Network Files System (Study Kasus Active Repository Opensource Undip)

Ajie Prasetyo[1)], Adian Fatchur Rochim [2)], Kodrat Iman Satoto[2)]
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

ABSTRACT

The development of komputer technology and network led many organizations that require large data storage. Diponegoro University is an organization whose storage requirements are considerable. Number of services available as a storage repository owned by some faculties, such as industrial engineering, Electrical, Komputer or other systems that have a lot of storage space spread. This causes a lack of coordination arrangements in terms of its data. The price of storage devices and servers owned by geographic location where the different is also the base making the centralized storage, but encounter obstacles removal device server that has been dimasing - each faculty.

NFS (Network File System) is a system file and resource sharing network protocol that enables a centralized data repository. This thesis research methodologies including literature study, system design, and testing of the system. In the literature study used methods of research library of reference books related. The design of this thesis uses a system that has been owned by the UNIX system that is NFS (Network File System). Last is testing this system on their performance on the distribution of server-side storage area used by the server that is reliable and affordable on an active repository opensource Undip.

The test results have a centralized storage area. This centralized storage facility has the performance of such a data center that collects all the data coming from multiple NFS servers. This centralized storage capacity of large data derived from multiple NFS servers into a single entity such that their own local directory. This system helps the problems that arise due to the expensive storage devices and a geographic location different server devices.

Keyword : *data storage, network file system,data center*

I. PENDAHULUAN

Latar Belakang

Perkembangan teknologi komputer yang semakin pesat mengakibatkan badan usaha maupun lembaga akademik mengiplementasikan teknologi ini untuk banyak keperluan-keperluannya. Sebanding dengan bertambahnya fungsi teknologi komputer, bertambah pula keperluan akan luas jaringan komputer yang diperlukan oleh badan tersebut, agar setiap anggota dari badan tersebut dapat menggunakan layanan-layanan teknologi yang disediakan.

Permasalahan yang muncul adalah ketersediaan suatu perangkat pendukung seperti tempat penyimpanan data yang menjadi suatu kebutuhan mutlak pada saat penerapan fungsi teknologi Internet. Aplikasi ataupun sumber data yang berkembang mendorong suatu lembaga ataupun perorangan memiliki tempat penyimpanan yang cukup besar hal ini dikarenakan adanya parameter ataupun penambahan data tiap harinya. Pada sisi server terdapat masalah penyediaan tempat penyimpanan yang cukup besar. Pemenuhan kebuthan tempat penyimpanan ini juga melihat beberapa faktor seperti tingginya harga, keamanan ataupun letak geografis dari server tersebut.

Semua Organisasi badan usaha ataupun lembaga akademis seperti UNDIP juga menghadapi permasalahan yang hampir sama. Masalah utamanya adalah keterbatasan kapasitas yang diberikan oleh server. Selain itu mahalnya tempat penyimpanan yang digunakan khusus untuk server juga menjadi suatu masalah yang cukup besar.

Pemecahan dari permasalahan tersebut maka dibangun sebuah sistem yang disebut *Network File system* (NFS) dimana tempat penyimpanan data pada server tidak hanya berasal dari perangkat lokal tersebut saja. Sistem NFS bekerja disemua platform dari sistem operasi. Hal tersebut mengakibatkan perangkat keras tempat penyimpanan yang digunakan tidak hanya jenis SAS (tempat penyimpanan khusus untuk server) yang harganya sangat mahal, akan tetapi dapat juga menggunakan tipe tempat penyimpanan SATA atau ATA yang mempunyai harga yang lebih murah. Dalam hal kecepatan SAS dan SATA/ATA mempunyai perbedaan yang cukup jauh, SAS mempunyai kecepatan yang lebih besar dibandingkan SATA/ATA. Hal ini dapat dipecahkan dengan kinerja sistem NFS yang menformat semua tempat penyimpanan yang berada didalam sistem tersebut mempunyai peforma yang sama karena direktori yang dipakai akan sama seperti direktori lokal mesin tersebut. Hal ini sangat menguntungkan untuk penggunanan suatu server database yang mengiginkan tempat penyimpanan yang besar dan cepat dengan biaya lebih murah.

Batasan Masalah

Agar pembahasan atau analisis tidak melebar dan terarah, maka permasalahan dibatasi pada :

a. Menggunakan *Linux* sebagai sistem operasi
b. Server NFS yang menggunakan perangkat lunak *Open Source* nfsd
c. NFS pada sisi server dan klien
d. Tidak membahas sisi pemrograman pada MRTG, Sedot , Phpsysinfo
e. Impementasi untuk web open source di Universitas Diponegoro

1) Mahasiswa Teknik Elektro UNDIP

2) Dosen Teknik Elektro UNDIP

II. LANDASAN TEORI

Network File dan Resource Sharing Protocols

Jaringan komputer dibuat untuk satu tujuan yaitu mengizinkan sesama pengguna untuk saling bertukar informasi. Kebanyakan informasi yang berada pada komputer berbentuk berkas yang disimpan disebuah perangkat tempat penyimpanan yang biasa dikenal *hard disks*. Dengan demikian satu tujuan jaringan komputer adalah mengizinkan penggunanya berbagi informasi. *File Transfer* dan *message transfer protocols* telah ada yang mempunyai sistem dengan mengizinkan pengguna secara manual memindahkan datanya dari suatu tempat ke tempat lain. Protokol internetworking mendukung kemampuan dalam pelaksanaan dari *network file* dan *resource sharing protocols*.

Network file dan *resource sharing protocol* mengijinkan semua pengguna saling berbagi sumber daya dengan mudahnya, akan tetapi terdapat skema sistem yang dikerjakan pada proses ini. Pada saat *sharing* sumber daya, skema tersebut berkerja satu sama lain untuk menentukan siapa protokol yang menulis dan siapa administrator yang mengerjakan operasi. Dibawah adalah semua komponen yang bekerja pada sistem ini :

a. *File System Model* dan *Architecture* : suatu mekanisme yang mendefinisikan sumber daya dan berkas yang akan digunakan secara bersama – sama dalam jaringan.
b. *Resource Acces Method* : tahapan – tahapan yang mengambarkan bagaimana pengguna melampirkan ataupun melepaskan sumber daya yang berasal dari tempat penyimpanan lokal mereka.
c. *Operation Set* : untuk mengatur operasi apa yang akan digunakan dan diperlukan pada saat pengguna menggunakan sumber daya yang digunakan bersama pada tempat penyimpanan lokal pengguna lain.
d. *Messaging Protocols* : format pesan yang berisi operasi yang akan digunakan seperti informasi status dan protokol yang digunakan untuk bertukar pesan ini antar mesin pengguna.
e. *Administrative tool* : kumpulan fungsi yang dibutuhkan untuk mendukung operasi protokol dan penggunaan komponen lain yang mendukung.

Network File System (NFS)

Network File and *resource sharing protocols* sangat penting karena mengizinkan penggunanya saling berbagi sumber daya dengan begitu mudahnya. Kebanyakan sistem operasi yang digunakan oleh pengguna komputer adalah Microsoft. Jauh sebelum adanya sistem operasi Microsoft sesama pengguna jaringan komputer dapat saling berbagi sumber daya dengan menggunakan *Network File System* (NFS) yang lama telah disediakan oleh sistem operasi UNIX.

Network File System (NFS) diimplementasikan sebagai sebuah sistem client/server yang menggunakan perangkat lunak server NFS dan klien NFS. Server NFS akan menggunakan protokol NFS untuk mengekspor berkas yang dimilikinya kepada klien NFS. Berkas tersebut akan dibaca oleh klien NFS sebagai berkas lokal klien tersebut .

Network File System (NFS) umumnya menggunakan protokol Remote Procedure Call (RPC) yang berjalan di atas UDP dan membuka port UDP dengan nomor port 2049 untuk komunikasi antara klien dan server di dalam jaringan. Klien NFS selanjutnya akan mengimpor berkas dari server NFS, sementara server NFS mengekspor berkas lokal kepada klien. Mesin-mesin yang menjalankan perangkat server NFS dapat saling berhubungan dengan perangkat lunak klien NFS untuk membaca, menulis, memodifikasi, menghapus berkas dan direktori yang berada di dalam server dengan menggunakan permintaan RPC seperti halnya READ, WRITE, CREATE, dan MKDIR. Sebelum dapat mengakses berkas yang berada di dalam server NFS, administrator harus melakukan mounting (proses mengakses file atau sumber daya yang telah diijinkan) terlebih dahulu berkas pada server yang dapat diakses oleh klien dan menetapkan izin akses terhadap berkas atau direktori tersebut.

Arsitektur dan Operasi NFS

Network File sistem (NFS) menggunakan kerangka komponen utama yang didalamnya menerangkan operasi apa saja yang terjadi didalamnya. Standar External Data Resprentation (XDR) yang mendefinisikan bagaimana data terlibat dalam pertukaran antara server dan klien. Protokol Remote Procedure Call (RPC) adalah suatu mekanisme penggunaan ataupun pemanggilan prosedur yang berada pada komputer lain dalam jaringan komputer. Terakhir adalah mengatur operasi dan prosedur NFS yang bekerja menggunakan protokol RPC terhadap setiap permintaan, dan satu elemen tambahan yaitu protokol mount yang digunakan untuk me-mount perangkat atau direktori yang telah dijelaskan diatas.

Kata kunci lain pada saat pembuatan kerangka NFS adalah sederhana (yang berhubungan juga dengan performance). Protokol NFS adalah jenis protokol yang *conectionless*, ini artinya NFS tidak memerlukan pengelolahan informasi tentang protokol apa yang bekerja pada server. Klien tetap mengirimkan semua informasi kebutuhan untuk mengirimkan permintaan kepada server, tapi server tidak mempunyai informasi tentang permintaan NFS sebelumnya, atau sistematika hubungan permintaan NFS satu sama lain, hal ini yang menyebut bahwa NFS merupakan system yang *stateless*.

Standar dan Versi NFS

Karena pada awalnya NFS dirancang dan dipasarkan oleh Sun, NFS mulai menjadi standar de facto. NFS versi 2 resmi ditetapkan sebagai standar TCP / IP ketika RFC 1094, NFS: Network File System Protokol Spesifikasi, diterbitkan pada tahun 1989.

NFS Versi 3 kemudian dikembangkan, dan dikeluarkan pada tahun 1995 sebagai RFC 1813, NFS Version 3 Protocol Specification. Hal ini mirip dengan versi 2, namun terdapat beberapa perubahan dengan menambahkan beberapa kemampuan baru. Didalamnya mendukung perpindahan data yang lebih besar, dukungan lebih baik untuk menetapkan atribut file, dan beberapa akses berkas baru dan manipulasi prosedur. NFS versi 3 menyediakan perpindahan data yang lebih besar daripada versi 2 .

NFS Versi 4 dikeluarkan pada tahun 2000 sebagai RFC 3010 Protokol NFS versi 4. Dimana perubahan NFS versi 3 relatif kecil hanya berisi perubahan ke versi 2, hampir semua format dalam NFS versi 4 mengubah sistem dasarnya yang mencakup yang diantaranya sebagai berikut:

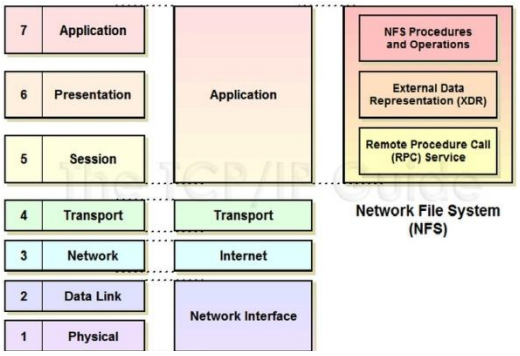
a. Mencerminkan kebutuhan internetworking modern, NFS versi 4 menempatkan penekanan lebih besar pada keamanan.
b. NFS versi 4 memperkenalkan konsep prosedur Compound, yang memungkinkan beberapa prosedur sederhana yang akan dikirim dari klien ke server sebagai sebuah kelompok.
c. NFS versi 4 penambahan prosedur yang dapat dilakukan klien dapat digunakan dalam mengakses file pada server NFS.
d. NFS Versi 4 juga membuat perubahan signifikan dalam pesan, dengan spesifikasi TCP sebagai protokol transport untuk NFS.
e. NFS versi 4 mengintegrasikan fungsi protokol mount ke protokol NFS dasar, yang merupakan protokol terpisah pada NFS versi 2 dan 3.

Arsitektur dan Komponen NFS

Network File System (NFS) adalah sebuah protokol yang berada pada layer aplikasi dari model TCP / IP. Sistem NFS bekerja meliputi lapisan sesi, presentasi dan aplikasi lapisan OSI Reference Model. Dalam beberapa kasus, lapisan-lapisan ini dapat membantu dalam memahami arsitektur dari sebuah protokol, dan itu yang terjadi dengan NFS.

Arsitektur dan Komponen Utama Pengoperasian NFS didefinisikan dalam bentuk tiga komponen utama yang dapat dipandang sebagai layanan yang berada di masing-masing dari tiga lapisan model OSI yang sesuai dengan TCP / IP application layer (lihat Gambar 2.1). Komponen-komponen ini adalah:

a. Remote Procedure Call (RPC):
   RPC adalah lapisan sesi pada lapisan OSI layanan ini umumnya digunakan untuk mengimplementasikan klien / server fungsi internetworking. RPC adalah sebuah program memanggil prosedur lokal pada komputer host tertentu, untuk pemanggilan prosedur pada perangkat remote melalui jaringan.
b. External Data Representation (XDR):
   XDR adalah bahasa deskriptif yang memungkinkan tipe data harus didefinisikan secara konsisten. XDR secara konseptual berada pada layer presentasi memungkinkan representasi data yang akan dipertukarkan menggunakan sistem NFS antara komputer.
c. NFS Prosedur dan Operasional:
   Fungsi sebenarnya dari NFS adalah diimplementasikan dalam bentuk prosedur dan fungsi operasi yang konseptual pada tujuh lapisan model OSI. Prosedur ini menentukan tugas-tugas tertentu yang akan dilaksanakan pada file melalui jaringan, menggunakan XDR untuk mewakili data dan RPC untuk menjalankan perintah di sebuah internetwork.



Gambar 2.1 : Arsitektur dan Komponen NFS

***Standar External Data Representation* (XDR)**

Standar External Data Representation (XDR) adalah untuk memungkinkan seseorang di satu komputer untuk membaca dari atau menulis ke berkas pada komputer lain semudah seperti yang mereka lakukan di mesin lokal. Tentu saja, berkas pada komputer lokal semua disimpan dalam sistem berkas yang yang sama, menggunakan struktur berkas yang sama dan sarana yang sama mewakili berbagai jenis data.

XDR bekerja dengan cara yang sama. Ketika informasi tentang cara mengakses sebuah file yang akan ditransfer dari perangkat A ke perangkat B, perangkat mengkonversikannya pertama dari A direpresentasi ke tipe data XDR tersebut. Informasi yang ditransmisikan melalui jaringan menggunakan enkoding XDR. Kemudian, perangkat B menerjemahkan dari XDR kembali ke representasi internal sendiri, sehingga dapat digunakan oleh pengguna seolah-olah pada sistem berkas lokal. Setiap perangkat hanya perlu tahu bagaimana mengkonversi dari bahasa mereka sendiri ke XDR dan kebalikannya. Perangkat A tidak perlu tahu pengkonversian ke bahasa B dan sebaliknya. Terjemahan semacam ini tentu saja merupakan pekerjaan klasik lapisan presentasi, dimana XDR ini berada. XDR itu sendiri didasarkan pada sebuah standar ISO yang disebut Abstract Syntax Notation.

XDR akan menjadi bahasa universal dengan meresprentasikan berbagi tipe data, itu harus memungkinkan penjelasan dari semua jenis data umum yang digunakan dalam komputer. Sebagai contoh, XDR harus membiarkan bilangan bulat, bilangan floating point, string dan konstruksi data lain

untuk dipertukarkan. Standar XDR menggambarkan banyak struktur tipe data dengan menggunakan notasi agak mirip dengan bahasa "C". Seperti yang diketahui, ini adalah salah satu bahasa yang paling populer dalam sejarah komputasi, dan sangat erat terkait dengan UNIX.

Remote Procedure Call (RPC)

Hampir semua aplikasi yang berkaitan dengan berkar dan sumber daya lain menggunakan operasi RPC. Ketika sebuah program perangkat lunak pada komputer tertentu yang ingin membaca sebuah file, menulis sebuah file atau melakukan tugas-tugas yang terkait, memerperlukan penggunaan instruksi perangkat lunak yang benar untuk tujuan ini. Untuk melakukan tindakan tersebut sebuah aplikasi melakukan panggilan prosedur tertentu. Prosedur yang menjalankan pengambil – alihan semantara untuk program utama dan melakukan tugas seperti membaca atau menulis data. Prosedur kemudian kembali mengontrol program dan kembali ke perangkat lunak yang memanggilnya, dan secara opsional mengembalikan data dalam keadaan baik.

RPC adalah proses sebenarnya dalam berkomunikasi NFS. NFS sebenarnya didefinisikan dalam istilah satu set prosedur dan operasi RPC yang tersedia pada server NFS yang digunakan oleh klien NFS. Prosedur dan operasi masing-masing memungkinkan suatu jenis tindakan yang harus diambil pada sebuah berkas, seperti membaca dari itu, menulis untuk itu atau menghapusnya.

RPC adalah layanan (*service*) yang dikendalikan oleh suatu program yang disebut *portmap*. Untuk melakukan proses *sharing* dan *mount* pada NFS, terdapat beberapa layanan yang bekerja secara bersama-sama yaitu :

a. nfs — menjalankan proses RPC untuk melayani permintaan sistem file NFS.
b. nfslock — layanan tambahan yang menjalankan proses RPC untuk mengijinkan NFS *client* untuk mengunci *file* pada *server*.
c. portmap — layanan RPC pada Linux yang merespon semua permintaan layanan RPC dan melakukan koneksi ke layanan RPC yang diminta.

Berikut ini adalah proses-proses RPC yang bekerja bersama-sama di belakang layar untuk memfasilitasi terjadinya layanan NFS

a. rpc.mountd — proses ini menerima permintaan mount (pengaktifan device/direktori) dan melakukan proses verifikasi sistem file yang di*eksport*. Proses ini dijalankan secara otomatis oleh service NFS dan tidak membutuhkan konfigurasi dari user.
b. rpc.nfsd — ini adalah proses utama NFS server yang bekerja pada kernel Linux untuk memenuhi kebutuhan NFS client .
c. rpc.lockd — merupakan proses tambahan yang mengijinkan NFS client untuk mengunci file pada server.
d. rpc.statd — Proses ini menjalankan Network Status Monitor (NSM) yaitu protokol RPC yang memberikan pesan kepada NFS client pada saat NFS server dijalankan ulang (restart). Proses ini dijalankan secara otomatis oleh service NFS dan tidak membutuhkan konfigurasi dari user.
e. rpc.rquotad — Proses ini menyediakan informasi kuota pemakai (user quota) untuk remote user. Proses ini dijalankan secara otomatis oleh service NFS dan tidak membutuhkan konfigurasi dari user.

Mount Protokol

NFS Sebelum dapat digunakan klien untuk mengakses berkas di server pada jaringan komputer, klien harus diberi cara untuk mengakses berkas. Ini berarti bahwa sebagian dari sistem berkas remote pada server harus tersedia untuk digunakan klien, dan berkas dibuka untuk diakses. Sebuah keputusan dibuat khusus ketika NFS diciptakan untuk menempatkan jenis akses berkas, membuka dan menutup fungsi ke dalam NFS . Protokol mount digunakan untuk bekerja sama dengan NFS untuk menyediakan pengambilan suatu berkas yang berda pada server.

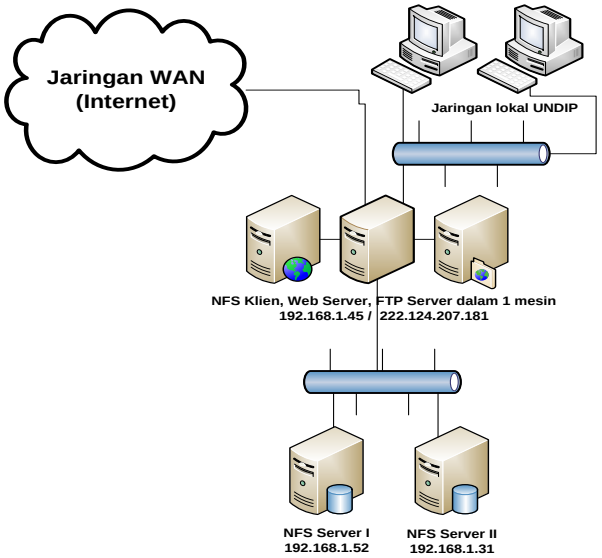
Implementasi aktual dari protocol mount ini sangat mirip dengan NFS itu sendiri. Seperti NFS protocol mount, menggunakan protokol XDR untuk menentukan tipe data yang akan dipertukarkan antara klien dan server, dan RPC untuk menetapkan satu set prosedur yang klien server dapat digunakan untuk melakukan operasi yang berbeda. Perbedaan utama antara protocol mount dan NFS adalah bahwa mount mendefinisikan prosedur-prosedur yang berkaitan dengan pembukaan dan penutupan *filesystem* daripada operasi akses file. Tabel 2.4 menunjukkan prosedur server yang digunakan dalam protokol Mount.

III. PERANCANGAN SISTEM

Perancangan sistem ini menggunakan minimal satu server yang berperan menjadi nfs server dan penambahan server dapat dilakukan dengan pengaturan secara khusus dibagian server dank lien. Pada sistem ini klien meminta sumber daya yang disediakan, pada peletakan server harus berada dalam satu jaringan komputer yang terhubung dengan satu sama lain, baik secara privat (jaringan yang alamat IP nya tidak perlu diketahui oleh jaringan internasional) ataupun publik.

Tahapan- tahapan perangcangan dibuat NFS server yang menangani permintaan *sharing* sumber daya kepada klien NFS. Pada tahapan pembuatan sistem NFS dibagi dua bagian utama yaitu server dan klien. Server NFS terdiri dua buah mesin dengan server NFS I yang mempunyai alamat IP 192.168.1.52 yang mempunyai sistem operasi Ubuntu server dan server NFS I yang mempunyai alamat IP 192.168.1.31 yang mempunyai sistem operasi FreeBSD server. Server Web dan FTP server terintegrate dalam satu mesin yang juga merupakan

Klien NFS yang mempunyai alamat IP 192.168.1.45 yang mempunyai sistem operasi Ubuntu server.

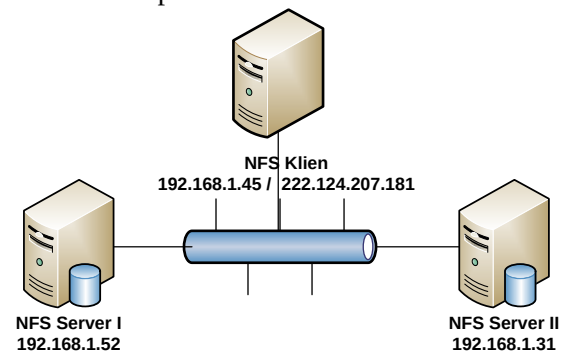


Gambar 6 Skema jaringan secara penuh

Perancangan yang dilakukan terdiri atas satu klien yang meminta sumber daya dengan dua server NFS. Penggunaan layanan – layanan pendukung dari sistem ini terdapat beberapa yang diantaranya server NFS, server Web dan server FTP yang masing – masing server tersebut melayani permintaan klien.

Perancangan Sistem NFS

Perancangan sistem NFS terdiri dari server dan klien yang menjadi kesatuan sistem. Server NFS mempunyai tugas sebagai penyedia sumber daya dan klien yang meminta sumber daya yang telah dibagikan tersebut. Server NFS terdiri dua buah mesin dengan server NFS I yang mempunyai alamat IP 192.168.1.52 yang mempunyai sistem operasi Ubuntu server dan server NFS I yang mempunyai alamat IP 192.168.1.31 yang mempunyai sistem operasi FreeBSD server. Klien NFS mempunyai alamat IP 192.168.1.45 yang mempunyai sistem operasi Ubuntu server. Klien



NFS melakukan protokol mounting untuk memasukan sumber daya kedalam sistem lokal mereka untuk melihat pemetaan protocol mounting yang dilakukan dapat dilihat pada table 3.1 dan skema perancangan sistem NFS dapat dilihat pada gambar 3.2 dibawah ini.

Gambar 3.2 Perancangan sistem NFS

Sumber daya yanga akan diekspor dari server terlihat pada tabel dibawah ini :

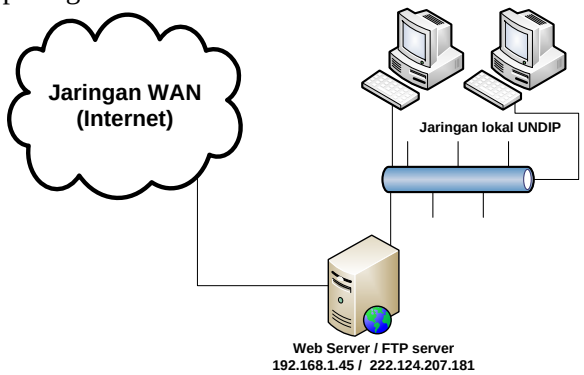
Tabel 3.1 : Susunan direktori pada proses NFS pada server dan klien

Alamat IP	Nama Berkas	
	Sumber	Tujuan
192.168.1.52	/home/ftp	/mnt/nfs/jaran/nfs 1
	/var/www/jaran /iso	/mnt/nfs/jaran/nfs 2
	/iso/data1	/mnt/nfs/jaran/nfs 3
192.168.1.31	/home/data1	/mnt/nfs/jaran/nfs 4
	/home/data2	/mnt/nfs/jaran/nfs 5
	/home/data3	/mnt/nfs/jaran/nfs 6

Tabel 3.1 melihatkan dikrektori yang berada pada server NFS dan selanjutnya dilakukan proses *mounting* yang dilakukan pada klien NFS.

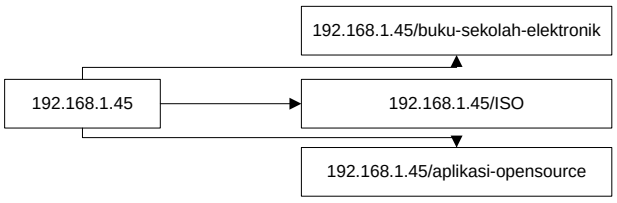
Perancangan web dan FTP server

Alamat yang digunakan dalam perancangan sama dengan yang ada pada perancangan NFS/ server Web dan FTP melayani permintaan yang berasal dari jaringan local dan public dari universitas diponegoro. Layanan ini bersasal dari web opensource UNDIP.Layanan yang diberikan antara server web dab FTP berbeda dapat dilihat dari gambar 3 dan gambar 3. . perancangan kedua sistem ini dapat dilihat pada gambar :

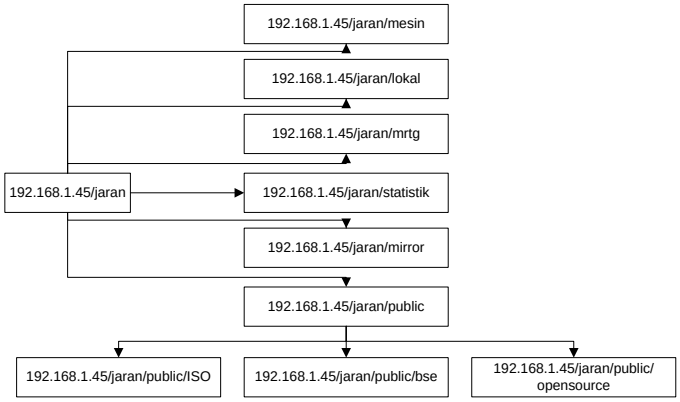


Gambar 3.3 Perancangan server web dan FTP

Perbedaan layanan server web dan ftp dapar dilihat pada gambar dibawah :



Gambar 3.6 Skema perancangan pada ftp server



Gambar 3.6 Skema perancangan pada web server

IV. IMPLEMENTASI DAN PENGUJIAN

Penanaman sistem ini, mengunakan mesin server IBM series dengan spesifikasi prossesor Intel(R)Xeon(R)CPU E5405 @ 2.00GHz, memory 1 GB, dan Hardisk 70 GB tipe SAS. Mesin server menggunakan sistem operasi Linux dan menggunakan turunan dari Debian, yaitu Ubuntu *Server* 9.10. Alasan digunakan turunan dari debian yaitu Ubuntu versi ini adalah karena *repository server* untuk server Ubuntu 9.10 mudah dijangkau sehingga diharapkan memudahkan proses konfigurasi. Sama dengan instalasi kebanyakan sistem operasi, instalasi Ubuntu menggunakan paket instalasi yang berupa CD.

Proses instalasi sistem operasi telah terinstall dengan baik, dilakukan pengaturan IP pada *server* tersebut serta daftar DNS agar *server* tersebut dapat terhubung dengan internet dan jaringan lokal. Setelah *server* dapat terhubung dengan baik ke internet dan jaringan lokal, pengaturan terhadap *server* tersebut dapat dilakukan melalui fasilitas *remote*, hal ini memungkinkan *server* dapat di akses darimana saja.

Pada Ubuntu instalasi program dapat dilakukan dengan menjalankan perintah "apt-get install <nama program>". Dengan syarat file /etc/apt/source.list telah berisi daftar repository. Penyedia repository yang digunakan pada implementasi ini adalah repo.undip.ac.id.

Aplikasi-aplikasi yang dipasangkan pada *server* untuk memungkinkan jalannya sistem NFS yang mendukung layanan server web dan FTP pada wen opensource UNDIP yaitu : :

- Server NFS sebagai pusat proses dan data sistem NFS, cukup dengan menginstal paket server NFS
- Klien NFS sebagai pihak yang meminta layanan sistem NFS, cukup dengan menginstal paket klien NFS
- Server Web dan FTP sebagai pihak yang mendistribusikan data yang berjalan pada Klien NFS, kedua aplikasi ini berkaitan erat dengan layanan yang diberikan pada web opensource Undip
- Aplikasi monitoring sebagai pihak yang berguna untuk memonitor layanan yang berjalan seperti Phpsysinfo, Mrtg dan Sedot sampai tua

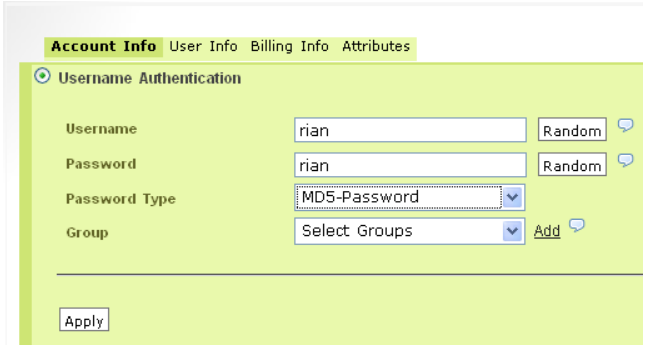
Komponen – komponen diatas saling bekerja sama membentuk suatu sistem NFS yang menyokong terbentuknya suatu layanan web opensource UNDIP menggunakan sistem NFS.

Konfigurasi Server NFS

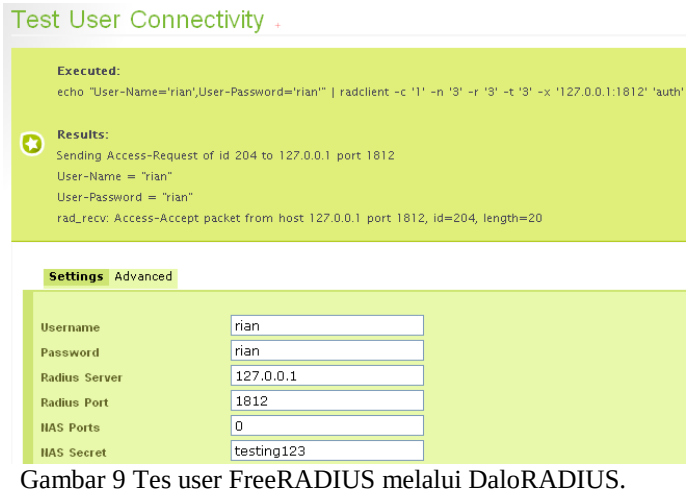
Server NFS yang digunanan terdapat didua mesin yang berbeda sistem operasi yaitu Ubuntu dan FreeBsd. Proses instalasi pada dua mesin ini sama yaitu dengan menginstal paket server NFS.Ubuntu mengunakan cara mengistal paket yang telah disediakan pada sistem tersebut dengan perintah :

```
root@jaran:~# apt-get install nfs-kernel-server nfs-common portmap
```

Gambar 7 Pengecekan status *server* dengan DaloRADIUS.



Gambar 8 Input user baru FreeRADIUS melalui DaloRADIUS



Gambar 9 Tes user FreeRADIUS melalui DaloRADIUS.

Agar RADIUS dapat bekerja dengan MySQL, ada beberapa pengaturan yang harus ditambahkan. Pengaturan pertama pada file `/etc/freeradius/sql.conf` perlu didaftarkan username dan password MySQL serta nama basis data yang disediakan untuk RADIUS, agar FreeRADIUS diberikan izin untuk mengakses basis data tersebut.

File `/etc/freeradius/clients.conf`, perlu diisi dengan daftar NAS(Networks Address *Server*)yang boleh mengakses *server* RADIUS, jika

NAS dan RADIUS berada pada satu *server* data diisikan dengan data localhost. File Selanjutnya adalah `/etc/freeradius/sites-enable/default.conf`. Pada file ini perlu diatur media penyimpanan data RADIUS, yaitu dengan menghilangkan tanda "#" sebelum teks "sql" pada isi file tersebut. Terakhir adalah file `/etc/freeradius/radiusd.conf`, baris 504, ubah kata status `proxy_request` menjadi `no` dan pada akhir bagian `modules` ditambahkan baris berikut.

```
pap {
authtype = md5
auto_header =  yes
}
```

Pemasangan HTTPS

Protokol HTTPS dapat dihasilkan dari *web server* apache yang dilengkapi dengan modul SSL berupa sertifikat. Pertama perlu dibuat kunci untuk sertifikat dengan perintah

#openssl genrsa -des3 -out *server*.key 1024

Kemudian buat sertifikat dengan perintah

#openssl req-new-key *server*.key –out *server*.csr
#openssl x509 -req -days 365 -in *server*.csr -signkey *server*.key -out *server*.crt

Perlu dibuat sertifikat "insecure" untuk memudahkan pengaktifan sertifikat.

#openssl rsa -in *server*.key -out *server*.key.insecure
#mv *server*.key *server*.key.secure
#mv *server*.key.insecure *server*.key

Baris-baris perintah di atas akan menghasilkan file-file sebagai berikut :

***server*.crt** : sertifikat yang dihasilkan oleh *server*
***server*.csr** : Permintaan penandatanganan sertifikat *server*.
***server*.key** : kunci pribadi (*Private*) *server*, yang tidak memerlukan *password* ketika memulai apache.
***server*.key.secure**: kunci pribadi *server*, yang memerlukan *password* ketika memulai apache.

Setelah sertifikat telah terbentuk, sertifikat ini perlu dimasukkan pada Apache sebagai modul tambahan. Pertama pindahkan sertifikat dan kunci agar mudah diambil.

#mv *server*.crt /etc/apache2/ssl/certs/
#mv *server*.key /etc/apache2/ssl/keys/

Setelah selesai, modul SSL diaktifkan dengan perintah.

#a2enmod ssl

dibuat satu situs khusus untuk akses HTTPS (HTTP lewat SSL):

#cp /etc/apache2/sites-available/default /etc/apache2/sites-available/ssl

Edit berkas `/etc/apache2/sites-available/ssl`, ubah 3 baris teratas menjadi seperti *snippet* berikut ini:

```
NameVirtualHost *:443
<VirtualHost *:443>
  ServerAdmin webmaster@localhost
  SSLEngine on
  SSLCertificateFile
/etc/apache2/ssl/cert/server.crt
  SSLCertificateKeyFile
/etc/apache2/ssl/keys/server.key
  DocumentRoot /var/secure/
#baris selanjutnya biarkan saja .....
```

Enable situs SSL yang baru dibuat tadi kemudian ubah situs *default* supaya tidak berbenturan dengan situs SSL.

 # a2ensite ssl
 # nano /etc/apache2/sites-available/default

Ubah 2 baris teratas berkas `/etc/apache2/sites-available/default` menjadi seperti *snippet* berikut ini:

```
NameVirtualHost *:80
<VirtualHost *:80>
#baris berikutnya biarkan saja.....
```

Dengan cara tersebut akses HTTP dan HTTPS akan dipisahkan baik *port* kerjanya ataupun letak file-file aplikasi *web* yang dijalankan. Apache juga harus mendengarkan *port* 443 untuk menerima permintaan HTTP, untuk itu ubah isi berkas `/etc/apache2/ports.conf` menjadi:

```
Listen 80
Listen 443
```

Simpan berkas tersebut kemudian *reload* Apache:

#/etc/init.d/apache2 force-reload

Untuk mencoba HTTPS bisa digunakan perintah

#nmap -A localhost|grep Apache

Konfigurasi OpenVPN

Setelah FreeRADIUS sudah dapat berjalan pada *server*, langkah selanjutnya adalah menambahkan aplikasi *OpenVPN* yang akan menangani *tunneling* pada jaringan publik sehingga dapat mengakses jaringan lokal. Agar kedua aplikasi ini dapat bekerja sama perlu ditambahkan suatu *plug-in* yang bernama **radiusplugin**, *plugin* ini nantinya akan menjadi penghubung antara aplikasi *OpenVPN* dengan FreeRADIUS. Radiusplugin bisa diunduh di alamat,

http://www.nongnu.org/radiusplugin/radiusplugin_v2.0c.tar.gz.

Selain paket OpenVPN ada paket lain yang harus di pasang agar OpenVPN dan radiusplugin dapat berjalan pada Ubuntu, paket tersebut adalah **libgcrypt11-dev dan g++**. Instalasi radiusplugin dilakukan dengan menjalankan perintah **make** pada folder radiusplugin. Proses instalasi radiusplugin adalah file radiusplugin.cnf dan radiusplugin.so, kedua file ini perlu dipindahkan ke folder openVPN. File radiusplugin.cnf perlu diubah, terutama pada password untuk mengakses radius dan alamat *server* radius dilihat dari letak file radiusplugin.cnf.

Langkah selanjutnya yang perlu dilakukan adalah pembuatan *Public Key Infrastructure*(PKI), yang berfungsi sebagai enkripsi data dan otentikasi klien. Untuk membuat PKI ini sudah tersedia *easy-rsa* yang telah disertakan oleh *OpenVPN*:

#cp -a /usr/share/doc/*OpenVPN*/examples/easy-rsa /etc
#cd /etc/easy-rsa/2.0/

Pembuatan PKI dapat dilakukan dengan menjalankan beris perintah berikut ini pada folder 2.0/

```
#source ./vars
#./clean-all
#./build-ca
#./build-key-server server
#./build-dh
```

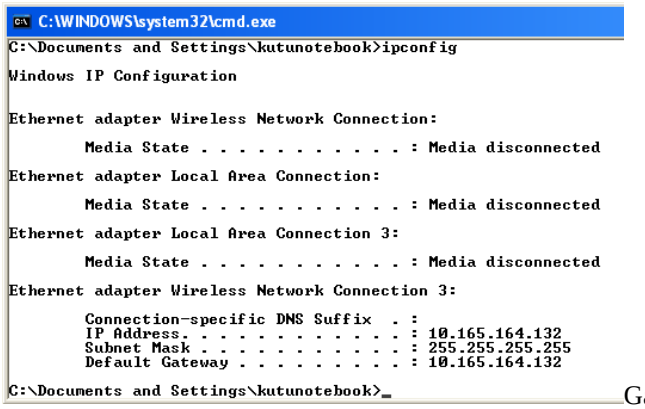
Hasil dari perintah-perintah di atas adalah *file* dh1024.pem, ca.crt, *server*.crt dan *server*.key yang terletak pada *folder* keys, *folder* keys perlu dipindahkan ke folder OpenVPN. File ca.crt tidak diperlukan *server* dan harus diberikan ke klien sebagai public key.

Untuk menjalankan fasilitas OpenVPN, harus dibuat file *server*.cnf yang berisi konfigurasi *server* VPN yang dibuat, seperti pengaturan nomor port, protokol yang digunakan, serta metode otentikasi dan enkripsi yang ingin digunakan. File *server*.cnf harus memanggil file-file radiusplugin dan PKI. File ini harus diletakkan pada folder OpenVPN. Menjalankan OpenVPN dapat dilakukan dengan perintah.

```
/etc/init.d/openvpn start
```

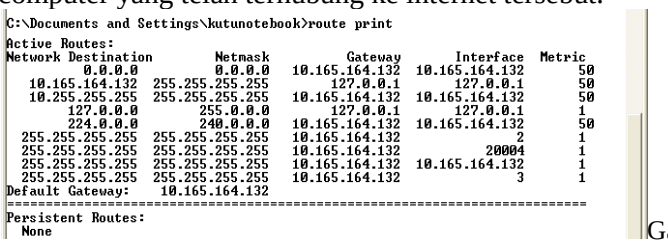
Pengujian OpenVPN

Pengujian aplikasi OpenVPN dapat dilakukan melalui komputer klien dengan menambahkan program OpenVPN GUI yang dapat didownload di http://OpenVPN.se. Agar program tersebut dapat berjalan perlu dibuat konfigurasi klien yang sesuai dengan konfigurasi pada *server*, serta file ca.crt yang bisa diambil di *server* VPN. Sebelum memulai sambungan ke *server* OpenVPN koneksi internet dari komputer pengguna perlu dicek terlebih dahulu.



Gambar 10 Konfigurasi IP pada klien dengan jaringan Indosat IM3

Selain alamat IP perlu diketahui juga tabel *routing* dari computer yang telah terhubung ke internet tersebut.

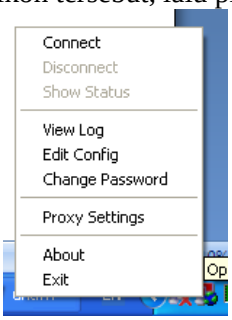


Gambar 11 Tabel *routing* klien sebelum sambungan VPN.

Tanda jika OpenVPN GUI sudah berjalan adalah ikon pada toolbar windows berupa dua layar yang jika berwarna merah maka menandakan *OpenVPN* GUI sudah berjalan tetapi belum tersambung pada *server* VPN manapun.

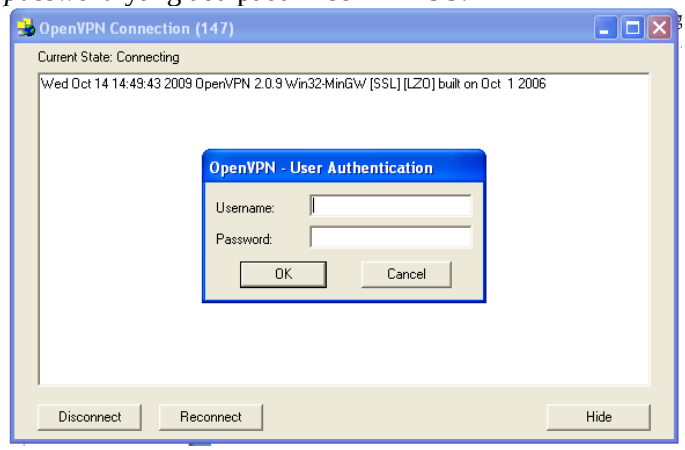
Gambar 12 Ikon yang menandakan OpenVPN GUI sedang Aktif.

Klik kanan pada ikon tersebut, lalu pilih *connect* .



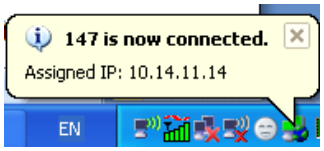
Gambar 13 Memulai sambungan ke OpenVPN.

Akan muncul kotak dialog yang menanyakan *username* dan *password*, isikan sesuai *username* dan *password* yang ada pada FreeRADIUS.



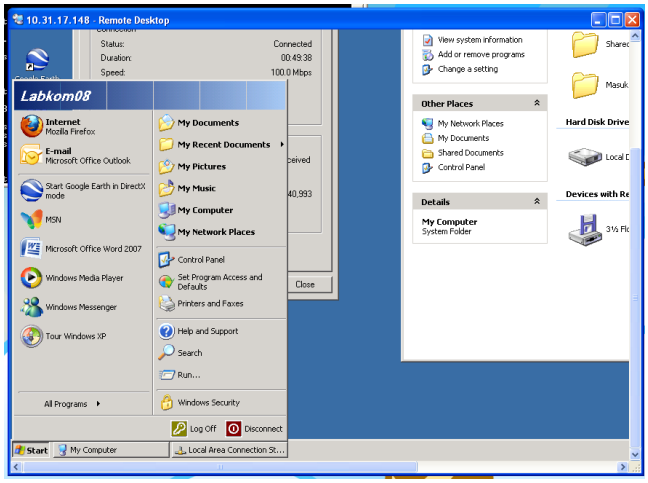
Gambar 14 Otentikasi Pada OpenVPN GUI.

Setelah proses penyambungan selesai ikon pada *toolbar* akan berubah menjadi hijau yang menandakan sambungan VPN telah terbentuk. Bisa dilihat pula alamat IP yang diberikan oleh *server OpenVPN*.



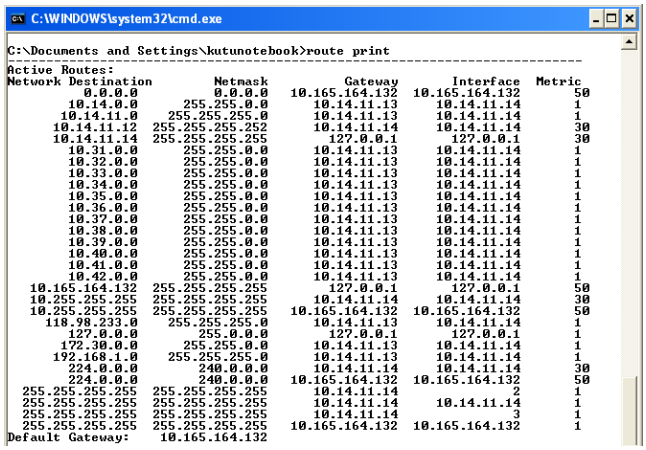
Gambar 15 *ikon* OpenVPN GUI yang berubah setelah tersambung

aplikasi-aplikasi lokal lain juga bisa diakses dengan memanfaatkan VPN ini. Aplikasi lokal tersebut antara lain adalah Remote Desktop dan juga File Sharing yang bisa dilihat pada gambar 4.22 dan 4.23.



Gambar 16 *Remote Desktop* ke komputer lokal

Pada komputer yang terhubung VPN telah ditambahkan tabel *routing*. Tabel routing memungkinkan komputer tersebut dapat menghubungi semua komputer dalam jaringan yang ada di tabel *routing*, untuk lebih jelasnya dapat dilihat pada tabel *routing* pada gambar 4.24.



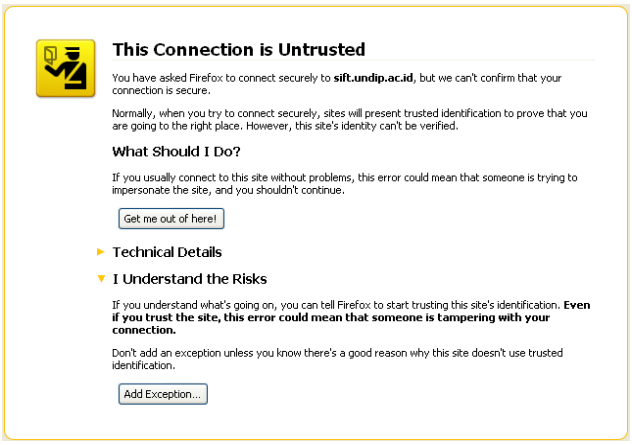
Gambar 17 Tabel *routing* pada klien setelah sambungan dengan *server* OpenVPN terbentuk.

Konfigurasi Glype Proxy

Agar Glype Proxy dapat berjalan, yang perlu dilakukan hanya meletakkan folder Glype Proxy tersebut pada /var/secure, yang merupakan folder untuk akses HTTPS. Pada implementasi ini, alamat *server web* proxy adalah sift.undip.ac.id, karena itu untuk mengaksesnya bisa dilakukan dengan mengetikan alamat https://sift.undip.ac.id, pada mesin broswer.

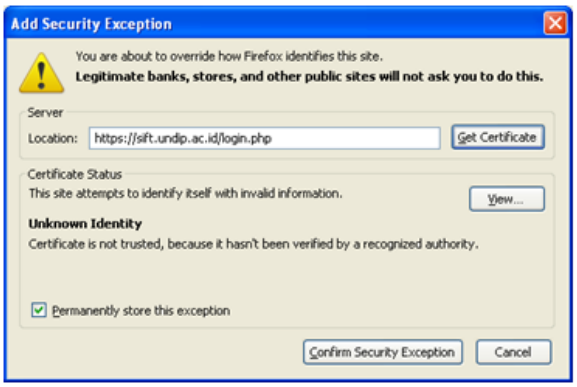
Pengujian Glype Proxy

Pengujian terhadap Glype Proxy dapat dilakukan dengan mengakses alamat dari *server* Glype Proxy. Saat akses untuk pertama kali akan muncul pesan bahwa sambungan tidak dipercaya, hal ini dikarenakan sertifikat yang dihasilkan *server* menanyakannya terlebih dahulu pada pengguna.



Gambar 18 Halaman yang muncul saat akan mengakses Glype Proxy.

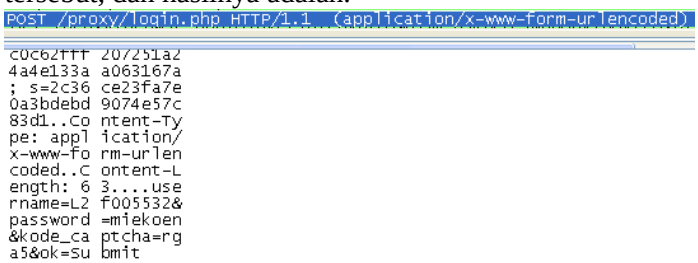
Pilih "*Add Exception*" untuk melanjutkan proses pemasangan sertifikat.



Gambar 19 Pemasangan Sertifikat secara manual.

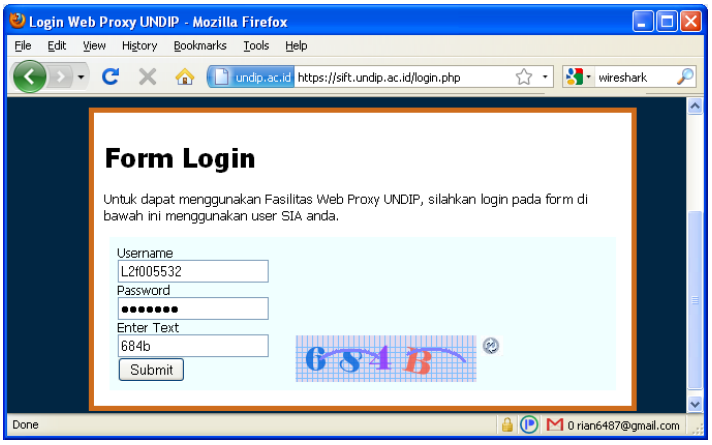
Pilih "*Get Certificate*" lalu "*Confirm Security Exception*" untuk menginstall sertifikat pada *browser*. Contoh diatas adalah untuk *browser* Mozilla Firefox versi 3.5 tiap *browser* memiliki halaman pesan yang berbeda tetapi pada intinya yang harus pengguna lakukan adalah menginstall sertifikat tersebut.

Gambar 19 di atas adalah form login yang diakses tanpa menggunakan protokol SSL, jika form tersebut telah di sumbit, data yang diisikan dapat terlihat di jaringan, di sini penulis menggunakan aplikasi *wireshark* untuk melihat paket-paket data tersebut, dan hasilnya adalah.



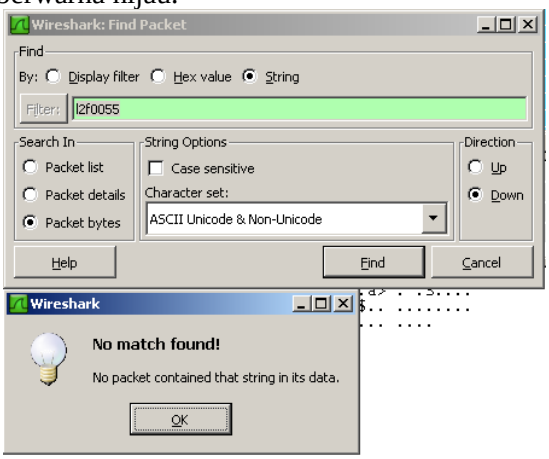
Gambar 20 Paket yang tertangkap oleh Wireshark.

Saat paket yang dikirim dilihat melalui wireshark, terlihat bahwa data yang dikirim ke jaringan masih dapat terbaca. Hal ini menjadi suatu celah keamanan karena data *username/password* dapat dilihat. Berbeda jika *web proxy* tersebut dilengkapi dengan modul SSL di *web server*nya, data *username/password* tersebut tidak dapat terlihat di jaringan atasnya.



Gambar 21 Halaman *Login* Glype Proxy menggunakan *protocol* HTTPS.

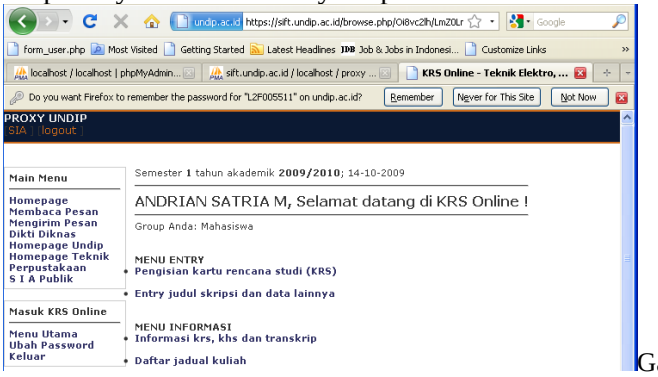
Contoh pada gambar 4.27 di atas adalah halaman *login* yang dilengkapi dengan modul SSL, Tanda bahwa suatu halaman *web* dilengkapi dengan modul SSL di *web server*nya adalah adanya tanda gembok pada pojok bawah mesin pencari, alamat URL juga berubah menjadi berwarna biru, menandakan sertifikat yang diinstall kurang keabsahannya, jika sertifikat tersebut dinyatakan sah secara penuh oleh mesin pencari, maka URL tersebut akan berwarna hijau.



Gambar 22 Paket-paket yang berjalan pada sambungan HTTPS.

Saat data *form* tersebut dikirim, paket data yang mengandung *username/password* tidak dapat ditemukan, jadi data aman dan tidak dapat diketahui isinya selain oleh *server* yang memiliki kunci dari data tersebut.

Jika proses *login* berhasil, yang artinya *username/password* tersebut terdapat pada *server* basis data SIA, selanjutnya *web proxy* akan langsung membuka halaman *sia.undip.ac.id*. Dan jika telah masuk ke *web proxy* ini layanan informasi lokal seperti *sia.ft.undip.ac.id* ataupun layanan lokal lainnya dapat dibuka.



Gambar 23 Masuk ke SIA FT menggunakan Glype Proxy.

PENUTUP

Kesimpulan

Dari hasil analisa dan pembahasan dapat disimpulkan bahwa :

1. Penggabungan kerja antara NFS server , Web server, FTP server dan aplikasi monitoring pada Ubuntu server 9.10 dapat berjalan dengan baik, ditandai dengan berjalannya sistem NFS yang dibangun dengan sistem operasi dan aplikasi-aplikasi tersebut.
2. Sistem NFS membantu mengatasi masalah tentang kebutuhan akan tempat penyimpanan yang terbatas dan terpisah secara geografis.
3. Instalasi lamp-server dapat dilakukan dengan mudah dengan sustu metode penginstalan tasksel.
4. Sistem NFS dapat bejalan di dua sistem operasi berbeda yaitu ubuntu server 9.10 dan FreeBsd.
5. Portmapper, Mountd dan NFS adalah tiga jenis RPC yang membangun sistem NFS.
6. Pemetaan direktori yang berasal dari sistem NFS tidak dapat langsung dipetakan kedalam server FTP. Perlu suatu proses mount dengan opsi bind.
7. Sistem NFS dapat mempunyai lebih dari satu server dengan satu klien.
8. Proses mounting pada sistem NFS ada dua cara yaitu manual dan otomatis.
9. Proses restart server NFS perlu dilakukan pada saat melakukan perubahan pada berkas /etc/exports dengan tujuan meregistrasikan perubahan tersebut kedalam server NFS.
10. Penggunaan aplikasi wordpress dapat dilkukan secara hierarki.
11. SNMP adalah protocol utama yang digunakan pada saat melakukan monitoring pada server.
12. Server NFS dengan tipe anonymous tidak dapat diakses bila permission pada direktori ftp bukan 755.

Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

1. Sistem NFS tidak hanya bisa berjalan pada jaringan lokal saja. Pengembangan sistem NFS ini dapat berjalan antar jaringan public yang berbeda.
2. Sistem operasi yang digunakan pada penelitian tidak hanya berbasis sistem operasi opensource saja. Penggabungan sistem NFS dapat pula digabungkan dengan sistem operasi yang bukan opersource

contohnya windows, SUN, dan sistem mainframe lainnya.

3. Pengaturan direktori sumber daya pada server sesuai dengan jenis dari sumber daya perlu dilakukan bertujuan memudahkan pengaturan direktorsi pada saat pengekporan.
4. Perancangan sistem NFS dengan lingkup yang lebih luas dengan jenis data yang dikhusukan kepada klien tertentu membutuhkan arsitekturs NFS yang baik serta perlunya membahas tentang factor keamanan.

DAFTAR PUSTAKA

[1]. Charles, M.Kozierok. 2003. The TCP/IP Guide. US : aquarelle.

[2]. Hal, Stren.2001. Managing NIS and NFS second edition. US : O'Relly and Associates inc

[3]. --,TCP/IP Network File System (NFS), http://www.tcpipguide.com/free/t_TCPIPNetworkFileSystemNFS.htm. Maret 2010

[4]. --,NFS Overview, History, Versions and Standards, http://www.tcpipguide.com/free/t_OverviewofFileandResourceSharingProtocolConceptsan.htm. Maret 2010

[5]. --,NFS Architecture and Components, http://www.tcpipguide.com/free/t_NFSArchitectureandComponents.htm. Maret 2010

[6]. --, NFS Data Storage and Data Types, and the External Data Representation (XDR) Standard, http://www.tcpipguide.com/free/t_NFSDataStorageandDataTypesandtheExternalDataRepres.htm. Maret 2010

[7]. --, NFS Client/Server Operation Using Remote Procedure Calls (RPCs), http://www.tcpipguide.com/free/t_NFSClientServerOperationUsingRemoteProcedureCallsR.htm. Maret 2010

[8]. --, NFS Server Procedures and Operations, http://www.tcpipguide.com/free/t_NFSServerProceduresandOperations.htm. Maret 2010

[9]. --, NFS File System Model and the Mount Protocol, http://www.tcpipguide.com/free/t_NFSFileSystemModelandtheMountProtocol.htm. Maret 2010

[10]. --, Introduction, http://tldp.org/HOWTO/NFS-HOWTO/intro.html. Maret 2010

[11]. --, Setting Up an NFS Server, http://tldp.org/HOWTO/NFS-HOWTO/server.html. Maret 2010

[12]. --, Setting up an NFS Client, http://tldp.org/HOWTO/NFS-HOWTO/client.html. Maret 2010

[13]. --, Troubleshooting, http://tldp.org/HOWTO/NFS-HOWTO/troubleshooting.html. Maret 2010

[14]. --, FTP server di debian menggunakan vsftpd, http://nugrahadi.pramono.info/2008/07/17/ftp-server-di-debian-menggunakan-vsftpd/. Maret 2010

[15]. --, NFS - sharing file systems across a network, http://www.freebsddiary.org/nfs.php. Maret 2010

[16]. --, Using NFS, http://www.freebsdmadeeasy.com/tutorials/freebsd/using-freebsd-nfs.php. Maret 2010

[17]. --, MRTG, Monitoring Network Yang Tidak Pernah Usang http://www.catatanlepas.com/component/content/article/108.html. Maret 2010

[18]. --, Instalasi awstats di ubuntu server 8.10, http://agungprasetyo.net/2008/11/11/instalasi-awstats-di-ubuntu-server-810/. Maret 2010

[19]. --, Instalasi, http://id.wordpress.org/. Maret 2010

BIODATA

AJIE PRASETYO (L2F005506) Dilahirkan di Jakarta 21 tahun yang lalu. Menempuh Pendidikan sampai sekolah menengah atas di jakarta. Dan semenjak tahun 2005 hingga kini sedang menyelesaikan studi Strata- 1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, Kensentrasi Informatika dan Komputer.

Menyetujui,
Dosen Pembimbing I

Adian Fatchur Rochim, S.T., M.T.
NIP. 19730226 1988021 001

Dosen Pembimbing II

Ir. Kodrat I. Satoto M.T.
NIP. 19631028 1993031 002