Implementasi SSO (*SINGLE SIGN ON*) Menggunakan Autentikasi NCSA untuk *Website* di *Web Server*

[1]Hendi Pangestu **[2] Periyadi, ST.** **[3] Prajna Deshanta, ST.**

[1,2,3] Teknik Komputer Politeknik Telkom
Jl. Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia

[1]hp.coqes@gmail.com [2]periyadi2000@yahoo.com [3]desh_ibnu@yahoo.co.id

ABSTRAKSI

Kebutuhan dan keinginan pengguna akan Internet makin meluas, jadi suatu keamanan apalagi dalam hak aksesnya sangat diperlukan. Untuk itu dibutuhkan suatu keamanan jaringan yang sangat penting, untuk hal ini khusus *Linux* menggunakan program *squid* melalui sebuah *proxy server.* Adapun cara kerja dari *proxy server* dalam s*quid*, pertama-tama akan memeriksa *request* yang datang. Jika *squid* di-*set* dengan autentikasi tertentu, *squid* akan memeriksa autentikasi *user* terlebih dahulu. Autentikasi ini termasuk *subnet area, user account*, jenis *file* yang di-*request*, alamat situs tujuan, dan properti-properti yang telah diatur pada *file* konfigurasi *squid*. Jika lolos dan telah sesuai dengan konfigurasi, *request* tersebut kembali diperiksa apakah objek yang diminta telah berada di *cache*. Jika sudah ada maka *proxy server* tidak perlu melanjutkan *request* ke Internet tetapi langsung me-*replay request* dengan objek yang diminta.

Untuk itu disini akan menjelaskan tentang autentikasi yang menjadikan suatu SSO (*single sign on*) antara autentikasi NCSA yaitu sebagai penggunaan autentikasi dan proteksi untuk *website* di *web server.* Proses ini akan di bentuk dengan satu akun dan bisa login ke beberapa aplikasi dengan satu akun yang di miliki pengguna itu sendiri. Dan juga sebagai keamanan yang berbentuk enkripsi agar bisa terlindung dari penyadapan. Adapun cara kerjanya ketika *user* ingin melakukan *request* pada *browser* maka akan muncul kotak login *username* dan *password,* jika *user* cocok sesuai dengan sistemnya maka akan bisa untuk melakukan koneksi internet melalui *proxy* yang ditentukan dan untuk layanan selanjutnya dengan cara kerja yang sama tapi dalam membuka *website* yang telah diberikan proteksi maka akan muncul juga suatu proteksi autentikasi, untuk login masukkan *username* dan *password* yang sama dengan autentikasi NCSA.

Kata kunci : *Network Security*, *Proxy Server,* Autentikasi NCSA, *Web Server.*

1. PENDAHULUAN

1.1 Latar Belakang

Dalam makalah proyek akhir ini penulis akan mengimplementasikan penggunaan *squid server* sesuai suatu kebutuhan perusahaan yang membutuhkan. Dan akan menerapkan sebuah *single sign on* yang mana akan memberikan satu akun dan satu kata sandi dalam autentikasi yang berbeda pada autentikasi NCSA dan proteksi *website* yang memiliki batas hak akses.

1.2 Perumusan Masalah

Perumusan masalah pada proyek akhir ini adalah :

1. Konfigurasi *single sign on* pada akun yang ada agar jaringan antara autentikasi di NCSA dan autentikasi pada *website* dapat berkomunikasi menjadikan satu akun.
2. Pembatasan dan keamanan dengan autentikasi NCSA sebagai hak akses yang diberikan pada *user* di *Squid Server*.

1.3 Tujuan

Tujuan dari proyek akhir adalah :

1. Implementasi sistem *single sign on* pada akun yang akan dibatasi pada website melalui autentikasi NCSA dan *website*-nya menjadi satu akun.
2. Mengimplementasikan pembatasan dan keamanan dengan autentukasi NCSA sebagai hak akses yang di berikan pada *user* di *Squid Server*.

1.4 Batasan Masalah

Untuk lebih memudahkan dalam pembahasan permasalahan dan untuk menghindari penyimpangan pembahasan dan pokok bahasan maka permasalahan dibatasi pada :

1. *Server* yang digunakan pada sistem operasi yang digunakan pada *Squid Server* adalah Ubuntu 9.10 . Autentikasi yang digunakan adalah NCSA program dan Web Server.
2. Pada autentikasi akan diberikan jaringan komunikasi berupa *single sign on* antara NCSA dan *website* yang di batasi.
3. Dalam hal ini proyek akhir menerapkan autentikasi yang di berikan pada pengguna internet sebagai hak akses dengan NCSA.

1.5 Jadwal Pelaksanaan

Tabel 1. 1 Jadwal Pelaksanaan

Kegiatan (Tahun 2010)	**Mei**	**Juni**	**Juli**	**Agustus**
Studi Literatur				
Pembangunan Model				
Implementasi				
Pembuatan Laporan				

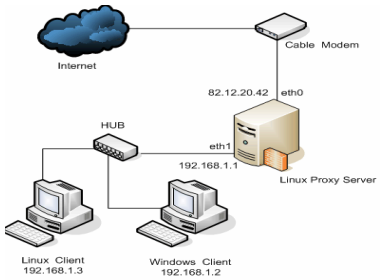
2. TINJAUAN PUSTAKA

2.1 Pengenalan *Single Sign ON*

Teknologi ***Single-sign-on*** (sering disingkat menjadi SSO) adalah teknologi yang mengizinkan pengguna jaringan agar dapat mengakses sumber daya dalam jaringan hanya dengan menggunakan satu akun pengguna saja. Teknologi ini sangat diminati, khususnya dalam jaringan yang sangat besar dan bersifat heterogen (di saat sistem operasi serta aplikasi yang digunakan oleh komputer adalah berasal dari banyak *vendor*, dan pengguna dimintai untuk mengisi informasi dirinya ke dalam setiap *platform* yang berbeda tersebut yang hendak diakses oleh pengguna). Dengan menggunakan SSO, seorang pengguna hanya cukup melakukan proses autentikasi sekali saja untuk mendapatkan izin akses terhadap semua layanan yang terdapat di dalam jaringan.

2.2 Pengenalan *Proxy Server*

Proxy server berbasis linux adalah *proxy server* yang di operasikan pada sistem operasi linux dengan menggunakan *squid* sebagai program *proxy server*nya. Dengan menggunakan fasilitas yang di sajikan *squid*, *proxy server* dapat di konfigurasi sehingga *server* tersebut dapat memberikan respon yang cepat atas sebuah *request* ke internet. Dengan *proxy server* berbasiskan linux dapat dibangun sebuah *proxy* yang handal, *realibel* dan *fleksibel* perkembangan teknologi informasi sekarang ini.



Gambar 2.1. Arsitektur *client-server* pada penerapan *proxy server*

Squid server merupakan sebuah aplikasi *web cache* dan *proxy server* yang berfungsi mempercepat akses internet dan menyaring serta memantau lalu lintas yang melalui jaringan. Dalam implementasinya *squid server* dibangun dengan arsitektur berbasis *client-server* (gambar 2.1). Dimana terdapat sebuah sistem yang berfungsi sebagai *squid server* dan sejumlah *client* yang terhubung dengan *server* tersebut. Dalam penggunaannya, bisaanya seorang *client* yang akan mengakses internet diharuskan untuk memasukkan autentikasi berupa *username* dan *password* yang sebelumnya telah terdaftar pada *squid proxy*.

Selanjutnya informasi ini akan dikirimkan ke *squid server* untuk diautentikasi, jika informasi yang dimasukkan benar, maka sang *client* berhak melakukan koneksi ke internet. Koneksi yang didapat oleh *client* akan selalu diawasi oleh *rule-rule* yang telah dibuat sebelumnya oleh penanggung jawab *server*. Autentikasi yang dilakukan oleh *proxy server* bisaanya tidak terenkripsi,sehingga memungkinkan pihak-pihak yang tidak berwenang mendapatkan informasi mengenai *username* dan *password* milik *client*. Dengan mendapatkan *username* dan *password* milik *client*, pelanggan yang berada dalam jaringan dapat melakukan akses ke internet tanpa perlu takut identitasnya ketahuan. Sistem autentikasi yang kurang sempurna ini juga mengakibatkan aliran data antar *client-server* yang tidak terenkripsi ini rentan terhadap penyadapan. Untuk itu sistem kriptografi berupa enkripsi terhadap perangkat lunak sangat dibutuhkan oleh *squid server*. Saat ini terdapat beberapa metode autentikasi yang digunakan oleh *squid server*, yaitu metode NCSA *authentication* ,metode PAM, LDAP *authentication, dsb.*

2.3 Sistem Autentikasi di *Squid*

Squid mendukung 4 skema autentikasi, yaitu:

1. Basic
2. Digest
3. NTLM
4. Negotiate (mulai dari versi 2.6)

Masing-masing skema autentikasi punya kelebihan dan kekurangan masing-masing.

2.3.1 *Basic Authentication*

Ini adalah skema autentikasi yang didukung oleh semua peramban (*browser*) utama. Dan lebih dari itu, bisa berfungsi dengan baik di semua platform OS. Jadi kalau ingin menggunakan skema autentikasi yang yakin berfungsi dengan baik di semua *browser*, pakailah skema autentikasi basic. Sayangnya skema autentikasi basic ini memiliki satu kelemahan utama, yaitu proses pengiriman data *user* dan *password* dikirim dalam format plain text. Jadi sangat rentan terhadap proses snip atau penyadapan saat proses autentikasi berlangsung.

2.3.2 Program membantu untuk autentikasi

Squid menyediakan beberapa program bantu untuk skema autentikasi basic. Kita bisa memilih mana yang cocok dengan keperluan.

1. LDAP: Autentikasi ke LDAP.
2. NCSA: Menggunakan format penulisan *username* dan *password* format NCSA.
3. MSNT: Autentikasi ke domain Windows NT.
4. PAM: Menggunakan skema autentikasi PAM yang umum digunakan di sistem operasi Unix/Linux.

5. SMB: Menggunakan *server* SMB seperti Windows NT atau Samba.
6. getpwam: Menggunakan cara kuno, berkas *password* di Unix/Linux.
7. SASL: Mengggunakan pustaka SASL.
8. mswin_sspi: Windows native authenticator.
9. YP: Menggunakan database NIS.

2.3.3 Autentikasi dengan NCSA atau Program ncsa_auth

Saat ini perkembangan internet sudah sangat pesat, sehingga sangat mudah untuk melakukan pencurian terhadap *password* milik seseorang yang berada pada jaringan yang sama dengan menggunakan *sniffer tool* bisaa. Karena itulah sangat dibutuhkan sebuah sistem autentikasi untuk menjaga keamanan dan kerahasiaan data yang dikirimkan melalui sebuah *proxy server*. Terdapat berbagai jenis autentikasi yang dapat digunakan pada sistem *squid server*, tetapi yang paling sederhana dari kesemua sistem autentikasi tersebut adalah NCSA *authentication.* NCSA *authentication* merupakan autentikasi berbasis httpd (*web server*) *password* yang memungkinkan seorang *client* melakukan koneksi setelah melakukan autentikasi berupa *username* dan *password. Username* dan *password* ini telah tersimpan di *server* dengan format yang telah ditentukan sebelumnya.

Cara kerja NCSA *authentication* adalah *:*

1. *Client* mengirimkan *username* dan *password* kepada sistem, yang telah terenrkripsi.
2. Sistem akan melakukan decoding ulang dari *password* dan membandingkan dengan berkas *passwd* yang ada pada *server*.
3. Jika *password* dan *username* cocok, maka *client* akan diizinkan untuk melakukan koneksi internet melalui *proxy*.

Program ncsa_auth adalah sebuah fungsi dalam *squid* yang mengijinkan *squid* untuk melakukan autentikasi dengan menggunakan NCSA/Apache httpd-style *password*. NCSA dapat membaca *file password* (lazim disebut *flat file*). Tapi NCSA tidak dapat membaca file /**etc**/**passwd**. NCSA hanya dapat membaca *file password* yang dibuat dengan utilitas **htpasswd** yang di instal bersama Apache (httpd). Jadi, *username/password login* Linux tidak ada sangkut pautnya dengan *username/password Squid*.

2.4 Perbandingan Autentikasi *Proxy Server* pada Penggunaan NCSA dan PAM

PERBEDAAN	NCSA	PAM
Jumlah Proses	Sekali Autentikasi	Selama autentikasi
Waktu Autentikasi	0,01 detik. Hanya dilakukan diawal dan dapat dilakukan berkali-kali oleh pengguna ditempat dan waktu yang berbeda.	0,05 detik. Mengizinkan seorang user melakukan autentikasi pada satu saat dan satu tempat.
Enkripsi	BASE64	BASE64, DES dan lain dari itu.
Proses *Password*	Sistem akan melakukan decoding ulang dari *password* dan membandingkan dengan berkas *passwd* yang ada pada *server*.	Terdapat *password* bayangan.

Tabel 2.1. Autentikasi *Proxy Server* pada Penggunaan NCSA dan PAM

3. ANALISIS KEBUTUHAN DAN PERANCANGAN

3.1 Skenario Jaringan

Dalam makalah tugas akhir ini, penulis akan melakukan pengujian pada simulasi Autentikasi NCSA yang di bentuk SSO (*Single sign On*) dengan website yang di batasi. Adapun skenario yang di berikan pada pengujian autentikasi ini adalah sebagai berikut :

Sebuah jaringan yang telah terbentuk adanya koneksi ke internet yang mana bisa mengakses apapun ke semua URL yang diinginkan seorang *user*. Perlunya suatu keamanan dalam membatasi hak akses terhadap internet dengan bentuk sebuah autentikasi, dalam hal ini pada setiap suatu organisasi ataupun sebuah perusahaan dan dibidang pendidikan memiliki sebuah website tersendiri, bilang saja salah satu *default* yang diambil penulis misal sebuah *website* bidang pendidikan sekolah. Maka di butuhkan pembatasan dalam penggunaan internet untuk mungkin antara guru dan siswa. Untuk itu, di gunakan autentikasi pada akses internet menggunakan Autentikasi NCSA dengan hak akses sebagai berikut :

No	Nama Komputer	Akun	IP address
1	PC 1	User : Chainur, Password : XXX	192.168.1.15
2	PC 2	User : Aswati, Password : XXX	192.168.1.16
3	PC 3	User : Salim, Password : XXX	192.168.1.17

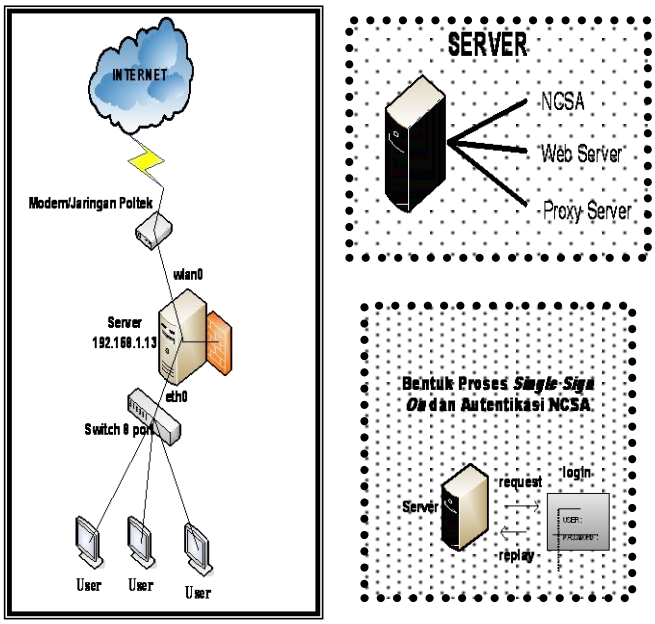
Tabel 3.1 Skenario Jaringan

Dan untuk kemudahan dalam akses pengguna, akan di berikan solusi yaitu SSO atau yang dikenal juga sebagai *Single Sign On.* Jadi antara akun hak akses ke internet dan hak akses pada *website* dari akunnya akan di lebur menjadi satu dan tersimpan dalam sebuah *file* saja, dan juga memudahkan bagi pengguna dalam melakukan hak akses. Adapun langkah-langkah untuk melakukan proses *Single Sign On* dengan menggunakan Autentikasi NCSA tersebut pada *website,* yaitu :

1. *User* menginginkan untuk masuk salah satu alamat URL yang di inginkan.
2. Sebelumnya, ketika membuka aplikasi *browser* muncul *box* untuk meminta memasukkan *username* dan *password,* jika si pengguna di izinkan untuk melakukan hak akses maka akan bisa melanjutkan proses, jika sebaliknya pengguna tidak melakukan akses internet.
3. Jika *user* memiliki hak akses maka bisa melakukan untuk penggunaan internet, jika si pengguna mengunjungi ke *website* yang di miliki sekolah untuk melakukan login, maka si pengguna harus memasuki *username* dan *password* yang sama di saat melakukan autentikasi NCSA.
4. Karena setiap autentikasi yang di berikan tertuju dalam bentuk komunikasi jaringan yang mana memiliki satu akun tapi bisa login di tempat yang berbeda atau disebut juga *single sign on.*

Jika benar login yang dilakukan dapat melakukan proses, jika tidak maka si pengguna tidak berhak atau tidak memiliki akun yang sebenarnya.

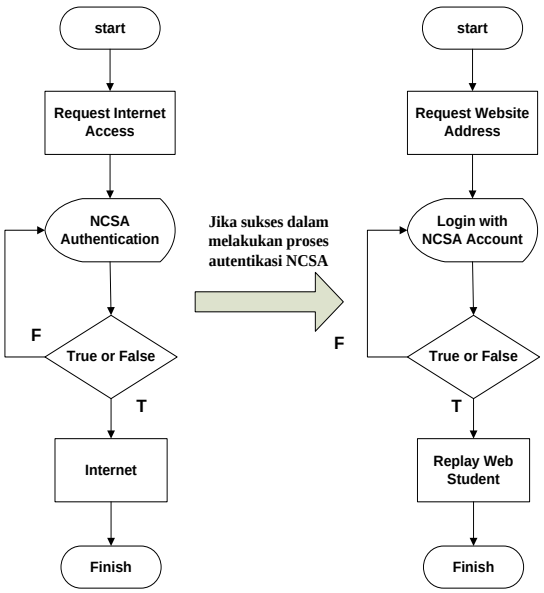
3.1 Arsitektur Sistem



Gambar 3.4 Disain Arsitektur Jaringan

3.2 Alur Proses Berdasarkan *Flowchart*

3.2.1 Kinerja Proses Autentikasi NCSA dan *Single Sign On*



Keterangan proses Autentikasi NCSA dan *Single Sign On* :

1. *User* menginginkan untuk masuk salah satu alamat URL yang di inginkan.
2. Sebelumnya, ketika membuka aplikasi *browser* muncul *box* untuk meminta memasukkan *username* dan *password*, jika si pengguna di izinkan untuk melakukan hak akses maka akan bisa melanjutkan proses, jika sebaliknya pengguna tidak melakukan akses internet.
3. Jika *user* memiliki hak akses maka bisa melakukan untuk penggunaan internet, jika si pengguna mengunjungi ke *website* yang di miliki sekolah untuk melakukan login, maka si pengguna harus memasuki *username* dan *password* yang sama di saat melakukan autentikasi NCSA.
4. Karena setiap autentikasi yang di berikan tertuju dalam bentuk komunikasi jaringan yang mana memiliki satu akun tapi bisa login di tempat yang berbeda atau disebut juga *single sign on.*
5. Jika benar login yang dilakukan dapat melakukan proses, jika tidak maka si pengguna tidak berhak atau tidak memiliki akun yang sebenarnya.

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi dan *Service Server*

➢ ***Domain Name Server***

Pada sebuah *proxy server* memerlukan DNS atau yang lebih dikenal sebagai *domain name server*. Untuk itu untuk membuktikan DNS 8ias dijalankan ketikkan pada *console* di Ubuntu " #/etc/init.d/bind9 restart "

➢ ***Web Server***

Dan dalam proyek akhir ini memerlukan juga sebuah *Web Server* yang menggunakan *apache2* pada Ubuntu, untuk menjalankan suatu *web server* 8ias dilakukan *service "* /etc/init.d/apache2 restart "

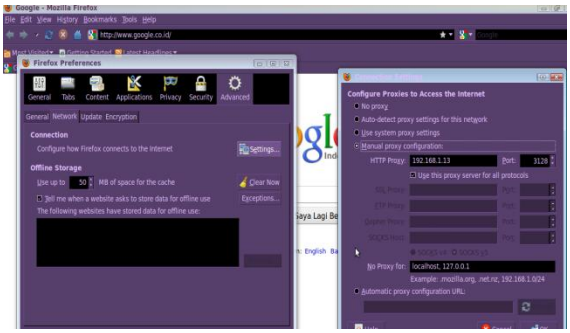
➢ ***Squid Server***

Dalam autentikasi NCSA di perlukan *squid server* juga yang mana sebagai *proxy server*, jadi dalam melakukan untuk *service* nya adalah " /etc/init.d/squid restart "

4.2 Pengujian Sistem

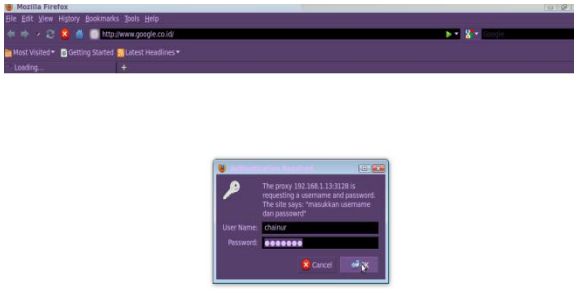
Pada tahap ini penulis akan mencoba melakukan *testing* terhadap konfigurasi serta pembuatan mulai dari penerapan *SSO (Single Sign On)* dan autentikasi NCSA untuk *website* di *web server*. Pada makalah ini penulis melakukan pengujian hanya pada pembatasan autentikasi NCSA dan *single sign on* itu sendiri yang dilakukan pada *web server*. Maka hasilnya akan seperti berikut:

A. Sebelumnya untuk memastikan *proxy server* tersebut pada web browser anda dengan perintah " edit > preferences > advanced > network > settings > " gantilah/pilihlah Manual proxy configuration dengan HTTP proxy : 192.168.1.13 Port: 3128, centrang Use this proxy server for all protocols, lalu "ok"



Gambar 4.4 *setting proxy server*

B. Pengujian terhadap pengaksesan yang mana pengguna yang bernamakan "chainur" ingin melakukan *browsing* melalui koneksi internet, ketika dia membuka salah satu aplikasi atau program salah satu *web browser* sebut saja Mozilla, maka akan muncul suatu box autentikasi, yaitu autentikasi NCSA. Dalam hal ini file yang menyimpan *user* yang 8ias *login* dibentuk dalam sebuah file yang tersimpan "/etc/rahasia/.htpasswd"

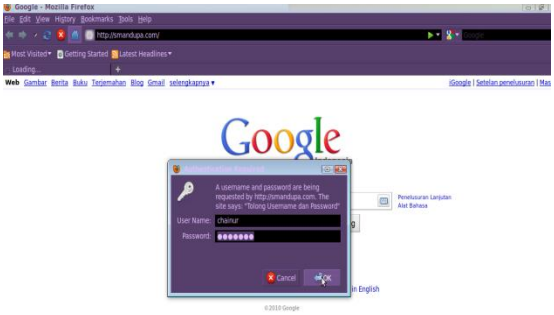


Gambar 4.5 login autentikasi NCSA

C. Jika si pengguna "chainur" tadi bener akan kata sandi yang di masukkan maka dia akan memperoleh untuk dapat koneksi internet, jika sebaliknya maka akan di ulang lagi untuk meminta *username* dan *password* yang lain. Berikut jika si pengguna dapat masuk konek ke internet.

Gambar 4.6 koneksi internet

D. Dan setelah melakukan autentikasi NCSA, maka si pengguna "chainur" menginginkan lagi untuk dapat mengakses sebuah website/web, yang mana web/website itu sendiri telah di beri sebuah proteksi autentikasi juga oleh server. Maka si pengguna harus login kembali dengan *username* dan *password* yang telah di berikan hak akses oleh server pada pengguna. Dan disini di berikan kemudahan pada pengguna oleh server, dengan satu akun saja melalui akun yang dimiliki pada autentikasi NCSA tadi bisa di akses lagi ke proteksi autentikasi yang muncul pada web/website yang d beri proteksi misal "smandupa.com"



Gambar 4.7 login proteksi *website*

E. Setelah itu, pengguna jika memasukkan dengan benar *username* dan *password* nya maka si pengguna akan bisa melakukan akses langsung web/website "smandupa.com" tetapi jika tidak bisa login maka *username* dan *password* yang dimasukkan berarti tidak benar dan seperti autentikasi NCSA, akan meminta atau pengulangan login. Berikut web "smandupa.com"

Gambar 4.8 halaman *website smandupa.com*

Untuk dapat melihat dalam penyelesain pengujian sistem ini sendiri, penulis menuliskan sebuah parameter berupa tabel hasil pengujian, agar bisa memahami apa yang di bentuk/proyek dalam makalah ini.

Tabel Hasil Pengujian

No	Pengujian Fungsi	Metode Pengujian	Hasil Dirapkan	Hasil Sebenarnya	Ket.
1	*Proxy Server*	-Koneksi Internet melalui *proxy server* -Koneksi Internet Langsung	-Terhubung ke Internet -Gagal terhubung ke Internet	-Terhubung ke Internet -Gagal terhubung ke Internet	Valid Valid
2	Autentikasi NCSA	Buka browser dan browsing internet	Permintaan *username* dan *password* yang akan diminta	Adanya permintaan *username* dan *password*	Valid
3	NCSA verikasi	Permintaan sambungan Internet dan kemudian *login*	- *login* yang benar memungkinkan akses internet - *login* tidak valid akan meminta permintaan *login* lain	-Akses internet di aktifkan - Login permintaan diminta, menyangkal akses internet	Valid Valid
4	*Apache Web Service*	Masukkan URL dalam browser smandupa.com	Smandupa.com Halaman Web muncul	Smandupa.com halaman web muncul	Valid
5	*Single Sign On*	Masukkan URL dalam browser smandupa.com	Permintaan *username* dan *password* yang akan diminta	Adanya permintaan *username* dan *password*	Valid
6	*Single Sign On* verifikasi	Masukkan URL dalam browser smandupa.com	-login yang benar memungkinkan bisa masuk ke web -login tidak valid akan meminta permintaan login lain	-smandupa.com bisa di akses -login permintaan diminta, menyangkal akses web	Valid Valid

Table 4.1 Parameter hasil pengujian

5. PENUTUP

5.1. Kesimpulan

Kesimpulan yang di ambil pada makalah ini, adalah :

1. Sistem *single sign on* pada akun yang akan dibatasi pada website melalui autentikasi NCSA dan *website*-nya menjadi satu akun. Maksudnya disini ketika nanti seorang pengguna ingin melakukan koneksi ke internet akan muncul sebuah autentikasi untuk melakukan *login* dengan *username* dan *password*. Jika pengguna memiliki hak akses dengan adanya akun maka pengguna bisa melakukan akses, tetapi jika sebaliknya maka akan ada pengulangan *login* dengan mencoba *login* yang lain.

2. Ketika pengguna telah bisa melakukan koneksi ke internet dan pengguna ingin memasukkan salah satu website misalnya "smandupa.com" maka akan muncul suatu proteksi seperti autentikasi NCSA. Disini sama dengan NCSA untuk masuk dengan *login*, jika salah/tidak benar maka akan lakukan pengulangan login lain. Dalam hal ini, di mudahkan bagi pengguna, untuk hanya memiliki satu akun saja yang mana penulis membuat kemudahan dengan satu akun bisa *login* di dua program/aplikasi yang disebut juga SSO (*Single Sign On*).

1.2 Saran

Saran yang dapat diambil dari implmentasi proyek makalah ini, dapat bisa membandingkan sebagai analisis dengan autentikasi yang lainnya. Dari keamanan agar lebih ditingkatkan dengan pemakaian yang bias dimanfaatkan *squid* juga *proxy server*.