

**IMPLEMENTASI SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON
BERBASIS LDAP**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh:

Nama : Nur Uswatun Hasanah

No. Mahasiswa : 12 523 041

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2016

**IMPLEMENTASI SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON
BERBASIS LDAP**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh:

Nama : Nur Uswatun Hasanah

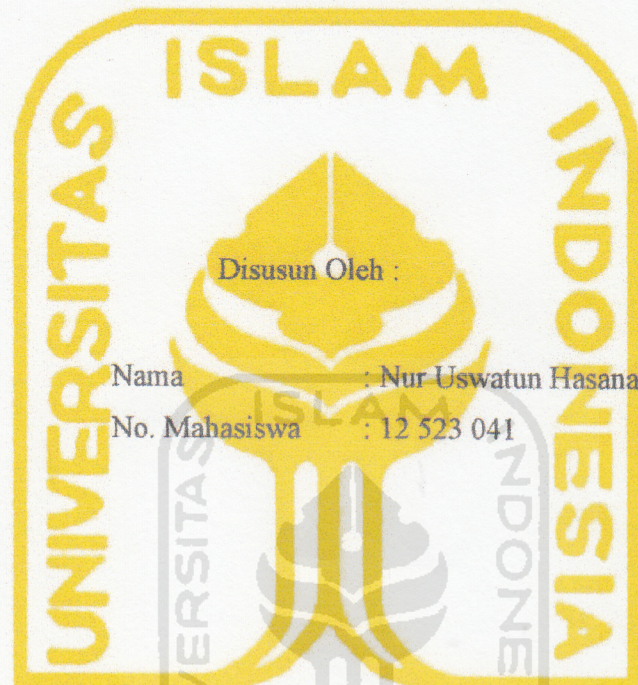
No. Mahasiswa : 12 523 041

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2016

LEMBAR PENGESAHAN PENGUJIAN
IMPLEMENTASI SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON
BERBASIS LDAP

TUGAS AKHIR



Yogyakarta, 21 November 2016

Dosen Pembimbing

(Yudi Prayudi, S.Si., M.Kom)

LEMBAR PENGESAHAN PENGUJIAN
IMPLEMENTASI SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON
BERBASIS LDAP

TUGAS AKHIR

Disusun Oleh :

Nama : Nur Uswatun Hasanah

No. Mahasiswa : 12 523 041

Telah Dipertahankan di Depan Sidang Pengujian Sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas Teknologi
Industri Universitas Islam Indonesia

Yogyakarta, 5 Desember 2016

Tim Penguji,

Yudi Prayudi, S.Si., M.Kom

Ketua

Ahmad Luthfi, S.Kom., M.Kom.

Anggota I

Hamid, S.T, M.Eng

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Hamid, S.T., M.Eng.)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini,

Nama : Nur Uswatun Hasanah

NIM : 12 523 041

Tugas Akhir dengan judul :

IMPLEMENTASI SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON BERBASIS LDAP

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 21 November 2016



(Nur Uswatun Hasanah)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini,

Nama : Nur Uswatun Hasanah

NIM : 12 523 041

Tugas Akhir dengan judul :

IMPLEMENTASI SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON BERBASIS LDAP

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 21 November 2016

(Nur Uswatun Hasanah)

HALAMAN PERSEMBAHAN

Alhamdulillah Rabbii 'alamin segala puji bagi Allah yang telah memberikan kemampuan dalam menyelesaikan tugas akhir. Puji syukur juga kepada Allah telah diberikan sedikit ilmu yang mendukung penyelesaian tugas akhir. Shalawat serta salam saya panjatkan juga kepada Nabi besar kita Nabi Muhammad SAW yang telah memberi syafaat kepada umat manusia hingga akhir zaman.

Skripsi ini saya persembahkan kepada kedua orang tua saya Bapak Indratmoko Prihantoro dan Ibu Rina Tri Susilowati yang telah mengerahkan seluruh kemampuannya untuk mendukung kesuksesan anaknya. Terima kasih atas segala dorongan semangat, nasihat, dukungan, pembelajaran pola pikir, serta telah mengusahakan yang terbaik untuk anaknya.

Terima kasih kepada bapak pembimbing saya, Bapak Yudi Prayudi S.Si, M.Kom selaku dosen pembimbing yang memberikan bimbingan dalam melakukan Tugas Akhir dan mengajarkan saya akan kemandirian dalam belajar. Terima kasih atas diberikannya saran dan kritik yang membangun dari pencarian judul hingga proses pengerjaan Tugas Akhir selesai. Semoga suatu saat nanti saya sempat membalas budi baik Bapak. Kalaupun tidak, semoga budi baik bapak dibalas oleh Sang Khalik.

Terima kasih kepada sahabat-sahabat yang telah mendukung saya dan memberi semangat dalam penyelesaian tugas akhir ini, baik selama berjalannya tugas akhir maupun sebelum dan sesudah pelaksanaannya. Terima kasih kepada Mayang yang selalu memberi nasihat positif, terima kasih kepada Dina yang menemani selama mengerjakan skripsi, terima kasih kepada Debi dan Danti yang telah memberi semangat dalam bentuk verbal maupun tindakan. Terimakasih kepada sahabat saya Eno dan Tika yang telah memberi saya motivasi karena telah lulus duluan. Terimakasih kepada sahabat saya Astri, Icha, Sharen, Nisa, Reza yang setia menemani dan mewarnai hari sejak SMP. Terimakasih kepada Yola yang telah banyak memberikan banyak pelajaran hidup yang berharga. Semoga kita semua dapat bertemu lagi ditempat yang berbeda dengan keadaan yang lebih baik.

Terimakasih kepada Arif Ardiansyah dan Arif Wahyudi yang telah banyak membantu saya dalam mengerjakan tugas akhir tersebut, semoga amal kebaikan kalian dibalas oleh yang maha kuasa dan ilmu yang kalian miliki dapat menolong kalian dalam keadaan sulit.

Skripsi ini saya persembahkan kepada TNI yang telah memberi saya kesempatan untuk mengaplikasikan ilmu saya meskipun ilmu saya sangat terbatas. Saya akan memanfaatkan kesempatan ini sebaik mungkin.



MOTTO

MAN JADDA WAJADA

Siapa yang bersungguh-sungguh pasti berhasil

MAN SHABARA ZHAFIRA

Siapa yang bersabar pasti beruntung

Allah kekasih hati yang paling utama

Adukan semua masalah kepada-Nya karena pertolongan Allah sangat dekat

Lebih baik mandi keringat dalam latihan daripada mandi darah dalam perang

Kesuksesan hanya dapat diraih ketika kita mampu untuk menyelesaikan.

Dan semua usaha akan terselesaikan bilamana kita bersungguh-sungguh.

Tanpa usaha kita tidak dapat untuk mencapai semua kesuksesan.

KATA PENGANTAR



Assalamu 'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah Robbil 'Alamin. Puji syukur kehadiran Allah SWT yang telah memberikan kenikmatan atas segalanya. Shalawat serta salam kepada nabi kita Muhammad SAW, sehingga atas berkat dan rahmat-Nya tugas akhir ini dapat terselesaikan dengan judul **“PENERAPAN SAML 2.0 PADA OTENTIKASI SINGLE SIGN ON BERBASIS LDAP”**. Laporan tugas akhir ini disusun sebagai salah satu syarat memperoleh gelar sarjana pada jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penyelesaian tugas akhir ini tidak lepas dari saran, bimbingan, serta dukungan dari berbagai pihak. Untuk itu ucapan terima kasih disampaikan kepada:

1. Kedua orang tua, Bapak (Indratmoko Prihantoro) dan Ibu (Rina Tri Susilowati). Semoga bapak dan ibu senantiasa berada dalam lindungan Allah SWT.
2. Dr. Ir. Harsoyo, M.Sc selaku Rektor Universitas Islam Indonesia.
3. Dr. Drs. Imam Djati Widodo M.Eng.Sc. selaku Dekan Fakultas Teknologi Industri.
4. Bapak Yudi Prayudi, S.Si, M.kom, selaku Dosen Pembimbing yang senantiasa memberikan arahan selama penyusunan Tugas Akhir.
5. Bapak Hendrik, ST., M.Eng., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri.
6. Keluarga besar yang senantiasa selalu mendukung dan mendorong untuk berbuat yang terbaik.
7. Adik tersayang yang selalu memberikan *support* dan mendoakan (Farzan Sirajuddin, Karin Leony Jamila, Bimo Bramantyo).
8. Keluarga Berencana adalah sahabat yang selalu memberikan saran dan dukungan. Sehingga dapat terselesaikan dengan baik.
9. Sahabat-sahabat di Yogyakarta Mayang, Dina, Debi, Danti, Septi, dan semua teman-teman yang ada di Jogja. Terima kasih atas semua kesabaran

dan pengertian dan kebaikan kalian.

10. Terimakasih kepada sahabat – sahabat saya yang telah mendukung saya dan memberi semangat dalam penyelesaian tugas akhir ini, baik selama berjalannya Tugas Akhir maupun sebelum dan sesudah pelaksanaannya. Terima kasih kepada Mayang yang selalu memberi nasihat positif. Terimakasih kepada Dina yang menemani selama mengerjakan skripsi. Terima kasih kepada Debi dan Danti yang telah memberi semangat dalam bentuk verbal maupun tindakan. Terima kasih kepada sahabat saya Eno dan Tika yang telah memberi saya motivasi karena telah lulus duluan. Terima kasih kepada sahabat saya Astri,Icha,Sharen,Nisa,Reza yang setia menemani dan mewarnai hari sejak SMP. Terimakasih kepada Yola yang telah banyak memberikan banyak pelajaran hidup yang berharga. Semoga kita semua dapat bertemu lagi di tempat yang berbeda dengan keadaan yang lebih baik.
11. Terima kasih juga kepada teman-teman lain yang tidak dapat disebutkan satu persatu.

Penyusunan Laporan Tugas Akhir ini, tentu tidak lepas dari kesalahan. Oleh karena itu saran dan kritik untuk kebaikan kedepan sangat diharapkan. Semoga Laporan Tugas Akhir ini dapat bermanfaat.

Wassalamu 'alaikum Wr.Wb.

Yogyakarta, 21 November 2016

Nur Uswatun Hasanah

SARI

Sebuah perusahaan harus merencanakan segala sesuatunya secara matang agar memperoleh keuntungan perusahaan yang maksimal. Otentikasi dan otorisasi merupakan bagian penting dalam jaringan heterogen yang ada di sebuah perusahaan seperti ini karena setiap pengguna memiliki wewenang berbeda agar pengguna dapat mengakses resource yang sesuai dengan wewenangnya. *Single Sign-On (SSO)* merupakan sebuah protokol jaringan yang terkait akses kontrol pada berbagai layanan terhubung dalam sebuah sistem. *SSO* memungkinkan *user* melakukan satu kali log-in untuk mengakses keseluruhan layanan dalam sistem yang diperlukannya. Salah satu *protocol SSO* adalah *SAML* atau *Security Assertion Markup Language* yaitu sebuah *framework XML* untuk pertukaran informasi otentifikasi dan otorisasi dan informasi lain dengan token mengandung pernyataan untuk menyampaikan informasi pengguna antara otorisasi *SAML*. *LDAP* adalah sebuah protokol perangkat lunak pengatur mekanisme pengaksesan layanan direktori yang berisi informasi user. Dalam kasus ini *LDAP* digunakan untuk menyimpan data informasi dari direktori yang akan diakses.

Kata Kunci : *Single Sign On(SSO), Service Provider, Identity Provider, SAML, LDAP*

TAKARIR

SAML	= <i>Secure Assertion Markup Language</i> .
SSO	= Otentikasi dan otorisasi dengan pintu masuk tunggal.
Credential	= Melakukan log-in dengan <i>username</i> dan <i>password</i> untuk mendapatkan hak akses.
Localhost	= <i>Host local</i> yang hanya bisa diakses dalam komputer <i>user</i> .
Protocol	= Aturan.
LDAP	= Protokol berbasis <i>tree</i> pencari direktori aktif.
Client	= Pengguna yang menerima layanan.
Server	= Komputer yang memberikan layanan ke klien.
Service Provider	= <i>Vendor</i> penyedia layanan.
Identity Provider	= Protokol yang berisi definisi informasi dari <i>user</i> .
Virtual Mecine	= Mesin virtual yang dapat dijalankan di sebuah komputer guna mensimulasikan sistem yang ingin kita bangun

DAFTAR ISI

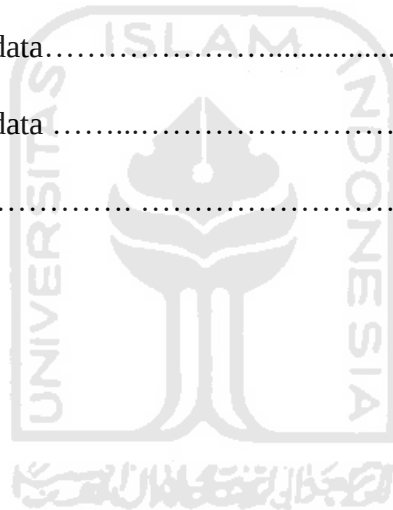
HALAMAN JUDUL	
LEMBAR PENGESAHAN PENGUJIAN	iii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PERSEMBAHAN	vi
MOTTO	viii
KATA PENGANTAR	ix
SARI	xi
TAKARIR	xii
DAFTAR GAMBAR	xv
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	4
1.7. Sistem Penulisan Laporan	5
BAB II	7
LANDASAN TEORI	7
2.1. Keamanan Komputer	7
2.2. Otentikasi dan Otorisasi	8
2.3. Single Sign-On (SSO)	9
2.3.1 Pengertian SSO	9
2.3.2 Manfaat SSO	11

2.4.	Secure Assertion Markup Language (SAML).....	12
2.5.	XML	13
2.6.	Light Weight Direktoary Access Protokol (LDAP)	14
2.7.	INTERNET PROTOCOL (IP)	14
2.8.	VMware	15
BAB III		17
METODOLOGI PENELITIAN		17
3.1	Gambaran Umum Sistem.....	17
3.1.1	Tahapan Perancangan.....	18
3.2.1	Perancangan Arsitektur Perancangan Alur Kerja SAML	18
BAB IV		33
HASIL DAN PEMBAHASAN		33
4.1	Hasil Pengujian Sistem.....	33
BAB V		2
KESIMPULAN DAN SARAN		2
5.1	Kesimpulan.....	2
5.2	Saran	2
DAFTAR PUSTAKA		3
LAMPIRAN		4

DAFTAR GAMBAR

Gambar 2.1 Otentikasi tanpa SSO	9
Gambar 2.2 Otentikasi dengan SSO	10
Gambar 3.1 Gambaran umum kerja SAML	16
Gambar 3.2 Gambaran alur kerja SAML	17
Gambar 3.3 Gambaran arsitektur logis SAML.....	18
Gambar 3.4 Gambaran SAML rquest-reply	19
Gambar 3.5 Hierarki pohon ldap.....	20
Gambar 3.6 Setting ldap.....	21
Gambar 3.7 OU pada LDAP.....	21
Gambar 3.8 CN pada people LDAP	22
Gambar 3.9 CN pada people LDAP	22
Gambar 3.10 CN pada people LDAP	23
Gambar 3.11 CN pada people LDAP	23
Gambar 3.12 Install Slapd.....	25
Gambar 3.13 Install Slapd.....	25
Gambar 3.14 Install Phpldapadmin.....	25
Gambar 3.15 Config LDAP.....	26
Gambar 3.16 Localhost LDAP.....	27
Gambar 3.17 Instalasi SAML.....	27
Gambar 3.18 Tool SAML	28
Gambar 3.19 Metadata.....	28
Gambar 3.20 Setting Domain Name.....	29
Gambar 3.21 Setting virtual host.....	29

Gambar 3.22 Setting Idp.....	30
Gambar 3.23 Setting S.....	30
Gambar 3.24 Konfigurasi SAML dan LDAP.....	31
Gambar 4.1 LDAP berhasil dijalankan.....	33
Gambar 4.2 User LDAP	34
Gambar 4.3 SAML berhasil dijalankan.....	34
Gambar 4.4 Test autentikasi.....	35
Gambar 4.5 Test autentikasi	35
Gambar 4.6 SAML login.....	35
Gambar 4.7 SAML metadata.....	36
Gambar 4.8 SAML metadata	36
Gambar 4.9 Binding data.....	37



BAB I

PENDAHULUAN

1.1. Latar Belakang

Sebuah perusahaan pada era globalisasi menuntut mereka harus merencanakan segala sesuatunya secara matang agar memperoleh keuntungan perusahaan yang maksimal. *Enterprise Resource Planning (ERP)* adalah sebuah konsep pengelolaan sumber daya perusahaan yang berpengaruh luas dari *management* paling atas hingga operasional di sebuah perusahaan bisa dimanfaatkan dengan optimal. Proses bisnis pada ERP sebagian besar merupakan aliran data yang berjalan dari satu domain ke domain lainnya atau satu aplikasi menuju ke aplikasi lainnya.

Otentikasi dan otorisasi merupakan sebuah keharusan yang dilakukan dalam mengakses aplikasi dan domain yang diperlukan dalam jaringan heterogen seperti ini karena setiap pengguna memiliki wewenang berbeda agar pengguna dapat mengakses *resource* yang sesuai dengan wewenangnya. Sebuah perusahaan yang memiliki jaringan *heterogen* menggunakan banyak vendor berbeda pada aplikasi dan OS nya. Pengguna diminta mengisi informasi mengenai dirinya pada saat mengakses platform yang berbeda. Untuk mendapatkan otentikasi dan otorisasi pada *software* tersebut *user* harus memasukkan *credential*. Jika banyak aplikasi maka banyak *credential* yang diperlukan. Sistem otentikasi terpusat merupakan solusi yang tepat. Di sinilah peran penting Single Sign On (SSO).

Browser berbasis Single Sign On menggantikan solusi konvensional berdasarkan beberapa kredensial, domain spesifik dengan menawarkan pengalaman pengguna yang lebih baik. *User* melakukan *log in* tunggal dalam operasi untuk penyedia identitas dan belum dapat mengakses sumber daya yang ditawarkan berbagai penyedia layanan. Selain itu, dengan mengganti beberapa kredensial (satu per penyedia layanan) dengan satu tunggal (terkait dengan penyedia identitas), solusi SSO diharapkan untuk meningkatkan keamanan secara keseluruhan sebagai pengguna cenderung menggunakan password yang lemah dan / atau menggunakan kembali password yang sama di berbagai penyedia jasa. Single Sign-On (SSO) merupakan sebuah protokol jaringan yang terkait akses kontrol pada berbagai layanan terhubung dalam sebuah sistem. SSO memungkinkan *user* melakukan satu kali log-in untuk mengakses keseluruhan

layanan dalam sistem yang diperlukannya. Menggunakan *session* atau proses autentikasi *user* yang memungkinkan *user* untuk menyediakan sebuah *credential* sekali dengan maksud untuk mengakses banyak aplikasi (Aaslund et al, 2007).

Sistem berbasis direktori harus menyediakan kemampuan yang tinggi dengan mereplikasi penyimpanan *credential*. Untuk menyediakan fleksibilitas yang lebih baik, sebuah *single sign on* harus menyediakan baik *server* terpusat dan metode dari penyimpanan *credential*. *Lightweight Directory Access Protocol (LDAP)* didesain untuk meng-*update* dan mencari direktori yang berjalan lewat jaringan *TCP/IP*. Banyak pengembang seperti Microsoft dengan Active Directory, Novell dengan Novell® eDirectory™ dan Netscape dengan Netscape Directory Server semua telah mengambil LDAP sebagai sebuah standar untuk direktori servis. Karena penggunaan yang luas dari LDAP direktori, sebuah produk *single sign on* harus menyediakan dukungan *built-in* untuk LDAP sehingga produk ini dapat bekerja secara efektif dengan infrastruktur modern seperti sekarang (Anonymous, 2006, Novell Inc).

SAML atau Security Assertion Markup Language yaitu sebuah *framework* dari bahasa XML untuk pertukaran informasi otentifikasi dan otorisasi dan informasi lain yang dibutuhkan. Dasarnya adalah menggunakan token yang mengandung pernyataan untuk identitas yang portabel sebagai alternative untuk menyampaikan informasi pengguna antara otorisasi SAML. SAML 2.0 memungkinkan skenario otentifikasi dan otorisasi berbasis web termasuk *cross-domain single sign-on*. SAML mendesain *protocol* otentikasi yang nantinya akan dijalankan oleh LDAP.

Berdasarkan dari pembahasan di atas, akan dicoba medesain sebuah sistem otentikasi *single sign-on* menggunakan SAML 2.0 dengan aplikasi *server* yang diimplementasikan pada aplikasi *vmware* dan akan disimulasikan pada sebuah sistem otentikasi sebuah jaringan *enterprise heterogen*. Sehingga akan dibuat dan disusun Penelitian dengan judul Implementasi SAML 2.0 pada Otentikasi Single Sign on berbasis LDAP.

1.2. Rumusan Masalah

Merujuk dari latar belakang yang ada, maka ditemukanlah rumusan masalah sebagai berikut:

- a) Bagaimana merancang desain penerapan protokol SAML untuk SSO ?
- b) Bagaimana simulasi implementasi penerapan SAML ?
- c) Bagaimana kinerja dari implementasi SAML ?

1.3. Batasan Masalah

Batasan masalah dalam Tugas Akhir ini adalah:

- a) Teknik virtualisasi *server computing* akan diimplementasikan dalam server Vmware 14.04
- b) SAML akan dibuat di sistem operasi Ubuntu 14.04.
- c) Berfokus pada bagaimana menjalankan SAML dan menjalankan LDAP.
- d) Penelitian ini berfokus pada bagaimana design SAML, implementasinya kemudian diuji bagaimana kinerja dari SAML tersebut.
- e) Tidak dibahas tentang *psycal security, social engineering, offline attack*, virus serta trojan dalam penggunaan purwarupa.
- f) Sistem ini merupakan simulasi contoh kasus bukan penerapan pada suatu perusahaan sebenarnya.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui bagaimana mendesain dari *protocol* otentikasi menggunakan SAML 2.0 kemudian mensimulasikan implementasinya pada *vmware*. Di situ kita akan melihat bagaimana SAML dengan bantuan LDAP sebagai *datastore* melakukan otentikasi *single sign on*.

Setelah itu akan diuji bagaimana kinerja dari proses otentikasi, apakah sesuai dengan desain yang dibuat sebelumnya atau tidak. Kemudian akan diambil kesimpulan dari percobaan yang dilakukan.

1.5. Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian ini adalah:

- a) Untuk dapat mengetahui bagaimana desain SAML 2.0
- b) Untuk dapat mengetahui bagaimana simulasi dari implementasi SAML 2.0 dalam melakukan otentikasi *single sign-on* (SSO).
- c) Perguruan tinggi dapat menambahkan wawasan bagi para pembaca, khususnya yang berminat untuk mengembangkan teknologi tersebut.

- d) Bagi pembaca dapat memberikan mobilitas untuk dapat mencari informasi secara langsung pada jurusan yang ada di fakultas dengan menggunakan sistem yang otentikasi.

1.6. Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini sebagai berikut :

- a) Studi Pustaka
Studi pustaka bertujuan untuk memperkaya informasi dan menambah wawasan dari sumber yang dapat dipertanggungjawabkan mengenai apa itu SSO, SAML, dan beberapa informasi pendukung lainnya.
- b) Persiapan
Persiapan ini bertujuan untuk menyiapkan hal-hal yang dibutuhkan untuk menyelesaikan pembuatan implementasi ini. Dimulai dari instalasi *service provider, identity provider, client* yang akan melakukan SSO, LDAP serta instalasi paket paket lain yang dibutuhkan.
- c) Implementasi
Setelah semua paket yang dibutuhkan telah siap maka dilakukan *setting* paket agar semua protocol dapat saling berhubungan dalam menjalankan *single sign-on*.
- d) Pengujian
Setelah tahap-tahap sebelumnya sudah dilakukan semua, maka dapat dikatakan bahwa SAML dapat diuji. Apakah SAML bisa melakukan otentikasi beberapa web server secara lancar atau belum. Tahap pengujian ini bertujuan untuk mendapatkan hasil dari konfigurasi yang sudah dilakukan pada tahap sebelumnya. Pengujian disimulasikan pada sebuah virtual mesin.
- e) Evaluasi
Tahap evaluasi diadakan bertujuan untuk mendapatkan kesimpulan bagaimana kinerja dari SAML dalam melakukan otentikasi SSO yang dilakukan dari tahap pengujian. Dengan harapan dapat memperbaiki hal-hal yang masih diperlukan dan diadakan pengembangan untuk kedepannya.
- f) Penarikan Kesimpulan

Dari hasil pengujian dan analisis data, maka dilakukan kesimpulan untuk menunjukkan hasil akhir secara keseluruhan dalam Tugas Akhir ini.

g) **Penulisan Laporan**

Pada tahap ini, semua hasil analisis dan penarikan kesimpulan dikumpulkan dalam bentuk laporan, agar penjelasan setiap bagian tersusun lebih struktural dan sistematis

1.7. Sistem Penulisan Laporan

Untuk memudahkan dalam membaca dan memberikan gambaran secara singkat mengenai permasalahan yang akan dibahas pada setiap babnya, maka dibuatkanlah sistematika penulisan dalam laporan Tugas Akhir. Sistematika penulisan ini dibagi menjadi 5 (lima) bab, adapun sistematika penulisannya sebagai berikut :

- a) Bab I Pendahuluan. Bab ini menjelaskan mengenai latar belakang yang membuat dilakukannya implementasi ini dan juga menjelaskan mengenai rumusan masalah, batasan masalah, manfaat penelitian, tujuan penelitian, dan metodologi penelitian.
- b) Bab II Landasan Teori. Bab ini berisi tentang beberapa teori yang berhubungan dengan Tugas Akhir, meliputi pengertian dari setiap layanan-layanan yang digunakan dalam sistem, beberapa bahasan dari penelitian sebelumnya, serta beberapa perangkat pendukung yang akan berperan dalam menyelesaikan Tugas Akhir.
- c) Bab III Metodologi. Dalam bab ini akan dibahas mengenai analisis kebutuhan yang nantinya akan digunakan dalam praktik Tugas Akhir, meliputi analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, serta perancangan sistem.
- d) Bab IV Hasil dan Pembahasan. Pada bab ini akan dibahas mengenai sistem yang dibuat, bagaimana pengimplementasiannya. Dan juga akan dibahas mengenai instalasi paket-paket yang digunakan dalam sistem dan pengujian sistem terhadap client.
- e) Bab V Kesimpulan dan Saran. Bab ini akan memuat kesimpulan dari Tugas Akhir dan merupakan bab penutup yang akan merangkum hasil dari pada

Tugas Akhir. Dan akan disertakan Saran dengan harapan penelitian dapat dikembangkan dikemudian hari.



BAB II

LANDASAN TEORI

2.1. Keamanan Komputer

Keamanan komputer pertama kali diimplementasikan secara sederhana dengan cara pengamanan fisik. Keamanan akan terancam apabila digunakan oleh orang yang tidak berwenang. Keamanan komputer dilakukan dengan cara menyediakan keamanan fisik berupa pembatasan akses bagi semua orang. Hanya beberapa individu tertentu saja yang berwenang. Dalam era jaringan global seperti sekarang ini, keamanan komputer melibatkan lebih banyak hal lain.

Ada 3 (tiga) tujuan utama dari keamanan komputer, yaitu:

- a) *Confidentiality* atau kerahasiaan sebuah data. Mensyaratkan bahwa data yang tersimpan pada sistem komputer, juga data yang ditransmisikan antar sistem komputer, hanya bisa diakses oleh individu yang sah saja.
- b) *Integrity* atau keaslian data. Hal ini memastikan bahwa data yang dikirim sama dengan data yang diterima. Dengan kata lain data yang tersimpan pada sistem komputer, juga data yang ditransmisikan antar sistem komputer, terbebas dari modifikasi atau penghapusan yang dari yang tidak berhak, juga mencakup dari pembuatan data oleh yang tidak berhak. Proses pembuatan, perubahan dan penghapusan data; oleh karena itu; hanya boleh dilakukan oleh pihak yang berwenang saja.
- c) *Availability*, bertujuan memastikan bahwa *authorized users* dari sistem komputer dan media komunikasi mendapatkan akses ketika memerlukannya. Tujuan ini juga terkait dengan konsep *denial of services* yang disebabkan penurunan kinerja sistem. Namun ini tidak terkait penurunan kinerja secara normal pada saat beban puncak operasional, tetapi lebih spesifik terkait aksi dari para penyerang untuk mendapatkan akses ke dalam sistem sebagai *authorized users*.

Otentikasi merupakan hal yang perlu diperhatikan, hanya *user* yang berwenanglah yang dapat mengakses sistem tersebut. *User* yang melakukan otentikasi memiliki wewenang dalam memastikan kerahasiaan sebuah data serta keasliannya.

2.2. Otentikasi dan Otorisasi

Otentikasi adalah sebuah tindakan dimana sistem memvalidasi apakah *user* ini punya wewenang dalam mengakses sistem tersebut atau tidak. Barisan pertahanan pertama pada sistem komputer dari para penyusup adalah melalui sistem otentifikasi. Sistem otentifikasi tersebut mencegah para *user* yang tidak sah dari memperoleh akses terhadap sistem dengan cara melakukan validasi ketika hendak menggunakan sistem. Validasi ini biasanya berupa penggunaan *password*. Metode otentifikasi lain dapat berupa penggunaan kunci fisik atau karakteristik unik dari fisik *user* seperti sidik jari.

Penggunaan otentifikasi tidak serta merta berarti bahwa *user* yang tidak memiliki hak akses tidak bisa mendapatkan akses dari sistem. Sudah menjadi hal yang cukup umum ketika *authorized user* menjadi lalai dalam menjaga kunci akses mereka. Dengan demikian, *unauthorized users* bisa mendapatkan akses ke dalam sistem dengan cara menyamar sebagai *user* yang terotorisasi. Hal ini dimungkinkan karena sistem tidak mengotentifikasi identitas dari seorang *user*, namun hanya pada *user* bertindak sebagai siapa. Karena sistem tidak dapat melakukan verifikasi identitas sebenarnya dari *user*, metode-metode yang digunakan harus diarahkan untuk mengurangi kemungkinan seorang *user* yang tidak sah untuk mengakses sistem.

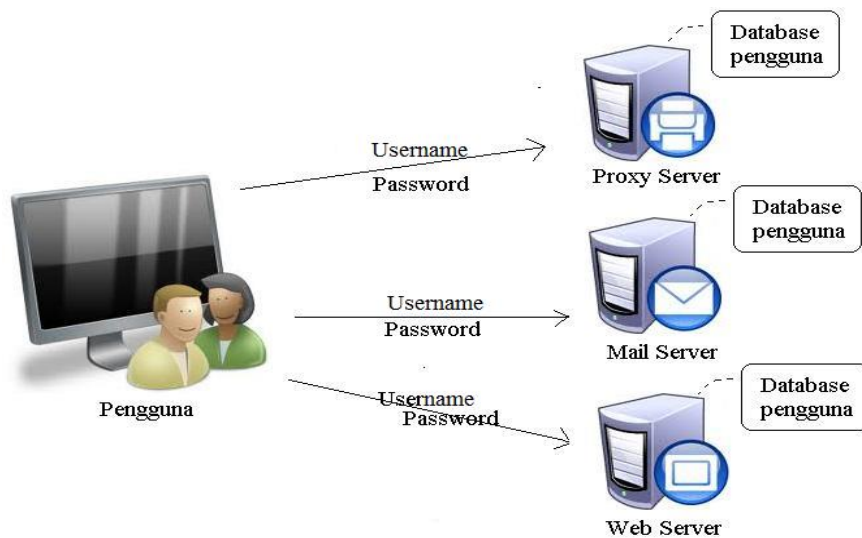
Pada sebuah perusahaan yang bersifat heterogen dimana aplikasi maupun sistem operasinya berasal dari banyak vendor maka *user* pun juga memerlukan lebih dari satu *credential*. Hal ini kurang begitu efektif. Disamping memakan waktu yang banyak, hal ini juga berdampak pada *database* yang digunakan, tentu akan memakan banyak memori yang banyak untuk peng-input-an *database credential user*. Disamping itu bila terjadi kesalahan seperti lupa *password* dan sebagainya, admin kesulitan mengidentifikasinya. Dengan SSO otentikasi menjadi lebih efektif.

2.3. Single Sign-On (SSO)

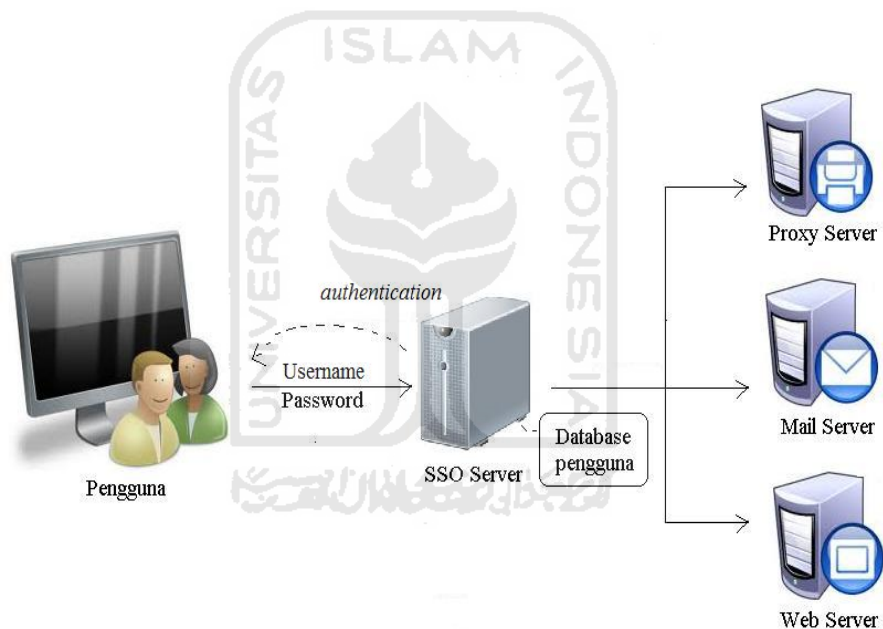
2.3.1 Pengertian SSO

SSO atau *Single Sign-on* adalah sebuah mekanisme yang membuat *user* hanya perlu mengingat satu *username* dan *password* yang otentik untuk membuka beberapa layanan sekaligus, (Priyo Puji Nugroho, 2012). Dari situ dapat disimpulkan bahwa Single Sign-on adalah merupakan suatu cara yang memungkinkan agar *user* dapat melakukan otentikasi dan otorisasi pada berbagai *resource* dari banyak vendor dalam suatu jaringan enterprise hanya dengan menggunakan satu *credential*. SSO juga memperkenalkan pertukaran informasi otentikasi antar vendor dan mengidentifikasi subjek secara ketat guna menghindari log-in ganda pada sistem atau kelompok sistem yang terpercaya. Berikut merupakan gambaran secara umum perbedaan antara otentikasi menggunakan SSO dan otentikasi tanpa menggunakan SSO.

Single sign on (SSO) mengautentikasi *user* untuk mengakses semua aplikasi yang telah di-*authorized* untuk diakses. Ini menghilangkan permintaan authentication berulang ketika *user* memakai aplikasi berbeda. *User* akan terotentikasi dan terotorisasi secara otomatis selama *session* masih berlaku, jika *session* habis maka *user* akan terputus koneksinya. SSO meminimalisir resiko kemungkinan pengguna terblokir dan dapat menekan biaya IT dari segi perawatan data informasi user, dan memaksimalkan waktu yang bisa digunakan untuk bekerja. Inti dari solusi SSO berbasis browser terletak protokol otentikasi berbasis browser. Tiga peran ambil bagian dalam protokol yaitu klien, penyedia identitas (IdP) dan penyedia layanan (SP). Tujuan dari klien yang biasanya web browser dipandu oleh pengguna, adalah untuk mendapatkan akses ke layanan atau sumber daya yang disediakan oleh SP. IdP mengotentikasi klien dan *protocol* terkait pernyataan otentikasi. Akhirnya, SP menggunakan pernyataan yang dihasilkan oleh IdP untuk menentukan hak klien untuk sumber daya yang diminta.



Gambar 2.1 Otentikasi tanpa SSO



Gambar 2.2 Otentikasi menggunakan SSO

Dari gambar di atas terlihat jelas bahwa otentikasi menggunakan SSO lebih simple dibandingkan melakukan otentikasi menggunakan cara manual. *Username* dan *password* dari user terpusat dalam *database server* SSO. SSO juga dapat memusatkan pengelolaan dari parameter sistem yang relevan pada saat bersamaan serta meningkatkan penggunaan *resource* secara keseluruhan, disamping itu juga mengurangi resiko yang bisa ditimbulkan dari kesalahan manusia, bagian terpenting dari kesalahan sistem yaitu kesalahan dalam mengkombinasi *password* dan *username*.

Otentikasi menggunakan SSO mengharuskan *user* untuk mengakses *service provider*. *Service provider* berisi portal yang akan diakses oleh *user*. Kemudian dari *service provider* *user* di *direct* ke *identity provider* yang berisi informasi dari *user* yang diperlukan dalam melakukan otentikasi. Jika data *user* valid *user* di *direct* kembali ke *service provider* yang selanjutnya *user* bisa mendapatkan akses dalam setiap portal yang ingin diakses.

Keamanan protocol SSO berbasis browser ini bergantung pada sejumlah asumsi mengenai kepercayaan dari klien dan *protocol* keamanan transpost yang digunakan untuk bertukar pesan. Seperti yang akan kita bahas secara lebih rinci nanti, IdP yang dipercaya. Tapi kita tidak berasumsi bahwa semua SP membawa klien yang mungkin memainkan protokol dengan dipercaya. Secara khusus, kita bertanya kepada diri sendiri, apakah SP berbahaya dapat mengganggu keamanan sesi protokol. Kami juga menganggap bahwa komunikasi antara klien dan SP, serta antara klien dan IdP, yang dibawa SSL unilateral atau koneksi TLS. Patut diperhatikan, dan fakta ini akan memainkan peran penting kemudian, bahwa pesan tunggal antara klien dan SP dapat dikirim melalui SSL / TLS koneksi yang berbeda, sejauh standar dan implementasi khas prihatin. Mencoba untuk menegaskan bahwa semua pesan tersebut selalu dikirim melalui SSL koneksi / TLS yang sama cukup sulit dalam prakteknya, seperti yang akan kita bahas nanti. Konsekuensinya bahwa SP yang berbahaya memang akan dapat mempengaruhi keamanan sesi protokol SAML dari klien dengan SP lainnya.

2.3.2 Manfaat SSO

Manfaat dari SSO antara lain yaitu :

- a) Pengguna tidak perlu mengingat banyak *username* dan *password*. Cukup dengan satu *credential* untuk mendapatkan izin akses terhadap semua layanan aplikasi yang tersedia.
- b) Kemudahan memproses data, hanya perlu satu kali proses data pada *server database back-end* nya.
- c) Tidak perlu membuat data pengguna yang sama pada setiap aplikasi karena langsung terhubung dengan server database yang terhubung

langsung dengan *back-end server* nya maka pengguna akan valid dalam pengaksesan seluruh layanan.

- d) Menghemat biaya pemeliharaan *password*. Ketika harus melakukan *reset*, pengelola layanan tidak perlu menghasbikan waktu dan *bandwith* untuk menemukan data *credential* pengguna.

Ada banyak *protocol* yang dibisa digunakan untuk melakukan penerapapan SSO. Salah satu diantaranya adalah SAML.

2.4. Secure Assertion Markup Language (SAML)

SAML SSO menyediakan standar terbuka dan mudah diatur sesuai dengan kebutuhan spesifikasi dari skenario aplikasi. Dalam tugas akhir ini akan difokuskan pada salah satu instansiasi yang paling banyak digunakan. Diprakarsai SSO dengan POST Bindings, yang lebih lanjutnya dijelaskan pada OASIS. Dalam tugas akhir ini kita akan merujuk kasus ini digunakan sebagai *kasus penggunaan SAML SSO* dan protokol yang terkait sebagai *SAML SSO*. SAML adalah sebuah aturan pokok dalam mengatur kerangka informasi dengan berbasis XML yang dikeluarkan oleh perusahaan OASIS. Berfungsi sebagai pertukaran otorisasi, atribut serta pernyataan otorisasi. Dalam penerapan SAML, keamanan informasi ditransfer oleh SAML. Standar SAML mendefinisikan peraturan secara akurat, lengkap semantik dan mendefinisikan kriteria untuk permintaan, merespon serta mendefinisikan penggunaan pernyataan. Model manajemen identitas berdasarkan SAML terdiri dari bagian - bagian berikut:

- a) Pernyataan dan Protokol

Mendefinisikan sintaks dan semantik dari peraturan format XML, *protocol* akses dan protokol respon. Terdapat tiga protokol utama pada SAML yaitu :

1. Otoritas otentikasi
Yaitu penentuan hak untuk menjalankan kekuasaan dalam melakukan otentikasi.
2. Atribut pernyataan
yaitu sebuah pernyataan yang membawa data penting dari atribut subjek.

3. Pernyataan akses otorisasi

yaitu sebuah pernyataan yang membawa data penting tentang kebutuhan otorisasi. Misal keputusannya membolehkan subjek mengakses sumber daya yang dituju.

b) Binding dan Dasar

Binding adalah sebuah pernyataan yang mendefinisikan tipe *port* yang menjadi sebuah protokol transport kongkret dan format data didalamnya. SAML mengirim permintaan dan respon pesan ke protokol komunikasi menggunakan pemetaan SOAP.

c) Konsistensi norma

Merupakan standar yang harus dipenuhi SAML. Hal ini dilakukan untuk meningkatkan kapabilitas suatu sistem.

d) Keamanan dan kerahasiaan

Pada bagian ini berisi penjelasan mengenai risiko keamanan dari arsitektur SAML serta bagaimana menangani risiko ini dan beberapa risiko yang tidak dapat diselesaikan.

2.5. XML

XML adalah sebuah bahasa markup lanjutan HTML yang dapat digunakan untuk pertukaran data dalam berbagai ragam sistem maupun beragam platform. XML terdiri dari dua yaitu :

- a) (Danny,Joe,Liam, 2012)XML *Encryption* yaitu memberikan keamanan pada tingkat pesan SOAP ,standar untuk bertukar pesan antar program dalam satu sistem operasi dengan basis XML.
- b) XML *Signature* mendefinisikan sintaks dan aturan pengolahan untuk mewakili tanda tangan *digital*. *Signatures* XML menyediakan integritas dan otentikasi. Seperti XML *Encryption*, XML *Signature* dapat menandakan bagian-bagian tertentu dari *file* XML. Hal ini relevan ketika data yang dihasilkan, diedit atau dilihat oleh beberapa pengguna, dan ketika integritas bagian tertentu dari data yang harus dipastikan, tetapi bagian-bagian lain mungkin masih dapat diubah.

2.6. Light Weight Direktory Access Protokol (LDAP)

Sukmana (2006), dalam artikel ilmukomputer.com yang berjudul “Pengenalan LDAP”, secara garis besar menjelaskan protokol yang memungkinkan seseorang mencari *resource* di sekitarnya. LDAP adalah sebuah protokol yang mengatur mekanisme pengaksesan layanan direktori (*Directory services*). Direktori ini biasanya digunakan untuk mendeskripsikan banyak informasi seperti *user*, organisasi, aturan, layanan dan lain sebagainya. LDAP menggunakan model *client-server*. *Client* mengirimkan data kepada server menggunakan protokol TCP/IP kemudian server akan mencarinya pada *Directory Information Tree* (DIT) yang tersimpan pada server. LDAP menggunakan hierarki dengan struktur pohon yang dilengkapi dengan SASL (Simple Authentication and Secure Layer) untuk memastikan apakah seorang *user* berhak atau tidak dalam mengakses sebuah direktori. Pada studi kasus yang akan dilakukan penulis, LDAP hanya berperan sebagai database store, untuk otentikasinya akan digunakan protokol SAML.

2.7. INTERNET PROTOCOL (IP)

Secara sederhana IP merupakan standar komunikasi data yang digunakan oleh *user* dalam melakukan proses transaksi tukar menukar data dari suatu titik komputer ke komputer yang lainnya dalam suatu jaringan. Agar jaringan internet berlaku, semestinya harus ada suatu standar yang diperlukan untuk mengaturnya untuk itulah diperlukan sebuah protokol internet yang akan menyamaratakan aturan standarisasi pengalamatan dalam suatu jaringan. IP terbagi kedalam 2 versi, ada IPv4 dan IPv6.

IPv4 menggunakan 32 bit dimana setiap 8 bit dipisahkan oleh titik. Sedangkan dalam IP itu sendiri ada beberapa komponen yang berfungsi sebagai identifikasi alamat, diantaranya adalah :

a) IP address

IP address digunakan sebagai identifikasi pengalamatan *interface* jaringan pada host dari suatu jaringan komputer. IP address diibaratkan sebagai alamat beserta nomer rumah agar Pak Pos dapat dengan mudah mengirimkan pakatnya. Dengan IP address tiap host dapat terhubung satu

dengan lainnya dan dapat saling bertukar informasi maupun paket melalui media transmisi. Pembagian IP address dibedakan menjadi 2 bagian yaitu ipv4 dan ipv6. Ipv4 terdiri dari 32 bit yang dibagi menjadi 4 bagian yang masing-masing terdiri dari atas 8 bit, sedangkan ipv6 terdiri dari 128 bit. Pada penelitian ini peneliti menggunakan ip dinamis dimana ip didapat dari vmware.

b) Subnet mask

Subnet mask adalah suatu istilah mengacu kepada angka biner 32 bit yang digunakan untuk membedakan network ID dengan host ID. *Subnet mask* menunjukkan apakah host tersebut terletak dalam satu jaringan atau berbeda. Network ID adalah bagian dari sebuah IP yang menunjukkan di jaringan mana komputer tersebut berada. Host ID menunjukkan host TCP/IP yang ada dalam sebuah jaringan.

c) Gateway

Gateway merupakan perangkat jaringan yang digunakan untuk menghubungkan dua jaringan berbeda sehingga bisa saling bertukar informasi.

d) Domain Name System (DNS)

DNS sederhananya dapat diartikan sebagai translator antara nama domain dengan alamat ip dari domain tersebut sehingga manusia dan sistem komputer dapat berkomunikasi dengan baik.

2.8. VMware

VMware merupakan aplikasi virtual PC dimana bisa menjalankan sistem operasi lain di dalam sistem operasi yang dijalankan secara bersamaan. Banyak fitur yang diberikan oleh VMware. VMware terbagi menjadi dua yaitu *VMware Workstation* dan *VMware server*.

Virtual server adalah sebuah virtualisasi server yang bertujuan untuk virtualisasi fungsional server, data center atau storage sistem. *VMware Workstation* adalah virtualisasi PC yang dimana membuat PC secara virtual namun bisa difungsikan layaknya PC yang sebenarnya.

2.9 Metadata

Metadata adalah informasi terstruktur yang mendefinisikan, menjelaskan, menemukan atau menjadikan suatu informasi menjadi mudah untuk ditemukan kembali. Metadata biasanya merupakan informasi mengenai sebuah informasi. Misal sebuah data teks maka metadata menjelaskan berapa ruas teks tersebut, berapa panjangnya kemudian apa tipe data dari data tersebut. Secara detail fungsi dari metadata yaitu :

- a) Membuat sumberdaya bisa ditemukan dengan menggunakan kriteria yang relevan.
- b) Mengidentifikasi sumberdaya.
- c) Mengelompokkan sumberdaya yang serupa.
- d) Membedakan sumberdaya yang tak miliki kesamaan.
- e) Memberikan informasi lokasi.

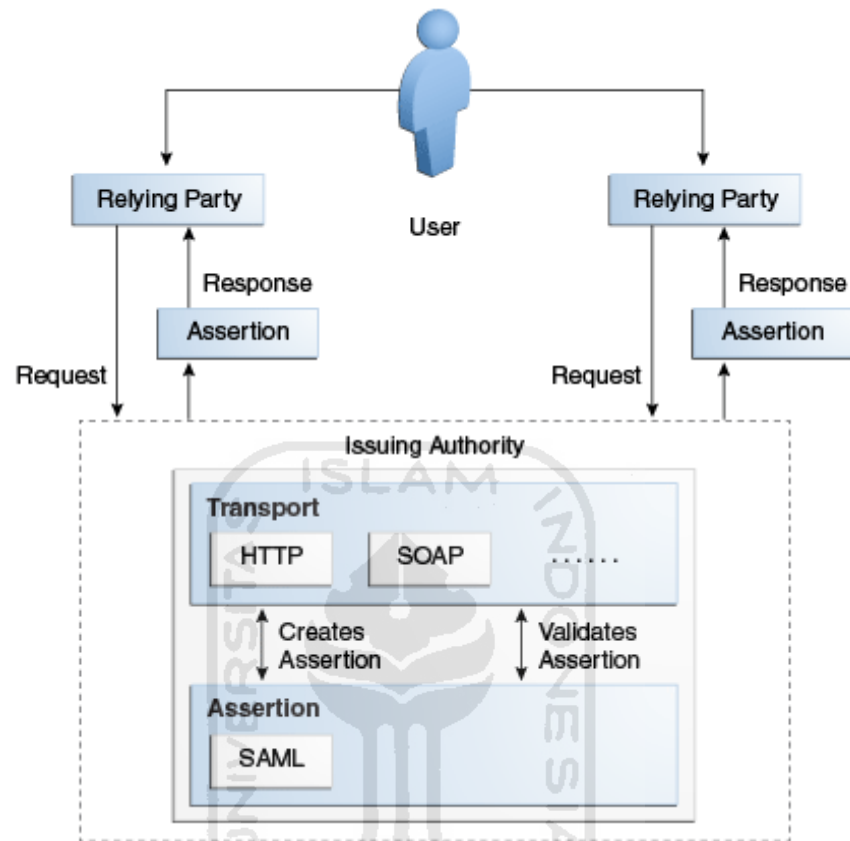
Terdapat tiga jenis utama metadata:

- a) *Metadata deskriptif* menggambarkan suatu sumberdaya dalam maksud seperti penemuan dan identifikasi. *Metadata deskriptif* bisa meliputi elemen semisal judul, abstrak, pengarang, dan kata kunci.
- b) *Metadata structural* menunjukkan bagaimana kumpulan objek disusun secara bersama-sama menjadi satu, semisal bagaimana halaman-halaman ditata untuk membentuk suatu bab.
- c) *Metadata administrative* menyediakan informasi untuk membantu mengelola sumberdaya, semisal terkait kapan dan bagaimana suatu informasi diciptakan, tipe dokumen dan informasi teknis lainnya, serta siapa yang bisa mengaksesnya.

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Sistem

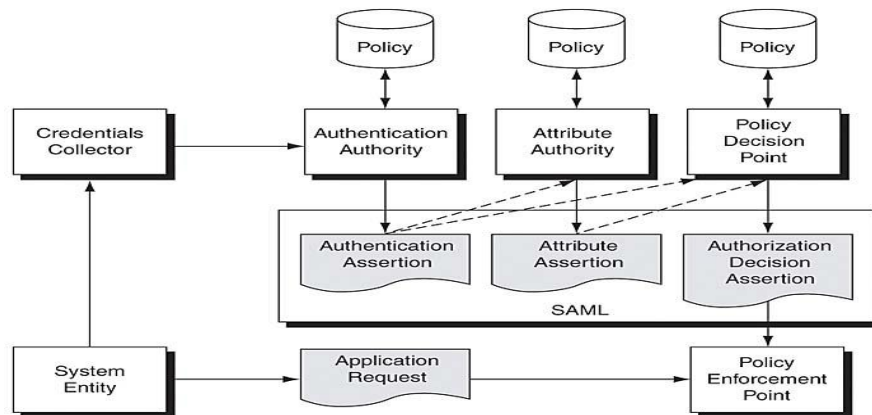


Gambar 3.1 Gambaran sistem secara garis besar

Secara garis besar desain kerja pada SAML dijelaskan pada gambar di atas. Terdapat tiga komponen pada SAML yaitu *service provider*, *identity provider* dan *user*, masing masing dari komponen tersebut memiliki fungsi berbeda. Seorang *user* yang ingin mengakses sebuah portal kemudian dilarikan ke *service provider*. *Service provider* yang mengadakan layanan. Disini *user* akan diidentifikasi apakah dia memiliki hak untuk mengakses *resource* tersebut atau tidak. Dari *service provider* *user* dilanjutkan menuju *identity provider*. Di dalam *identity provider* *user* dikenai pengecekan apakah dia sudah log-in atau belum. Data tersebut ada di sebuah *datastore* milik LDAP. Jika *user* belum melakukan log-in maka idp akan meneruskan *user* ke *browser* yang kemudian memunculkan *form login*. Namun sebaliknya, jika *user* sudah tercatat log-in maka idp melanjutkan ke *browser* yang diinginkan oleh *user*.

3.1.1 Tahapan Perancangan

3.1.1.1 Perancangan Arsitektur Perancangan Alur Kerja SAML

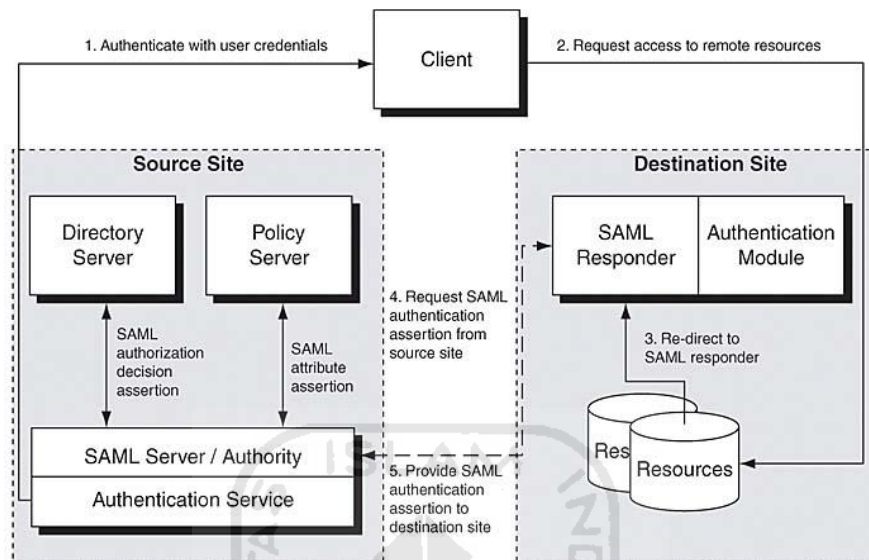


Gambar 3.2 Alur kerja SAML

Sebuah entitas sistem yang ingin mengirimkan permintaan aplikasi untuk mengakses sumber daya sistem akan dialihkan ke entitas yang menyajikan *credential collector* yang akan mengotentikasikan dengan otentikasi pernyataan, atribut pernyataan serta otorisasi keputusan kemudian entitas itu akan melakukan request aplikasi. Sebelum entitas sistem tersebut diberikan akses, otorisasi keputusan akan memproses permintaan aplikasi berdasarkan hak akses yang sudah diatur sebelumnya. Semua request terwakili dalam SAML. Permintaan akses dari user berasal dari sistem *entry point* dan dialihkan ke otoritas berbeda dimana permintaan tersebut akan dikonfirmasi kemudian dilanjutkan untuk dieksekusi. Dalam arsitektur SAML, model domain SAML menggambarkan entitas informasi dan peranan mereka tetapi tidak menyerupai komponen infrastruktur tingkat server direktori. Dengan demikian arsitek dan pengembang perlu memetakan domain ke dalam komponen arsitektur logis.

Klien (*user*) meminta sumber daya remot dibawah *single sign on*. Situs tujuan dan situs resource bersamaan menyediakan keamanan bagi pengaksesan SSO menggunakan SAML. Hal ini membuat SAML menerapkan SAML Responder (agen SAML-enabled) yang dapat mencegat permintaan aplikasi untuk *resource* dan melakukan permintaan pernyataan SAML. *Resource site* telah membangun layanan otentikasi, server direktori (Atribut Authority yang menyimpan atribut kebijakan), dan server kebijakan (penentu hak dari seorang user). SAML Server memproses permintaan pernyataan SAML dan merespon

SAML Responder. Entitas domain ini dapat dengan mudah memetakan ke komponen arsitektur keamanan. Komponen arsitektur ini dapat mengambil lebih dari satu peran misalnya, modul otentikasi dapat bertindak sebagai *Authentication Authority* dan pernyataan kebijakan. Berikut adalah arsitektur logis dari SAML.



Gambar 3.3 Arsitektur logis SAML

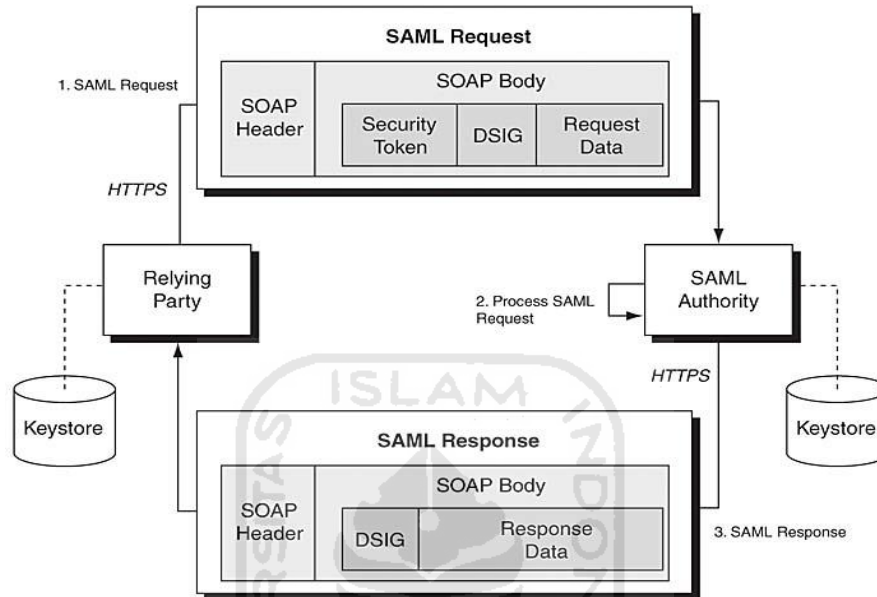
Berikut penjelasan dari gambar di atas yaitu :

- Klien telah dikonfirmasi dengan layanan otentikasi yang ditawarkan oleh situs sumber.
- Klien membuat permintaan aplikasi ke sumber terpicil di situs tujuan. Situs tujuan memiliki responder SAML (agen SAML-enabled) yang menggunakan modul otentikasi untuk menghasilkan pernyataan otentikasi.
- LDAP mengarahkan permintaan aplikasi ke responden SAML.
- SAML responder mengeluarkan permintaan penegasan otentikasi SAML ke situs sumber
- Layanan otentikasi SAML-enabled proses permintaan otentikasi pernyataan SAML dan memberikan respon terhadap situs tujuan.

Sekarang modul otentikasi dari situs tujuan tahu bahwa klien sudah dikonfirmasi. Ini tidak akan memerlukan klien untuk log-in kembali lagi. Situs tujuan akan memulai permintaan penegasan atribut SAML dan penegasan

permintaan keputusan otorisasi untuk situs sumber untuk menentukan apakah klien berhak untuk mengakses sumber daya jarak jauh.

Setelah otoritas SAML menerima permintaan, itu akan memproses permintaan dan menciptakan respon yang tepat termasuk laporan pernyataan SAML. Berikut adalah gambaran dari model request reply dari SAML :

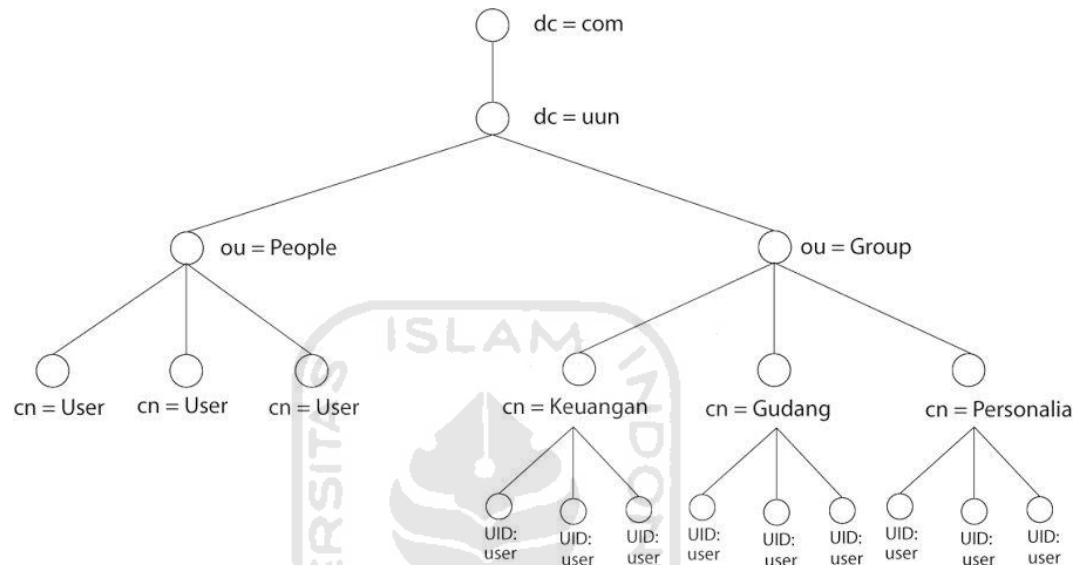


Gambar 3.4 SAML request-reply Model

SAML menggunakan model permintaan balasan untuk melaksanakan interaksi antara SAML responder dan otorisasi. SAML ini biasanya menggunakan protocol SOAP. Data permintaan dibungkus dalam SOAP dan dikirim menggunakan HTTP. Permintaan SAML terdiri dari amplop dan tubuh SOAP yang berisi permintaan SAML. Permintaan SAML elemen dapat mengandung unsur *AuthenticartionQuery*, *AttributQuerriy*, atau *AuthorizationDecisionQuery*. Dari permintaan *request user*, *request* dikirim dengan basis HTTP menuju SAML *request*. Di dalam SAML *request* terdapat sebuah struktur data yang terdiri dari *header* dan *body*. Pada *body* terdapat token keamanan yang berisi aturan keamanan kemudian mengharuskan *user* untuk melakukan tanda tangan digital setelah melakukan *request data*. *User* diverifikasi berdasarkan *protocol* yang telah dirancang oleh SAML. Setelah itu *user* mengotentikasi keaslian datanya. Jika data yang dikonfirmasi sesuai dengan yang ada pada *datastore* maka permintaan *user* dilanjutkan.

3.1.1.2 Perancangan Hierarki LDAP

Pada dasarnya LDAP merupakan sebuah protocol layanan direktori yang menerapkan sistem hierarki pohon. Layanan direktori bisa dikatakan sebagai database tempat penyimpanan data yang terkait dengan objeknya. Seperti database pada umumnya namun LDAP menyimpan data. Adapun hierarki yang akan diterapkan pada server LDAP adalah sebagai berikut :



Gambar 3.5 Hierarki Pohon LDAP

Secara prinsip, struktur *database directory service* yang digunakan oleh LDAP adalah hierarki pohon. Suatu *directory service* akan memiliki item yang akan dijadikan sebagai *root*. Sebuah *root* secara umum ditunjukkan dengan variabel *Domain Component* (DC) atau bisa juga dengan menggunakan *Organization Unit* (OU). Pada penelitian ini digunakan *domain name* dengan nama *uun.com* sebagai *root*. Kemudian pada titik *leaf* biasanya akan berisi node akhir berupa *User ID* (uid), atau *Common Name* (cn), atau bisa juga *Organization Unit* (OU). Dibawah *root* penulis menggunakan *Organization Unit* dengan dua kelompok yaitu *group* dan *people*. Pada *group* terdapat *Common Name* dengan pengelompokan direksi pada perusahaan seperti direksi keuangan, direksi personalia dan direksi gudang. Di bawahnya lagi terdapat *leaf Common Name* yang berisi dari user masing-masing direksi. Jadi, apabila ingin mendapatkan informasi mengenai *user1* pada objek keuangan maka dapat dituliskan

pengalamatan “dn=uid=user1, cn=keuangan, ou=group, dc=uun, dc=com”.

Konsep yang seperti inilah yang digunakan oleh directory service LDAP.

```
root@server:/home/uun# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <dc=uun,dc=com> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# uun.com
dn: dc=uun,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: uswatun-company
dc: uun
```

Gambar 3.6 Setting LDAP

```
# admin, uun.com
dn: cn=admin,dc=uun,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# groups, uun.com
dn: ou=groups,dc=uun,dc=com
objectClass: organizationalUnit
objectClass: top
ou: groups

# people, uun.com
dn: ou=people,dc=uun,dc=com
objectClass: organizationalUnit
objectClass: top
ou: people
```

Gambar 3.7 OU dengan nama group dan people

```

# users, people, uun.com
dn: cn=users,ou=people,dc=uun,dc=com
gidNumber: 501
cn: users
objectClass: posixGroup
objectClass: top

# nur uswatun, people, uun.com
dn: cn=nur uswatun,ou=people,dc=uun,dc=com
cn: nur uswatun
givenName: nur
gidNumber: 501
homeDirectory: /home/users/nuswatun
sn: uswatun
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uidNumber: 1000
uid: nuswatun

```

Gambar 3.8 CN pada people

```

# uswatun hasanah, people, uun.com
dn: cn=uswatun hasanah,ou=people,dc=uun,dc=com
cn: uswatun hasanah
givenName: uswatun
gidNumber: 501
homeDirectory: /home/users/uhasanah
sn: hasanah
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uidNumber: 1001
uid: uhasanah

# coba, uun.com
dn: ou=coba,dc=uun,dc=com
objectClass: organizationalUnit
objectClass: top
ou: coba

# search result
search: 2
result: 0 Success

```

Gambar 3.9 CN pada pople

```
# personalia, groups, uun.com
dn: cn=personalia,ou=groups,dc=uun,dc=com
gidNumber: 503
cn: personalia
objectClass: posixGroup
objectClass: top

# gudang, groups, uun.com
dn: cn=gudang,ou=groups,dc=uun,dc=com
gidNumber: 504
cn: gudang
objectClass: posixGroup
objectClass: top

# budi dodo, people, uun.com
dn: cn=budi dodo,ou=people,dc=uun,dc=com
cn: budi dodo
givenName: budi
gidNumber: 503
homeDirectory: /home/users/bdodo
sn: dodo
objectClass: inetOrgPerson
objectClass: posixAccount
```

Gambar 3.10 CN pada people

```
# rama sinta, people, uun.com
dn: cn=rama sinta,ou=people,dc=uun,dc=com
cn: rama sinta
givenName: rama
gidNumber: 504
homeDirectory: /home/users/gudang/ang-rsin
sn: sinta
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uidNumber: 1003
uid: gudang-rsin

# search result
search: 2
result: 0 Success

# numResponses: 13
# numEntries: 12
```

Gambar 3.11 CN pada people

Gambar 3.11 merupakan representasi dari *ldap server* di dalam *vmware*. LDAP yang digunakan memiliki root *uun.com*. Di bawah *root* digunakan *Organization Unit* dengan dua kelompok yaitu *group* dan *people*. Terdapat user *nur uswatun*, *nur hasanah* dan *coba* sebagai user pada *cn people*.

LDAP Pada *group* terdapat *Common Name* dengan pengelompokan direksi pada perusahaan seperti direksi keuangan, direksi personalia dan direksi gudang. Di bawahnya lagi terdapat *leaf Common Name* yang berisi dari user masing-masing direksi. Masing-masing *user* hanya bisa mengakses *source* yang diatur oleh idp. Sebagaimana dijelaskan juga pada gambar 3.11, Rama Sinta masuk di dalam *group* gudang.

3.1.1.3 Arsitektur Perancangan Perangkat dalam Implementasi SAML

Akan dirancang sebuah desain dari SAML dalam melakukan otentikasi. Infrastruktur ini akan dijalankan menggunakan *vmware* 14.04 dengan sistem operasi *Ubuntu* 14.04 dengan *ip* yang digunakan adalah *localhost*. Penulis menjalankan SAML menggunakan aplikasi *simpleSAMLphp*. *SimpleSAMLphp* adalah sebuah aplikasi *php* yang bekerja di beberapa *protocol* dan mendukung mekanisme otentikasi, *simpleSAMLphp* bertindak sebagai *identity provider*.

Pada dasarnya *protocol* SAML mempunyai tiga komponen yaitu *client*, *service provider* dan *identity provider*. Pada aplikasi *simpleSAMLphp*, menyediakan sebuah metadata yang mengatur komunikasi antara *service provider* dan *identity provider*. Metadata pada *identity provider* tersebut berisi informasi user yang diperlukan oleh sistem serta *protocol* untuk mengakses *credential collector*.

3.1.1.4 Konfigurasi Perangkat Lunak

a) Konfigurasi LDAP

Langkah pertama yang dilakukan adalah menginstall paket-paket ldap seperti gambar dibawah ini

```
root@ldap:~# apt-get install slapd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libdb4.2 libltdl3 libperl5.8 libslp1
  odbcinst1debian1 unixodbc
Suggested packages:
  openslp-doc slpd ldap-utils libct1 libmyodbc
  odbc-postgresql
The following NEW packages will be installed:
  libdb4.2 libltdl3 libperl5.8 libslp1
  odbcinst1debian1 slapd unixodbc
0 upgraded, 7 newly installed, 0 to remove and 0 not up
graded.
Need to get 2873kB of archives.
After this operation, 7283kB of additional disk space w
ill be used.
Do you want to continue [Y/n]? █
```

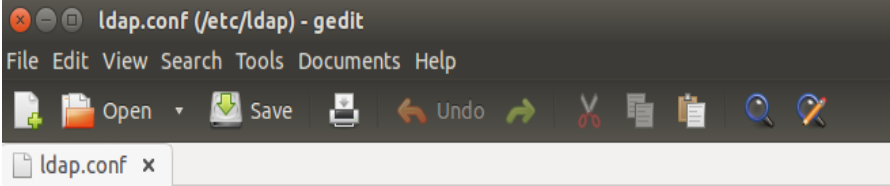
Gambar 3.12 Install slapd

```
root@ldap:~# apt-get install ldap-utils
Reading package lists... Done
Building dependency tree... 0%
```

Gambar 3.13 Install ldap-utils

```
root@ldap:~# apt-get install phpldapadmin
Reading package lists... Done
Building dependency tree... 50%
```

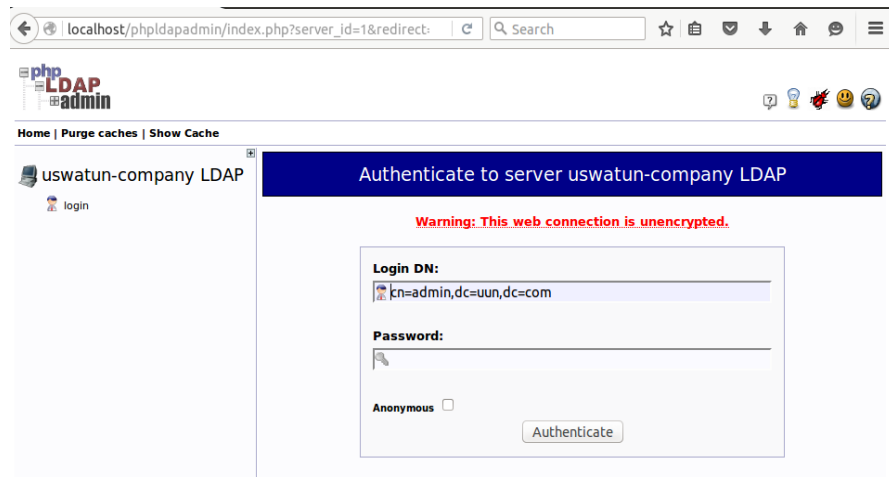
Gambar 3.14 Install phpldapadmin



```
#  
# LDAP Defaults  
#  
  
# See ldap.conf(5) for details  
# This file should be world readable but not world writable.  
  
BASE    dc=uun,dc=com  
URI      ldap://172.0.1.1      ldap://172.0.1.1:666  
  
#SIZELIMIT    12  
#TIMELIMIT    15  
#DEREF        never  
  
# TLS certificates (needed for GnuTLS)  
TLS_CACERT    /etc/ssl/certs/ca-certificates.crt
```

Gambar 3.15 Config LDAP

Seperti yang dijelaskan pada gambar 3.15, akan dilakukan instalasi beberapa paket agar bisa menjalankan LDAP. Pertama, menginstall slapd kemudian apache 2 sebagai web server dari LDAP, php 5 untuk tampilan dari aplikasi LDAP tersebut selanjutnya MySQL untuk database yang akan digunakan menyimpan data *user* serta phpldapadmin sendiri. Jika semua paket sudah terinstall maka dilakukan konfigurasi domainnya dan alamat IP yang bisa digunakan untuk mengakses. Di sini digunakan domain uun.com dengan ip localhost yang berada pada jaringan 172.0.1.1 dengan port 666 agar ip yang didapatkan selalu statis sehingga mudah dalam melakukan konfigurasi selanjutnya. Jika semua sudah terinstall maka dilakukan pengecekan dengan cara membuka *web browser* seperti gambar 3.15.

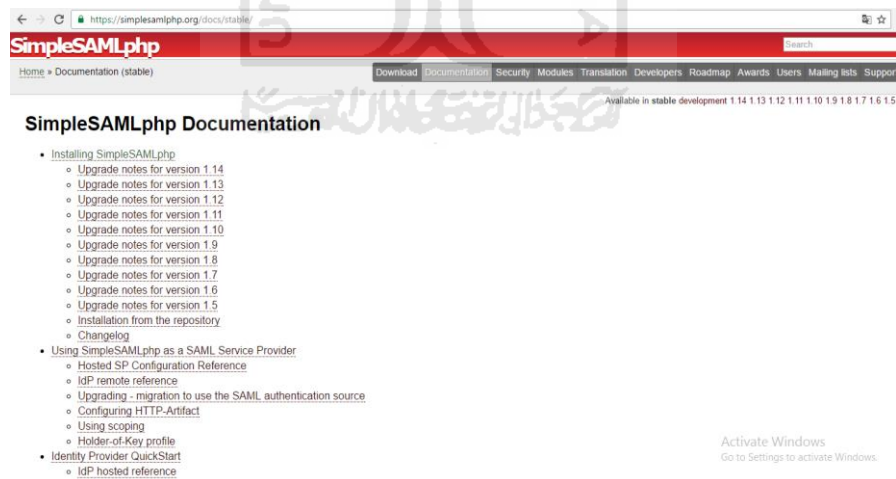


Gambar 3.16 Localhost LDAP

Ldap dengan nama uswatun-company siap digunakan sebagai *datastore* implementasi SAML berbasis LDAP. Pengaksesannya di dalam localhost dan dengan alamat phpldapadmin.

b) Konfigurasi SAML

Di sini menggunakan *SimpleSAMLphp* sebagai aplikasi yang akan digunakan untuk menjalankan SAML. Langkah pertama yang dilakukan adalah menginstall paket-paket ldap seperti gambar di bawah ini :



Gambar 3.17 Instalasi SAML

Penggunaan tutorial yang ada di *simpleSAMLphp.org* adalah untuk instalasi tool yang akan digunakan setelah semua paket terinstall. Gambar di bawah ini adalah paket yang sudah terinstall dan siap digunakan.

drwxr-xr-x	2	root	root	4096	Des	2	23:14	attributemap
drwxr-xr-x	2	root	root	4096	Des	2	23:14	bin
drwxr-xr-x	2	root	root	4096	Des	4	16:14	cert
-rw-r--r--	1	root	root	2300	Des	2	23:15	composer.json
-rw-r--r--	1	root	root	62225	Des	2	23:16	composer.lock
drwxr-xr-x	2	root	root	4096	Des	6	14:30	config
drwxr-xr-x	2	root	root	4096	Des	2	23:14	config-templates
-rw-r--r--	1	root	root	8049	Des	2	23:14	CONTRIBUTE.md
-rw-r--r--	1	root	root	962	Des	2	23:14	COPYING
drwxr-xr-x	2	root	root	4096	Des	2	23:14	dictionaries
drwxr-xr-x	3	root	root	4096	Des	2	23:14	docs
drwxr-xr-x	2	root	root	4096	Des	2	23:14	extra
drwxr-xr-x	3	root	root	4096	Des	2	23:14	lib
-rw-r--r--	1	root	root	24487	Des	2	23:14	LICENSE
drwxr-xr-x	2	root	root	4096	Des	2	23:16	log
drwxr-xr-x	2	root	root	4096	Des	4	23:18	metadata
drwxr-xr-x	2	root	root	4096	Des	2	23:14	metadata-templates
drwxr-xr-x	52	root	root	4096	Des	2	23:17	modules
-rw-r--r--	1	root	root	491	Des	2	23:14	README.md
drwxr-xr-x	2	root	root	4096	Des	2	23:14	schemas
drwxr-xr-x	3	root	root	4096	Des	2	23:14	templates
drwxr-xr-x	4	root	root	4096	Des	2	23:14	tests
drwxr-xr-x	3	root	root	4096	Des	2	23:14	tools
drwxr-xr-x	8	root	root	4096	Des	2	23:16	vendor

Gambar 3.18 Tool untuk menjalankan SAML

Pada gambar di atas terdapat *direktori cert*. Directory tersebut merupakan sertifikat yang berisi aturan enkripsi keamanan dari data log-in pada idp. Di sini enkripsinya menggunakan SSL. Kemudian terdapat juga file metadata yang berisi aturan-aturan yang dibuat sesuai desain SAML pada Bab 3. Aturan tersebut mengatur secara detail bagaimana setiap langkah sp dan idp berkomunikasi dalam melakukan pengolahan data user.

Di bawah ini adalah isi dari direktori metadata :

```

root@server:/var/simplesamlphp/metadata# ls -l
total 48
-rw-r--r-- 1 root root 296 Des 2 23:16 adfs-idp-hosted.php
-rw-r--r-- 1 root root 268 Des 2 23:16 adfs-sp-remote.php
-rw-r--r-- 1 root root 2374 Des 4 23:18 saml20-idp-hosted.php
-rw-r--r-- 1 root root 2370 Des 4 15:43 saml20-idp-hosted.php~
-rw-r--r-- 1 root root 950 Des 2 23:16 saml20-idp-remote.php
-rw-r--r-- 1 root root 1187 Des 2 23:16 saml20-sp-remote.php
-rw-r--r-- 1 root root 620 Des 2 23:16 shib13-idp-hosted.php
-rw-r--r-- 1 root root 414 Des 2 23:16 shib13-idp-remote.php
-rw-r--r-- 1 root root 232 Des 2 23:16 shib13-sp-hosted.php
-rw-r--r-- 1 root root 418 Des 2 23:16 shib13-sp-remote.php
-rw-r--r-- 1 root root 225 Des 2 23:16 wsfed-idp-remote.php
-rw-r--r-- 1 root root 167 Des 2 23:16 wsfed-sp-hosted.php

```

Gambar 3.19 Metadata

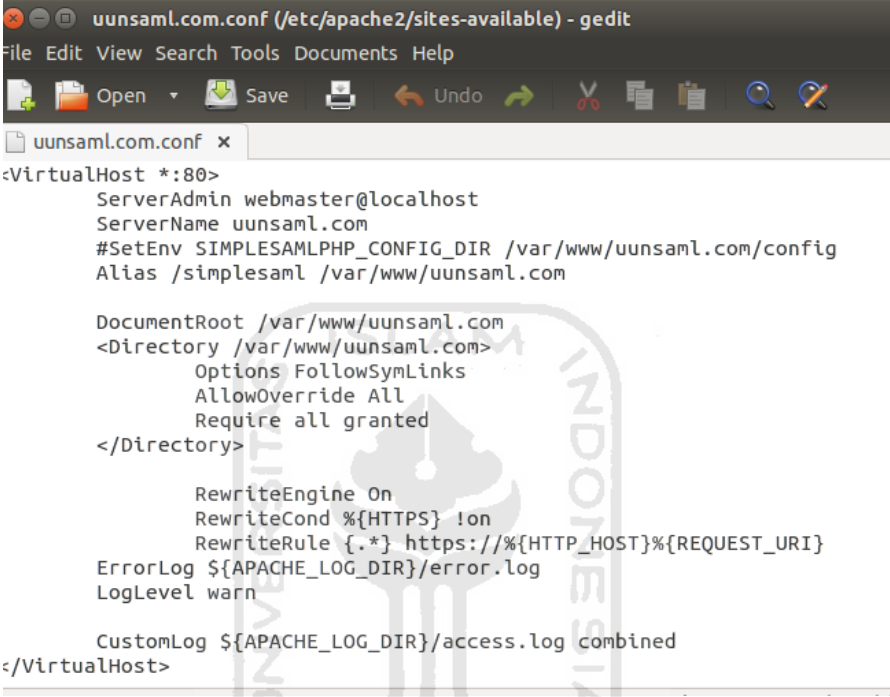
Terdapat banyak aplikasi yang ada di dalam direktori metadata. Dalam hal ini digunakan SAML, baik itu SAML idp maupun SAML sp. Untuk dapat menjalankan *protocol* SAML kita perlu membuat sebuah *virtual host*. Virtual host tersebut ada di dalam `/etc/apache2/sites-available` seperti yang ada pada gambar 3.19.

```

root@server:/var/www# ls -l
total 4
drwxr-xr-x 2 root root 4096 Nov  8 16:20 html
lrwxrwxrwx 1 root root   20 Des  4 03:25 uunsaml.com -> ../simplesamlphp/www

```

Gambar 3.20 Setting domain name



```

uunsaml.com.conf (/etc/apache2/sites-available) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Redo
uunsaml.com.conf x
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName uunsaml.com
    #SetEnv SIMPLESAMPLPHP_CONFIG_DIR /var/www/uunsaml.com/config
    Alias /simplesaml /var/www/uunsaml.com

    DocumentRoot /var/www/uunsaml.com
    <Directory /var/www/uunsaml.com>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    RewriteEngine On
    RewriteCond %{HTTPS} !on
    RewriteRule {.*} https://%{HTTP_HOST}%{REQUEST_URI}
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn

    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

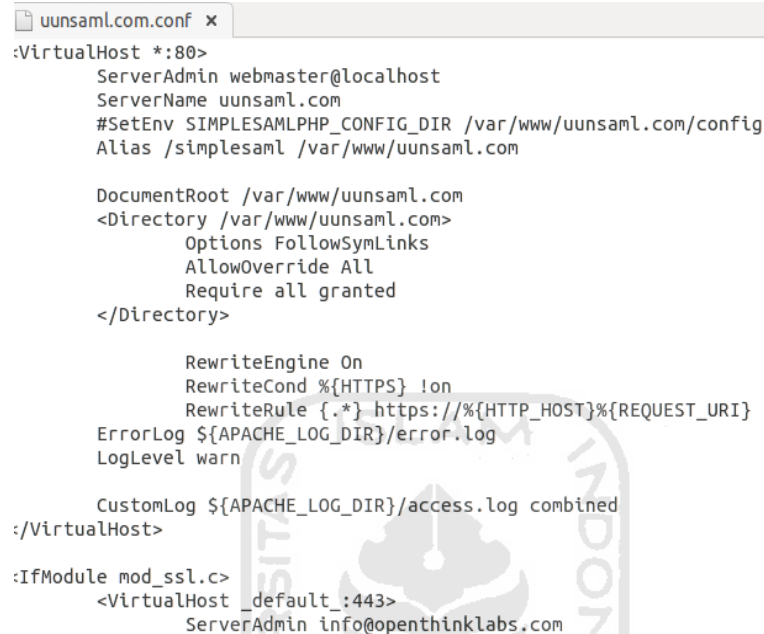
```

Gambar 3.20 Virtual host

Sebelumnya telah dibuat domain uunSAML.com dalam direktori /var/www/ seperti yang terlihat pada gambar 3.19. Kemudian dibuatlah virtual host seperti gambar 3.20.

c) Konfigurasi antara SAML dengan LDAP

Untuk menghubungkan SAML dengan ldap maka dilakukan setting pada direktori `/var/simpleSAMLphp/config/authsources.php` yang kemudian direpresentasikan oleh gambar di bawah ini :



```
uunsaml.com.conf x
:VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName uunsaml.com
    #SetEnv SIMPLESAMLPHP_CONFIG_DIR /var/www/uunsaml.com/config
    Alias /simplesaml /var/www/uunsaml.com

    DocumentRoot /var/www/uunsaml.com
    <Directory /var/www/uunsaml.com>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    RewriteEngine On
    RewriteCond %{HTTPS} !on
    RewriteRule {.*} https://%{HTTP_HOST}%{REQUEST_URI}
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn

    CustomLog ${APACHE_LOG_DIR}/access.log combined
:VirtualHost>

:IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin info@openthinklabs.com
```

Gambar 3.22 Setting idp



```
server.uun.com.conf x
/VirtualHost *>
    ServerName server.uun.com
    DocumentRoot /var/www/server.uun.com

    SetEnv SIMPLESAMLPHP_CONFIG_DIR /var/simpleSAMLphp/config

    Alias /simplesaml /var/simpleSAMLphp/www

    <Directory /var/simpleSAMLphp/www>
        <IfModule !mod_authz_core.c>
            # For Apache 2.2:
            Order allow,deny
            Allow from all
        </IfModule>
        <IfModule mod_authz_core.c>
            # For Apache 2.4:
            Require all granted
        </IfModule>
    </Directory>
/VirtualHost>
```

Gambar 3.23 Setting sp

```

adfsources.php x
*/

'example-userpass' => array(
    'exampleauth:UserPass',

    // Give the user an option to save their username for future login attempts
    // And when enabled, what should the default be, to save the username or not
    //'remember.username.enabled' => FALSE,
    //'remember.username.checked' => FALSE,

    'student:studentpass' => array(
        'uid' => array('test'),
        'eduPersonAffiliation' => array('member', 'student'),
    ),
    'employee:employeepass' => array(
        'uid' => array('employee'),
        'eduPersonAffiliation' => array('member', 'employee'),
    ),
),
),
),

```

Gambar 3.24 Konfigurasi SAML dan ldap

Sebelum melakukan konfigurasi untuk menghubungkan ldap dengan SAML, maka sebelumnya dibuat *virtual host* terlebih dahulu dari idp dan sp. Seperti yang dijelaskan pada gambar 3.20 bahwa domain dari idp adalah uunSAML.com. Kemudian seperti yang dijelaskan pada gambar 3.21, domain dari SAML sp adalah server.uun.com . Jika *virtual host* sudah dipastikan aktif, maka langkah selanjutnya adalah mengkonfigurasikannya sesuai dengan gambar 3.22. *Username* nya adalah *student* dan *passwordnya* adalah *studentpass*. Itu adalah id secara default yang belum diubah sesuai kebutuhan simulasi sistem pada Tugas Akhir ini.

BAB IV

HASIL DAN PEMBAHASAN

Pada pembahasan sebelumnya sudah dijelaskan mengenai metodologi-metodologi yang digunakan untuk menyelesaikan Tugas Akhir ini, mulai dari analisis kebutuhan, analisis perangkat keras, analisis perangkat lunak, dan beberapa metode lainnya. Setelah selesai dengan pembahasan mengenai metodologi yang digunakan, maka akan dilanjutkan kepada hasil dan pembahasan dari praktik Tugas Akhir yang telah dilakukan.

4.1.1 Hasil

Pada tahap ini akan menjelaskan mengenai implementasi dari pengujian server SAML dan LDAP yang dijalankan pada mesin linux ubuntu. Pengujian ini bertujuan untuk mengetahui dan menguji coba sistem. Apakah sistem dapat berjalan dengan semestinya, apakah *client* dapat mengakses akun yang sudah didaftarkan dalam database LDAP atau tidak. Pengujian akan dilakukan dengan menjalankan SAML dan LDAP

4.1.2 Implementasi

Dalam SAML dibangun dengan dua komponen utama yaitu *service provider* dan *identity provider* sebagai komunikator otentikasi dengan LDAP server sebagai *manajemen user* dengan konsep pohon hierarkinya.

4.1.3 Batasan Implementasi

Dalam implementasinya, SAML ini memiliki beberapa batasan-batasan. Batasan-batasan tersebut di antaranya :

- a) Pada saat log-in, komputer *client* akan menggunakan akun yang sudah didaftarkan pada idp yang menggunakan *datastore* LDAP.
- b) Komputer yang digunakan menggunakan satu komputer virtual dengan LDAP dan SAML di dalamnya.
- c) Ada dua jenis pengguna yang akan digunakan dalam sistem yaitu *group* dan *people*.
- d) Sistem ini hanya terbatas pada pengimplementasian hingga SAML dan LDAP bisa dapat dioperasikan.

Untuk melakukan implementasi SAML berbasis LDAP ini, dibutuhkan beberapa perangkat pendukung dari perangkat lunak maupun perangkat keras. Adapun perangkat pendukung tersebut diantaranya :

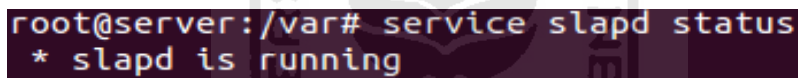
- a) Sistem operasi Linux Ubuntu.
- b) Aplikasi web Browser Google Chrome.
- c) Vmware Workstation Version 14.04.

4.2 Hasil Pengujian Sistem

Pada tahap implementasi akan dijelaskan hasil realisasi dari perancangan yang telah dibuat pada bab sebelumnya. Kemudian diimplementasikan ke dalam sistem. Hasil pengujian sistem dengan judul implementasi SAML 2.0 pada autentikasi *single sign-on* berbasis LDAP bertujuan memastikan bahwa sistem yang dibuat sesuai dengan yang diharapkan. Sistem ini berjalan di sebuah *localhost* dengan ip 127.0.0.1.

Adapun penjelasan dari hasil implementasi sistem adalah sebagai berikut :

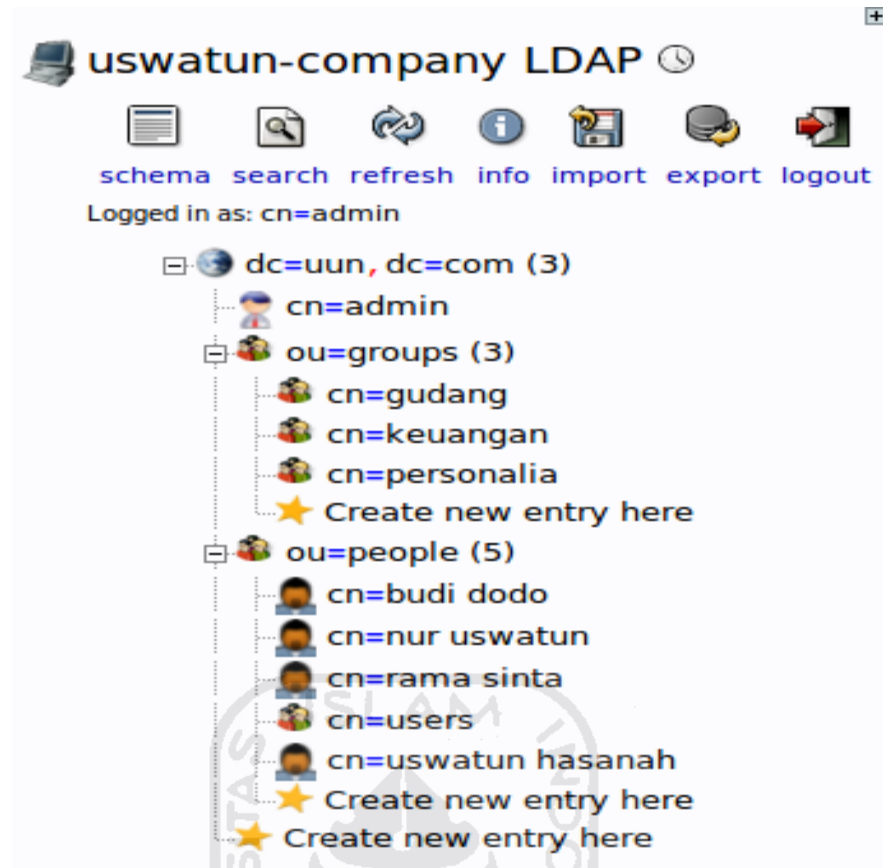
- a) Service LDAP dapat dipastikan berjalan dengan baik



```
root@server:/var# service slapd status
* slapd is running
```

Gambar 4.1 LDAP sudah dalam keadaan berjalan

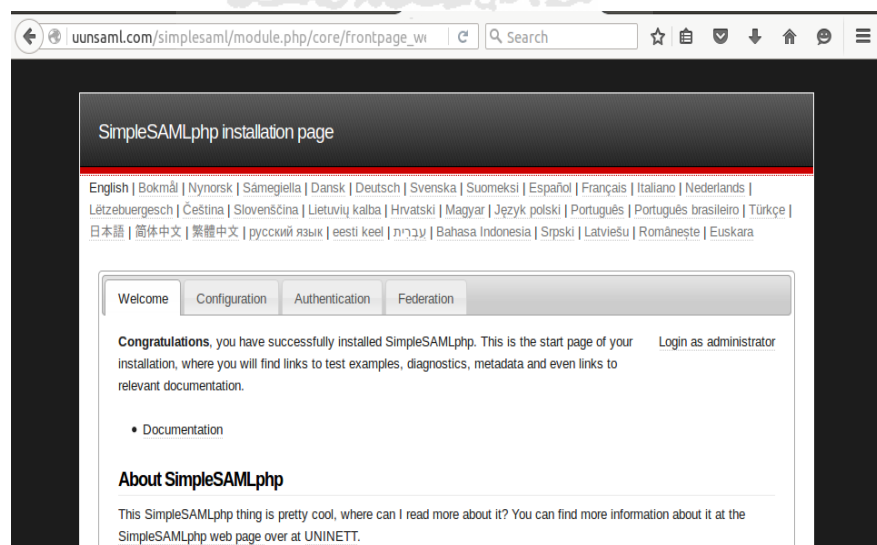
Setelah ldap sudah dipastikan berjalan maka hal pertama yang dilakukan adalah melakukan *reconfigure* paket slapd dan mengubah pengaturannya sesuai dengan kebutuhan sistem. Untuk DNS *domain name* diberi nama *uswatun-company* kemudian menggunakan database *hdp* karena merupakan database *backend* dengan *layout* hierarki. LDAP yang digunakan adalah LDAPv3. Ketika semua sudah bisa dipastikan bisa berjalan kemudian kita mengakses *localhost/phpldapadmin*. User didaftarkan sesuai dengan *root* hierarki yang dirancang pada Bab 3 ke dalam LDAP. Terdapat dua organisasi unit pada ldap yaitu *group* dan *people*. Tiga *group* tersebut yaitu gudang keuangan dan personalia. Masing masing *group* memiliki *user* dan *admin*.



Gambar 4.2 User pada LDAP

Gambar 4.2 adalah tampilan yang menunjukkan daftar *user* yang dibuat pada uswatun-company berdasarkan *root* pada Bab 3.

b) SAML dapat dipastikan dengan baik



Gambar 4.3 SAML sudah dalam keadaan berjalan

SAML sudah bisa diakses dengan domain uunsaml.com . Terdapat 4 halaman yaitu halaman awal, *configuration*, *authentication*, *federation*. Pertama, admin harus melakukan log-in dulu untuk mendapatkan akses dalam pengaturan yang ada di SAML. Kemudian setelah itu untuk menghubungkan antara SAML dan ldap, admin masuk ke kolom *authentication*. Gambar di bawah ini adalah tampilan dari kolom *authentication*.



Gambar 4.4 Test SAML authentication

English | Bokmål | Nynorsk | Sámeigiella | Dansk | Deutsch | Svenska | Suomi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עברית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara

Test authentication sources

- [admin](#)
- [default-sp](#)
- [example-userpass](#)
- [example-ldap](#)

Gambar 4.5 Test SAML authentication

Gambar 4.6 SAML login

Dari gambar 4.6 dan 4.7 bisa kita lihat tampilan *authentication* SAML. Pilih axample-ldap untuk mengetahui apakah ldap dan SAML sudah dapat berjalan atau belum. Kemudian akan muncul tampilan log-in seperti pada gambar 4.8.

Pada otentikasi nantinya akan ada tahap torisasi dengan tandatangan digital pada permintaan otentikasi kepada user membatasi kemampuan penyerang mengubah isi dari pesan yang dimodifikasi terbuka untuk serangan XSS. Namun, kontrol keamanan ini saja tidak cukup untuk memblokir semua serangan XSS. Memang penyerang mungkin menemukan URL yang parameter yang rentan terhadap injeksi kode berbahaya, permintaan kata URL dan memberi makan Otentikasi Permintaan ini menandatangani kontrak dengan korbannya.

Selain itu, tanda tangan bahkan mungkin dihapus dari Permintaan Authentication: sebagai contoh, pengalihan HTTP mengikat mandat tanda tangan yang akan disimpan sebagai parameter URL dikodekan dan ditambahkan ke URL yang berisi permintaan. Dengan menghapus parameter mengatakan, penyerang masih dapat memodifikasi isi RelayState dan dengan demikian dapat memicu serangan XSS yang sukses. Vektor serangan tersebut diblokir hanya jika IdP menolak untuk menerima permintaan yang tidak ditandatangani.

Adapun dalam menjalankan tugasnya, idp dan sp diatur pada metadata. Admin menentukan mana aplikasi yang dapat diakses oleh *user* dan mana yang tidak. Admin melakukan pengaturan *user* berdasarkan *protocol* yang ada di metadata idp.

```
$metadata['server.uun.com'] = array (
  'SingleLogoutService' =>
    array (
      0 =>
        array (
          'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
          'Location' => 'http://uunsaml.com/simplesaml/module.php/saml/sp/saml2-logout.php/default-sp',
        ),
      ),
  'AssertionConsumerService' =>
    array (
      0 =>
        array (
          'index' => 0,
          'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST',
          'Location' => 'http://uunsaml.com/simplesaml/module.php/saml/sp/saml2-acis.php/default-sp',
        ),
      1 =>
        array (
          'index' => 1,
          'Binding' => 'urn:oasis:names:tc:SAML:1.0:profiles:browser-post',
          'Location' => 'http://uunsaml.com/simplesaml/module.php/saml/sp/saml1-acis.php/default-sp',
        ),
      2 =>
        array (
          'index' => 2,
          'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact',
          'Location' => 'http://uunsaml.com/simplesaml/module.php/saml/sp/saml2-acis.php/default-sp',
        ),
      3 =>
        array (
          'index' => 3,
          'Binding' => 'urn:oasis:names:tc:SAML:1.0:profiles:artifact-01',
          'Location' => 'http://uunsaml.com/simplesaml/module.php/saml/sp/saml1-acis.php/default-sp/artifact',
        ),
      ),
  'certData' => 'MIID1TCCAr2gAwIBAgIJAN7pocHa3BSaMA0GCSqGSIb3DQEBCwUAMIGAMQswCQYDVQQGEwJBVTESMBAGA1UECwJQXVzdHJhbGhlbMQ6w',
);
```

Gambar 4.7 Metadata SAML

```

<?xml version="1.0"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" e
<md:SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protoco
  <md:KeyDescriptor use="signing">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:X509Data>
        <ds:X509Certificate>MIID1TCCAr2gAwIBAgIJAN7pocHa3BSaMA0GCSqGSIb3DQEBQwUAMIGAMQswCQYDVQQGEwJBVTESMBAGA1UECAwJQXVz
      </ds:X509Data>
    </ds:KeyInfo>
  </md:KeyDescriptor>
  <md:KeyDescriptor use="encryption">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:X509Data>
        <ds:X509Certificate>MIID1TCCAr2gAwIBAgIJAN7pocHa3BSaMA0GCSqGSIb3DQEBQwUAMIGAMQswCQYDVQQGEwJBVTESMBAGA1UECAwJQXVz
      </ds:X509Data>
    </ds:KeyInfo>
  </md:KeyDescriptor>
  <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="http://uunsaml.com/simp
  <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="http://uunsaml.com/sin
  <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post" Location="http://uunsaml.com/
  <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact" Location="http://uunsaml.com/
  <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01" Location="http://uunsaml.com/s
  </md:SPSSODescriptor>
</md:EntityDescriptor>

```

Gambar 4.8 Metadata SAML



Metadata yang dijelaskan pada gambar 4.4 dan 4.5 adalah protocol data dari SAML *service provider*. Binding terletak di dalam HTTP. Gambar 4.4 menjelaskan *user* yang ingin mengakses sebuah portal *direct* ke sebuah *web browser*. Data milik *user* disimpan dalam bentuk *array*. Kemudian nantinya akan dikenakan aksi *POST* jika *user* sudah teridentifikasi pada *datastore*. Selanjutnya pada gambar 4.5 dijelaskan bahwa standar sertifikat yang digunakan adalah X509 dengan menggunakan aturan standar XML.

```

$dst = $idpMetadata->getEndpointPrioritizedByBinding('SingleLogoutService', array(
    SAML2_Const::BINDING_HTTP_REDIRECT,
    SAML2_Const::BINDING_HTTP_POST)
);

if (!$binding instanceof SAML2_SOAP) {
    $binding = SAML2_Binding::getBinding($dst['Binding']);
    if (isset($dst['ResponseLocation'])) {
        $dst = $dst['ResponseLocation'];
    } else {
        $dst = $dst['Location'];
    }
    $binding->setDestination($dst);
    $lr->setDestination($dst);
    $binding->send($lr);
} else {
    throw new SimpleSAML_Error_BadRequest('Unknown message received on logout endpoint: ' .
    get_class($message));
}

```

Gambar 4.9 Binding data

Gambar 4.6 menjelaskan metode *binding* data milik *user*. Dengan SAML_SOAP informasi yang dibawa oleh *idp* menuju ke *sp* dibinding kemudian dibawa ke halaman *browser*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil proses studi dan implementasi pada pengujian Tugas Akhir, maka dapat disimpulkan :

1. Kesimpulan yang didapat dari implementasi SAML 2.0 pada sistem *single sign on* berhasil dilakukan.
2. Dengan LDAP sebagai otentikator, sistem juga mengalami otentikasi SSO yang merupakan hasil dari redirect SAML sp sebagai *Service Provider* dan SAML idp sebagai *Identity Provider*. Untuk sisi keamanan sistem cukup aman karena menggunakan https pada SAML sp SAML idp dan cert sehingga memungkinkan untuk diterapkan di sistem pada perusahaan yang bersifat heterogen.

5.2 Saran

Dalam melakukan pengujian ini, ada beberapa saran yang diajukan mengenai topik ini :

1. Untuk menghubungkan SAML dengan LDAP buatlah sebuah metadata yang berisi daftar *user* beserta *privilagenya* masing-masing.
2. Penentuan parameter kualitas jaringan yang lebih kompleks.

Demikian beberapa kendala yang penulis temukan dalam melakukan pengujian SAML ini. Semoga kendala-kendala tersebut dapat dijadikan sebagai bahan sekaligus saran untuk penelitian selanjutnya. Jika terdapat kekeliruan dalam penyampaian pada penulisan mohon dimaafkan. Semoga dengan diperolehnya hasil implementasi SAML berbasis LDAP dapat memberikan sebuah kemanfaatan peningkatan efisiensi otentikasi.

DAFTAR PUSTAKA

- Aaslund, K., Larsen, S. (2007). OTS-Wiki: A Web Community for Fostering Evaluation and Selection of Off-The-Shelf Software Components. Department of Computer and Information Science. Norwegian University of Science and Technology (NTNU).
- Annonymous (2006). Single sign on: Finding the Best Fit for Your Business. White Paper Security and Identity Solution. Novel,Inc. Waltham, MA 02451 USA.
- Rudy, Riechie, Gunadi, O., 2007. *Integrasi Aplikasi Menggunakan Single Sign-On Berbasis Lightweight Directory Access Protocol Dalam Portal Binus. Indonesia*
- Sukmana, R. C. (2006). Pengenalan LDAP, 1–9
- Dwi Rita Sari, Yesi Novaria Kunang, Ari Muzakir,(2015), “*Sistem Keamanan SSO Berbasis SAML pada Jalur Komunikasi dengan Menggunakan XML Encryption*”
- OASIS Standart, *Binding for the OASIS Security Assertion Markup Language (SAML) V2.0.* March 2005. <https://docs.oasis-open.org/security/SAML/v2.0/SAML-biding-2.0-os.pdf>.

LAMPIRAN

