan daripada harus mengembangkan modul baru untuk fungsi yang dibutuhkan yang telah diuraikan di atas.

IV. Modul LDAP Integration

Secara umum, bagian dari modul LDAP *Integration* adalah LDAP *auth* yang bertugas melakukan otentikasi informasi user pada saat *sign-on* ke server LDAP. Ketika validasi dan otentikasi berhasil, maka informasi user dari server LDAP akan disimpan dalam database user pada *primary domain*. Proses otentikasi ini merupakan bagian kritis ketika akan menyimpan

informasi user yang valid untuk dapat digunakan ketika terjadi request ke *secondary domain*.

Adapun fungsi untuk proses otentikasi LDAP pada modul LDAP integration dilakukan oleh function ldapauth_authenticate(), di mana proses utama otentikasi LDAP dituliskan seperti pada berikut ini :

```
function ldapauth_authenticate($form_values = array()) {
  global $user, $_ldapauth_ldap;
 //menangkap user password pada form input
  $name = $form_values['name'];
  $pass = trim($form_values['pass']);

  // Otentikasi LDAP user.
  if (!($dn = _ldapauth_auth($name, $pass)))
    return;
  if (!$account) {
      $userinfo = array('name' => $name, 'pass' => $pass_new,
'mail' => $mail, 'init' => $init, 'status' => 1, 'authname_ldapauth'
=> $name, 'ldap_authentified' => TRUE, 'ldap_dn' =>
$ldap_user['dn'], 'ldap_config' => $_ldapauth_ldap-
>getOption('sid'));
      $user = user_save('', $userinfo);
    }
  else {
    // Login user.
    $data = array(
      'ldap_dn' => $dn,
      'ldap_config' => $_ldapauth_ldap->getOption('sid'),
    );
    // Bila login sukses.
    // simpan data login.
    if (LDAPAUTH_LOGIN_PROCESS ==
LDAPAUTH_AUTH_MIXED &&
LDAPAUTH_SYNC_PASSWORDS)
      $data['pass'] = $pass;
      $user = user_save($account, $data);
  }

  // Simpan data user & password ke session.
  $_SESSION['namauser'] = $dn;
  $_SESSION['kodesandi'] = $pass;

}
```

Pada fungsi di atas, informasi user yang telah terotentikasi disimpan dalam variabel session namauser dan kodesandi. Variabel inilah yang nantinya akan dipergunakan sebagai informasi yang dikirimkan ke *external link* pada *secondary domain* ketika terjadi *request* ke *external link* tertentu.

V. MODUL IFRAME PAGE

Dengan modul *Iframe Page*, *external link* pada *secondary domain* akan ditampilkan seolah-olah sebagai *page* pada Drupal. Dengan cukup menuliskan link yang dituju, maka link tersebut akan ditampilkan dalam bentuk page yang sebenarnya merupakan bentuk iframe dari link tersebut. Sebagai contoh, misal server aplikasi memiliki alamat http://122.188.1.1 dan aplikasi keuangan berada pada alamat http://122.188.1.1/ aplikasi_keuangan, aplikasi administrasi pada alamat http:// 122.188.1.1/aplikasi_administrasi. Alamat-alamat masing-masing aplikasi inilah yang dituliskan pada link di modul *Iframe Page*.

Permasalahan pada *sign-on* secara konvensional adalah bahwa ketika diakses pada alamat salah satu aplikasi di atas, akan muncul *login form* untuk dapat masuk ke halaman *dashboard* aplikasi. Sehingga apabila link aplikasi tersebut dituliskan pada link di modul *Iframe Page*, keadaan *sign-on* secara konvensional juga akan terjadi. Tentu saja dengan demikian proses *single-sign-on* tidak akan bekerja. Oleh karena itu, sebagai solusi, tiap aplikasi harus memiliki fungsi login dengan otentikasi ke LDAP server. Sebagai permisalan fungsi ini dituliskan dalam file login.php. sehingga link login untuk aplikasi dapat dituliskan http:// 122.188.1.1/aplikasi_administrasi/login? username=<<username>>&&password=<<password>>. Jika login berhasil, maka akan mengarah ke *dashboard aplikasi.* Demikian juga untuk aplikasi-aplikasi yang lain. Dengan skenario seperti ini, maka proses *single-sign-on* dapat dilakukan dengan menuliskan link fungsi login pada aplikasi pada modul iframe page. Sedangkan parameter <<username>> dan <<pasword>> didapatkan dari variabel namauser dan kodesandi yang disimpan sebagai session variabel pada modul *LDAP Integration*.

Untuk melakukan render terhadap link yang diisikan pada modul *iframe page* untuk menggantikan parameter <<username>> dan <<password>> maka perlu dilakukan modifikasi pada function theme_iframe_page(), di mana inti dari proses tersebut dapat dituliskan seperti berikut ini:

```
function theme_iframe_page_iframe($node) {
  $params = ($node->send_get_parameters) ? $_GET : array();

//Parse link url pada iframe page untuk mendapatkan nama
parameter user dan password
$QUERYVAR= parse_url($urltes, PHP_URL_QUERY);
$GETVARS = explode('&',$QUERYVAR);
$i=0;
foreach($GETVARS as $string){
    list($is,$what) = explode('=',$string);
    if($i==0){
       $username = $is;
       $paramuname = $what;
    }
    else{
      $password = $is;
      $parampwd = $what;
     }
    $i++;
}

//menyusun kembali link url dengan melakukan penggantian
string <<username>> dan <<password>> dengan variabel
session namauser dan kodesandi

$url = parse_url($node->url, PHP_URL_SCHEME). '://'.
parse_url($node->url, PHP_URL_HOST) .parse_url($node->url,
PHP_URL_PATH). '?' . $username . '=' .
$_SESSION['namauser']; . '&' . $password . '=' .
$_SESSION['kodesandi'];

//membentuk iframe dengan sumber link url yang telah disusun
kembali
 return '<iframe src="' . $url . '" height="' . $node->height . '"
width="' . $node->width . '" frameborder="0"></iframe>';
}
```

Dengan proses tersebut di atas, secara tidak langsung modul ini telah melakukan login secara otomatis dengan mengirimkan informasi user dan password ke *external link* pada *secondary domain* dan akan langsung mengarah ke halaman *dashboard* aplikasi yang di-*request*.

VI. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Dari penelitian ini, hal-hal yang dapat disimpulkan adalah sebagai berikut :

1) Drupal sebagai salah satu solusi *content management system* dapat dimanfaatkan untuk mengelola *single-sign-on* dan manajemen aplikasi berbasis web
2) Metode pengiriman informasi *sign-on* dari *primary domain* ke *secondary domain* lebih efektif dengan cara menyimpan sementara karena dikirimkan hanya kepada *external link* pada *secondary domain* yang di-*request* saja.
3) Untuk proses *single-sign-on* berbasis LDAP dan manajemen aplikasi *single-sign-on* pada Drupal versi 6 dapat memanfaatkan modul LDAP *Integration* dan *Iframe Page*.

B. *Saran*

1) Link login pada tiap aplikasi yang dituliskan pada modul *iframe page* hendaknya diberi nama alias sehingga yang akan muncul pada url adalah alamat alias dari link login aplikasi
2) Bila dipergunakan untuk manajemen aplikasi, hendaknya fungsi *register new user* pada Drupal dihilangkan dari form login.

REFERENSI

[1] Gerald Charter, *LDAP System Administration*, O'Reilly Media Inc., 2003

[2] Howes Tmothy A., Smith Mark C., and Good Gordon S., *Understanding and Deploying LDAP Directory Services*, Second Edition, Pearson Education Inc., 2003

[3] Jogiyanto HM. (2005), *Sistem Informasi Strategik untuk Keunggulan Kompetitif*, Andi Yogyakarta, 319-335

[4] Matt Scafidi-McGuire, *LDAP Integration*, http://drupal.org/project/ldap_ integration, 2005

[5] Mikkel Hogh, IFrame Page, http://drupal.org/project/iframe_page, 2010

Rinta Kridalukmana, dilahirkan di Semarang, Indonesia, pada tahun 1977. Mendapatkan gelar Sarjana Komputer dari jurusan Sistem Informasi Universitas Stikubank Semarang, pada tahun 2003 dan gelar magister dari Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, pada tahun 2007. Saat ini aktif menjadi dosen di program studi Teknik Sistem Komputer Universitas Diponegoro sejak tahun 2011. Bidang penelitian yang digeluti adalah : Sistem Informasi dan Desktop Application.