

REMOTE VIRTUAL THE DATA STORAGE

Rendra Gustriansyah

Program Studi Teknik Informatika, STMIK Indo Global Mandiri

Email: rendra_g@yahoo.com

ABSTRAK: Hampir setiap pengguna komputer mengenal *Windows Explorer*. Karena dengan *Windows Explorer*, user dapat melakukan operasi *file/direktori*, seperti melihat struktur hirarki dari direktori, menyalin *file* dan lain sebagainya. Dengan kata lain, *Windows Explorer* memiliki fasilitas pengaturan *file (file manager)*. Penelitian ini akan membahas pembangunan aplikasi penyimpanan data berbasis *web* yang mempunyai fasilitas pengaturan *file (remote virtual the data storage)* dengan hak akses yang berbeda (*privileges*). Penelitian ini dapat diimplementasikan untuk manajemen data perkuliahan di kelas maupun di laboratorium komputer suatu perguruan tinggi.

Kata kunci: *remote virtual the data storage, file manager, web, NAS.*

ABSTRACT: Almost all computer users know *Windows Explorer*. With the *Windows Explorer*, a user could do the operation *file/directory*, such as seeing the structure of the hierarchy from the directory, copied some files et cetera. In other words, a *Windows Explorer* had *file manager facilities*. This research will discuss the development of the application of the data storage based on *web* that had *file manager facilities (remote virtual the data storage)* with the *privileges*. This research could be applied for the management of the lectures data in the class or in the computer laboratory of a institution.

Keywords: *remote virtual the data storage, file manager, web, NAS.*

PENDAHULUAN

Seiring dengan perkembangan sistem informasi yang fleksibel, kredibel dan *mobile*, mahasiswa sebagai peserta ajar juga menuntut adanya kemudahan mengakses dan mengatur data perkuliahan, baik di laboratorium maupun di kelas tanpa adanya marginal waktu dan tempat.

Hal inilah yang menjadi dasar pemikiran untuk membangun aplikasi penyimpanan data perkuliahan secara maya (*remote virtual the data storage*) standar dengan fasilitas pengaturan *file (file manager)* sehingga dapat meningkatkan proses pembelajaran.

Bahkan sistem *remote virtual the data storage* ini bisa dikaitkan dengan Sistem Informasi Perpustakaan sehingga peserta ajar dapat mencari bahan referensi dalam mengerjakan tugas perkuliahan mereka. Konsep dasar penyimpanan data ini dikenal dengan istilah *Network-attached storage (NAS)*.

DASAR TEORI

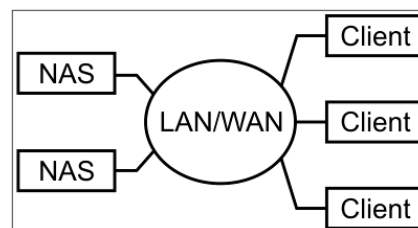
Storage

Storage adalah tempat menyimpan instruksi program dan data untuk keperluan penggunaan dengan komputer [5].

Network-Attached Storage [2]

Network-attached storage (NAS) adalah suatu konsep penyimpanan bersama pada suatu jaringan. NAS berkomunikasi menggunakan *Network File*

System (NFS) untuk UNIX, *Common Internet File System (CIFS)* untuk Microsoft Windows, FTP, http, dan protokol *networking* lainnya. NAS membawa kebebasan *platform* dan meningkatkan kinerja bagi suatu jaringan. NAS *device* biasanya merupakan *dedicated single-purpose machine*. NAS dimaksudkan untuk berdiri sendiri dan melayani kebutuhan penyimpanan yang spesifik dengan sistem operasi mereka dan perangkat keras/perangkat lunak yang terkait. NAS mirip dengan alat *plug-and-play*, akan tetapi manfaatnya adalah untuk melayani kebutuhan penyimpanan. NAS cocok digunakan untuk melayani jaringan yang memiliki banyak *client, server*, dan operasi yang mungkin menangani *task* seperti *web cache* dan *proxy, firewall, audio-video streaming, tape backup*, dan penyimpanan data dengan *file serving*. Adapun bagan umum arsitektur NAS dapat dilihat pada Gambar 1.



Gambar 1. NAS

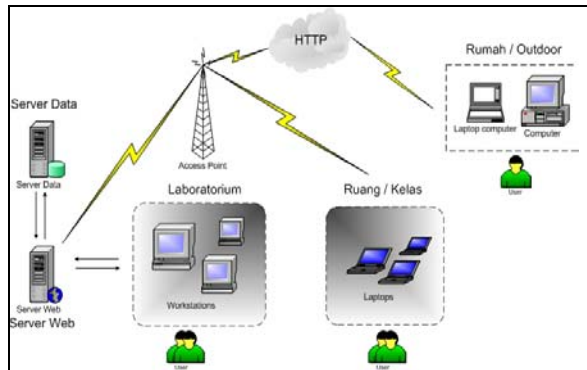
NAS menyediakan keamanan dan melaksanakan semua berkas serta *storage service* melalui protokol standard jaringan. Seperti menggunakan TCP/IP

untuk transfer data, Ethernet dan Gigabit Ethernet untuk media akses, serta CIFS, http, dan NFS untuk *remote file service*. Sebagai tambahan, NAS dapat melayani UNIX dan Microsoft Windows untuk berbagi data yang sama antar arsitektur yang berbeda. Untuk client, NAS adalah teknologi pilihan untuk menyediakan penyimpanan dengan akses *unencumbered* ke berkas.

Walaupun NAS menukar kinerja untuk manajebilitas dan kesederhanaan, namun NAS bukan merupakan *lazy technology*. Gigabit Ethernet memungkinkan NAS untuk memilih kinerja yang tinggi dan latensi yang rendah, sehingga mungkin untuk mendukung banyak sekali *client* melalui suatu antarmuka tunggal. Banyak NAS *devices* yang mendukung berbagai antarmuka dan dapat mendukung berbagai jaringan pada waktu yang sama.

Arsitektur Remote Virtual The Data Storage (RVDS)

RVDS menerapkan arsitektur *client-server three-tiers*, dimana *application web server* akan bertindak sebagai *middle-tier* yang menyediakan seluruh aplikasi yang diperlukan *user* sebagai *front-end* untuk berinteraksi dengan *file server* sebagai *back-end* (Gambar 2).



Gambar 2. Arsitektur RVDS

RVDS dibangun dengan menggunakan perangkat lunak JavaScript dan VBScript yang berjalan pada sisi *client*. Dan pada sisi *application web server* digunakan *Active Server Pages (ASP)* dan *IIS*. Sedangkan sisi *file server* hanya merupakan tempat penyimpanan data yang menggunakan sistem operasi *Windows 2K Professional*.

DESAIN

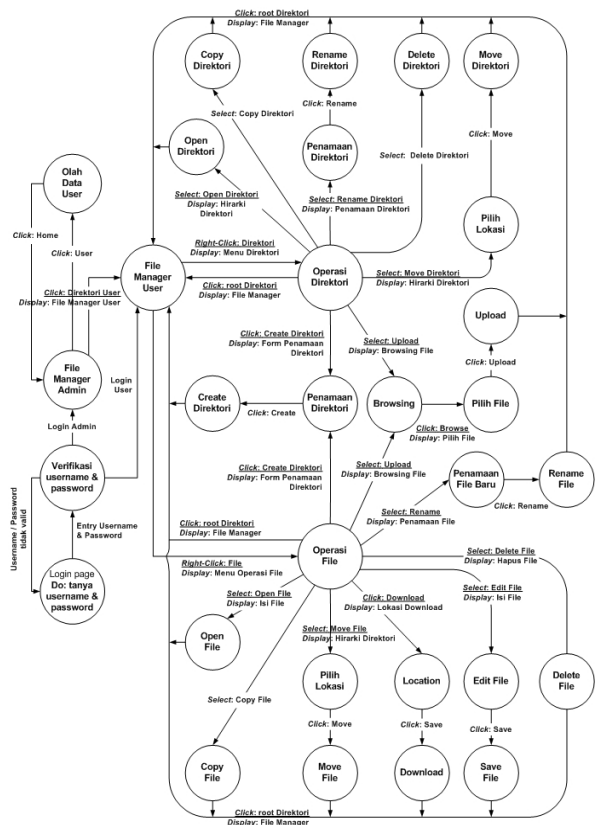
Aplikasi yang akan dibangun dalam penelitian ini dirancang menggunakan antarmuka (*user interface*) yang *friendly* berbasis web dengan fasilitas *file manager* sehingga memungkinkan memiliki spesifikasi (*features*) sebagai berikut:

- *User* dapat membuat/mengedit/menghapus/memindahkan direktori/file/data yang *user* miliki, tetapi tidak dapat melihat direktori/file/data *user* lain;
- *User* dapat men-*upload* atau men-*download file* atau data yang dimiliki;
- Hanya *user* yang telah terdaftar/registrasi yang dapat mengakses *server file*;
- *Administrator* dapat memberikan definisi tipe akses yang akan diberikan pada *user* apakah akses tersebut berupa *read* dan/atau *write*;
- Aplikasi akan dibangun dengan *Microsoft Active Server Pages (ASP)*.

Desain State Transition Diagram (STD)

State Transition Diagram (STD) merupakan suatu diagram yang terdiri atas lingkaran untuk melambangkan *state* dan garis untuk melambangkan transisi antar *state* [8].

State login page pada Gambar 3 merupakan halaman utama dari aplikasi *Remote Virtual the Data Storage*, dimana *admin/user* dapat mengentri *username* dan *password*-nya. Kemudian aplikasi akan memverifikasi *username* dan *password* *admin/user* tersebut. *Flowchart* proses verifikasi seperti terlihat pada Gambar 5. Setelah proses verifikasi, *user* dapat beralih ke *state File Manager*.



Gambar 3. State Transition Diagram Remote Virtual The Data Storage

Pada *state File Manager Admin*, administrator dapat mengolah data *user* yang meliputi menambah, menampilkan, mencari, mengedit dan menghapus data *user* (dosen/mahasiswa). Disamping itu, administrator dapat juga mengakses *state File Manager User* untuk mengatur data *user*.

Pada *state File Manager User*, administrator/*user* dapat melakukan operasi *file* dan operasi direktori. Operasi *file* terdiri dari *open file*, *copy file*, *download file*, *edit file*, *delete file*, *rename file*, *upload file*, *move file*, dan *create* direktori. Sedangkan operasi direktori meliputi *open* direktori, *copy* direktori, *rename* direktori, *delete* direktori, *create* direktori, *move* direktori, dan *upload file*.

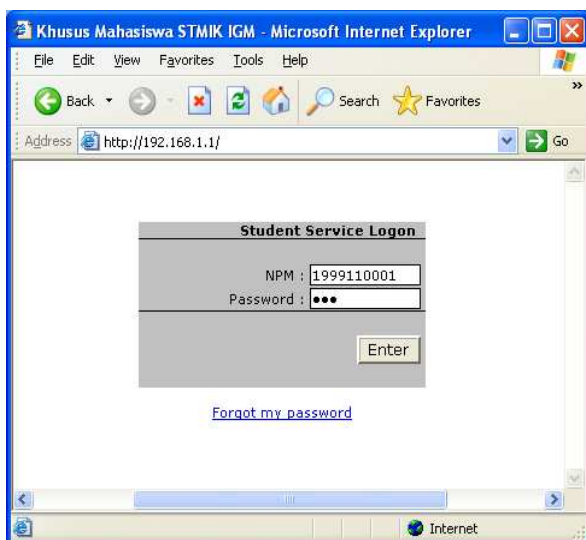
IMPLEMENTASI

Aplikasi *remote virtual the data storage* ini memiliki dua halaman utama, yaitu halaman untuk administrator, dan halaman untuk *user* (dosen/peserta ajar).

Halaman untuk administrator digunakan untuk manage data keseluruhan, mengatur hak *user* (dosen dan peserta ajar) dan mengakses seluruh data *user* (Gambar 6a). Sedangkan halaman untuk *user*, hanya dapat manage data *user* itu sendiri tanpa bisa melihat data *user* lain (Gambar 6b).

Langkah awal, seorang *user* haruslah terhubung ke intranet/internet dan memiliki program *web browser* untuk mengakses halaman-halaman *web*.

Setelah membuka *browser web* maka *user* akan melakukan permintaan alamat HTTP dengan memanggil alamat situs *remote virtual storage*. *User* yang telah terhubung dengan *server web* dapat mengakses *server* dan melakukan manajemen *file user* setelah melewati proses validasi seperti terlihat pada Gambar 4.



Gambar 4. Login

Terlihat pada Gambar 4 digunakan NPM: 1999110001, yang merupakan Nomor Pokok Mahasiswa. Proses validasi ini melibatkan keanggotaan *user* pada *database* yang terdapat dalam *server web* ini.

Saat validasi *user*, aplikasi akan mengecek *username* yang dientri oleh *user*. Jika *username* berisi data "admin" maka aplikasi akan menggunakan tabel admin yang berisi *field* nama dan *password*.

Jika *username user* mengandung kode dosen maka aplikasi akan menggunakan tabel dosen yang memuat tiga *field*, yaitu *field* *nirp*, *nm_dsn*, dan *passwd*. Sedangkan, jika *username user* mengandung kode peserta ajar maka aplikasi akan menggunakan tabel *mhs*, yang memuat *field* *npm*, *nm_mhs*, dan *passwd*. Ketiga tabel tersebut berada dalam *database* yang diberi nama *user.mdb*. *Database* ini bisa diimport dari *database* perguruan tinggi.

Selanjutnya, aplikasi ASP akan melakukan proses pengecekan *npm* dan *password user* ke *database*. Jika validasi sah maka *user* akan dibawa langsung ke dalam direktori kerja *user (File Manager)* seperti pada Gambar 6b, dimana *user* bisa melakukan proses manajemen *file/direktori user* dengan tingkat akses data yang terbatas hanya pada *user* atau administrator saja. *User* tidak dapat melihat, apalagi melakukan perubahan data *user* lain.

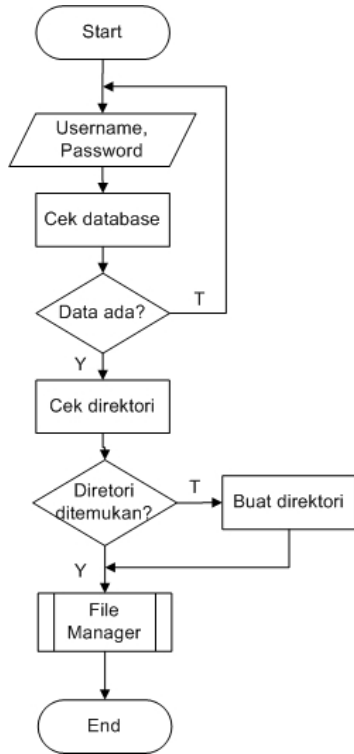
Saat *user login*, aplikasi akan mengecek apakah direktori dengan nama **npm user** telah tersedia atau belum, jika belum maka program akan membuat direktori secara otomatis sesuai dengan bagan alirnya seperti pada gambar 5. Direktori inilah yang merupakan basis *remote virtual the data storage*.

Tingkat manajemen data yang bisa dilakukan *user* terbatas hanya untuk melihat isi direktori/*file*, mengubah isi *file*, memindahkan data dari *client* ke *server* atau sebaliknya, membuat direktori baru, memindahkan direktori, menyalin direktori/*file*, dan mengedit nama direktori/*file* (lihat Gambar 7a dan Gambar 7b).

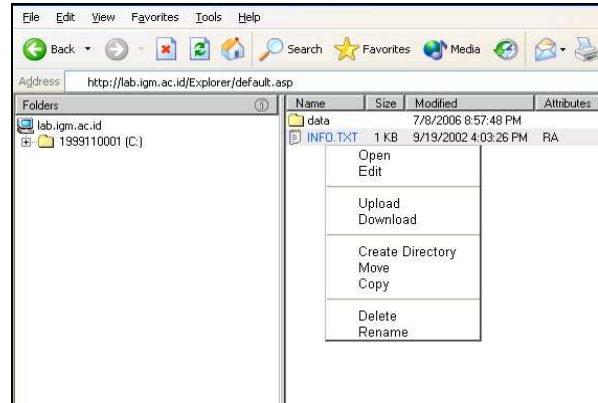
Administrator dapat membatasi kapasitas/daya tampung *data storage* setiap *user*, sehingga *user* akan memanfaatkan *data storage* dengan efisien. Selain itu, administrator dapat juga membatasi ukuran *file* yang akan dipindahkan ke *server (upload)* sehingga beban *server* bisa dikurangi.

Pengujian

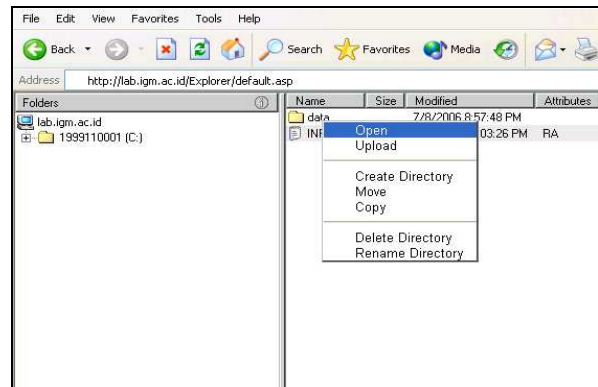
Saat pengujian perangkat lunak, *server web* yang digunakan memiliki konfigurasi komputer Pentium IV 2,66 GHz RAM 256 MB, DLINK Fast Ethernet. Sistem Operasi yang dipakai adalah *Microsoft Windows 2K Advanced Server* yang didukung *Internet Information Services 3.0 (IIS)*. IIS dipergunakan untuk melayani permintaan *user* dalam mengakses *server web* tersebut, segala informasi mengenai *server web* diatur pengaksesannya oleh IIS.



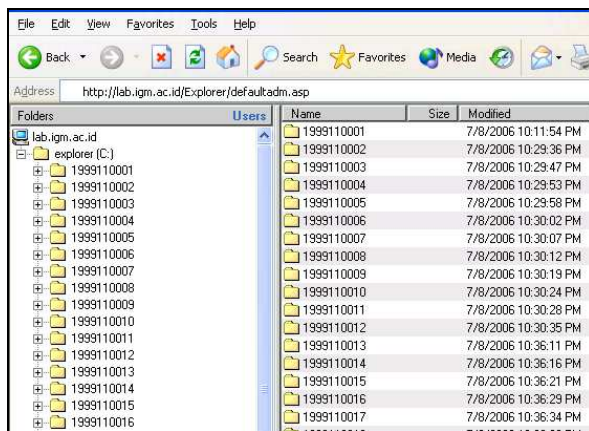
Gambar 5. Flowchart Verifikasi User dan Direktori User



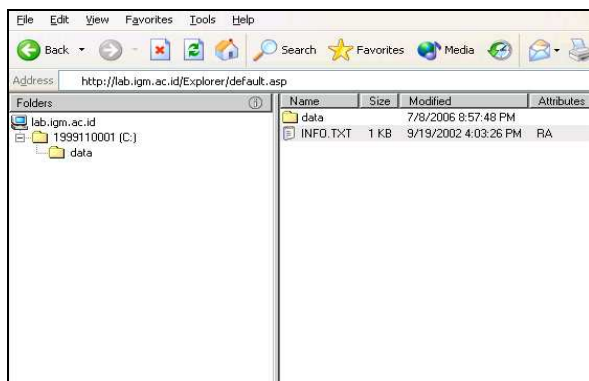
Gambar 7a. Operasi File



Gambar 7b. Operasi Direktori



Gambar 6a. File Manager Admin



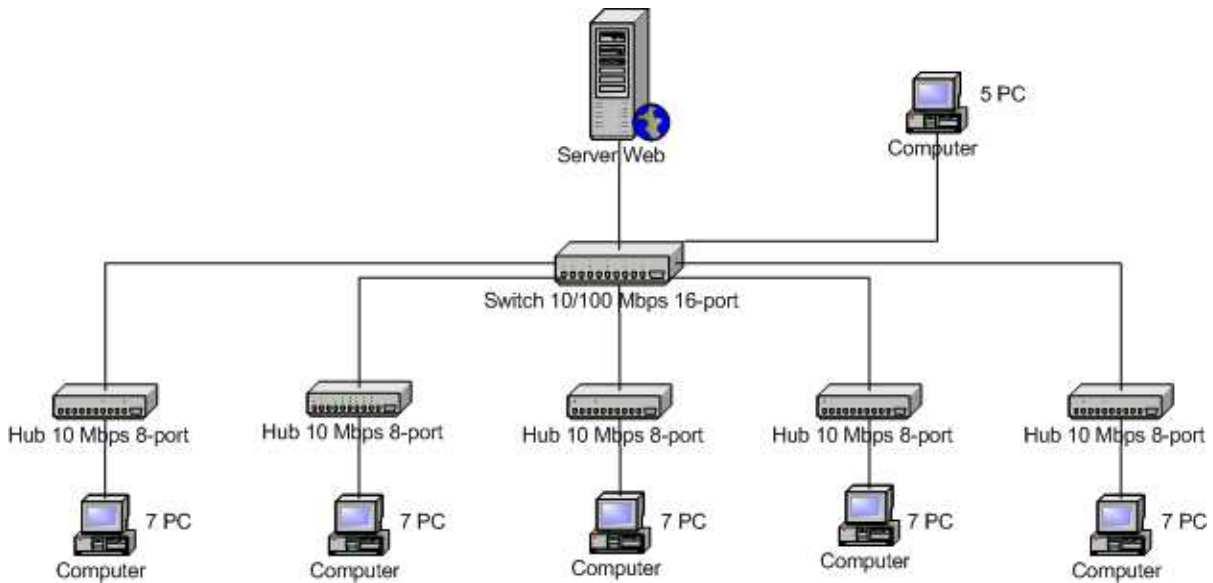
Gambar 6b. File Manager User

Komputer server web yang menjalankan aplikasi terhubung secara local area network (LAN) dengan empat puluh komputer user melalui switch 10/100 Mbps, seperti terlihat pada Gambar 8.

Setting IP (Internet Protocol) Address untuk komputer server web dalam pengujian ini adalah 192.168.1.1 atau http://lab.igm.ac.id, sedangkan halaman web default.htm yang nanti akan diakses oleh pengguna diletakkan pada direktori \explorer\. Dengan konfigurasi seperti ini, jika user ingin mengakses halaman utama harus memanggil alamat (url) http://192.168.1.1 atau http://lab.igm.ac.id, seperti pada Gambar 4.

Setelah aplikasi selesai dibangun, aplikasi diuji dengan software Ethereal Network Analyzer versi 0.99.0 [10] yang terinstal pada server web pada Gambar 8.

Saat pengujian, dua puluh user dalam waktu yang relatif bersamaan dengan menggunakan browser Internet Explorer 6.0 mengakses alamat http://192.168.1.1 (url server web). Seluruh user dikondisikan tidak melakukan aktivitas lain selain mengakses alamat http://192.168.1.1. Paket data dan bandwidth jaringan yang berhasil direkam oleh Ethereal terlihat pada Gambar 9a dan Gambar 9b. Terlihat pada Gambar 9a di atas bahwa rata-rata bandwidth LAN yang terpakai sekitar 3 kbps.



Gambar 8. Struktur LAN Uji Coba

Ethereal: Summary

File
 Name: D:\DOCUME~1\USER-011\LOCAL5~1\Temp\etherXXXXT6DPHT
 Length: 26560 bytes
 Format: libpcap (tcpdump, Ethereal, etc.)
 Packet size limit: 65535 bytes

Time
 First packet: 2006-10-14 09:48:40
 Last packet: 2006-10-14 09:49:46
 Elapsed: 00:01:05

Capture
 Interface: VIA Rhine II Fast Ethernet Adapter
 Dropped packets: unknown
 Capture filter: tcp port http

Display
 Display filter: ip
 Marked packets: 0

Traffic	Captured	Displayed
Between first and last packet	65,673 sec	65,673 sec
Packets	74	74
Avg. packets/sec	1,127	1,127
Avg. packet size	342,000 bytes	342,595 bytes
Bytes	25352	25352
Avg. bytes/sec	386,031	386,031
Avg. MBit/sec	0,003	0,003

Gambar 9a. Kondisi Paket Data dan LAN

Setelah itu, pada saat yang bersamaan dua puluh user tersebut melakukan login pada server web. Pada saat yang relatif bersama, server web akan menjalankan aplikasi untuk mengecek keabsahan setiap user yang login dan membuat direktori bagi setiap user yang berhasil login.

Ethereal (Capturing)

No.	Time	Source	Destination	Protocol	Length	Info
1	2006-10-14 09:48:40.654892	192.168.1.6	192.168.1.1	TCP	1035	192.168.1.1 [ACK] Seq=1 wfin=17520 Len=0
2	2006-10-14 09:48:40.654913	192.168.1.6	192.168.1.1	HTTP	GET	GET / HTTP/1.1
3	2006-10-14 09:48:40.655918	192.168.1.1	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
4	2006-10-14 09:48:40.655916	192.168.1.6	192.168.1.1	TCP	60	[TCP segment of a reassembled PDU]
5	2006-10-14 09:48:40.658955	192.168.1.6	192.168.1.1	TCP	1035	192.168.1.1 [ACK] Seq=1771 Ach=2921 wfin=17520 L
6	2006-10-14 09:48:40.658977	192.168.1.6	192.168.1.1	HTTP	HTTP/1.1 200 OK [text/html]	
7	2006-10-14 09:48:40.658955	192.168.1.6	192.168.1.1	TCP	1035	192.168.1.1 [ACK] Seq=1771 Ach=2921 wfin=17520 L
8	2006-10-14 09:48:40.658977	192.168.1.6	192.168.1.1	HTTP	HTTP/1.1 200 OK [text/html]	
9	2006-10-14 09:48:40.672084	192.168.1.22	192.168.1.1	TCP	1035	192.168.1.1 [ACK] Seq=1771 Ach=2921 wfin=17520 L
10	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
11	2006-10-14 09:48:40.672084	192.168.1.22	192.168.1.1	HTTP	GET / HTTP/1.1	
12	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
13	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
14	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
15	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
16	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
17	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
18	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
19	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
20	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
21	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
22	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
23	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
24	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
25	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
26	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
27	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
28	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
29	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
30	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
31	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L
32	2006-10-14 09:48:40.672084	192.168.1.1	192.168.1.22	TCP	1035	192.168.1.22 [ACK] Seq=1771 Ach=2921 wfin=17520 L

Gambar 9b. Kondisi Paket Data dan LAN Saat Mengakses Alamat Server Web

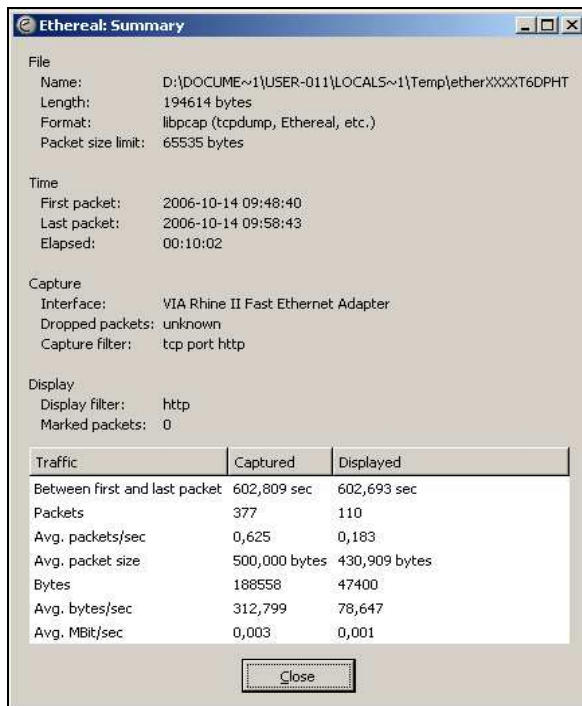
Setelah direktori user dibuat, aplikasi secara otomatis juga akan membuatkan sebuah direktori data dan sebuah file teks yang berisi informasi umum ke dalam direktori user tadi, seperti pada gambar 7a dan Gambar 7b. Kondisi paket data dan jaringan yang berhasil direkam oleh Ethereal pada proses ini terlihat pada gambar 10.

Selama proses pengujian berlangsung, komputer server tidak melakukan aktivitas lain. Hasil pengujian yang diperoleh sangat memuaskan. Dua puluh user (100 %) dapat login dan mengakses root direktori masing-masing secara bersamaan dengan menggunakan bandwidth sekitar 3 kbps.

Hal ini mungkin disebabkan jumlah user yang mengakses server web masih sedikit (puluhan) dan

jumlah *user* yang dicek keabsahannya kurang dari 2000 *user* sehingga kekhawatiran *server web* akan kacau (*hang*) dapat dihindari.

Hasil pengujian juga telah membuktikan bahwa ASP merupakan program yang dapat diandalkan untuk membangun aplikasi *remote virtual the data storage*.



Gambar 10. Kondisi paket data dan LAN setelah pembuatan direktori dan folder

KESIMPULAN

- *User* dapat melakukan penyimpanan data secara *remote*;
- *User* dapat melakukan operasi *file*/direktori seperti membuat/mengubah/menghapus direktori/*file*/data yang mereka miliki;
- *User* dapat men-*upload* atau men-*download file*/data yang mereka miliki;
- Keamanan data *user* terjamin dari 'keisengan' *user* lain;
- *User* dapat mengakses dan me-*manage* data perkuliahan di laboratorium komputer maupun di kelas melalui internet atau melalui komputer manapun yang terhubung dengan LAN tanpa adanya marginal waktu dan tempat;
- Aplikasi ini sangat berguna bagi sivitas akademika perguruan tinggi yang mempunyai fasilitas intranet/internet, yang bisa diakses melalui jaringan kabel lokal (LAN) maupun melalui *wireless*.

DAFTAR PUSTAKA

1. Masyarakat Digital Gotong Royong (MDGR). *Pengantar Sistem Operasi Komputer*, (Online), ([http://bebas.vlsm.org/v06/Kuliah/SistemOperasi/BUKU/SistemOperasi/ ch46s05.html](http://bebas.vlsm.org/v06/Kuliah/SistemOperasi/BUKU/SistemOperasi/ch46s05.html), diakses 10 Juni 2006).
2. (BenWhite@columbus.rr.com). *Advanced Remote File Explorer*, (Online), ([http://www. Planet-Source-Code.com/vb/scripts/ShowCode.asp?txtCodeId=7515&lngWId=4](http://www.Planet-Source-Code.com/vb/scripts/ShowCode.asp?txtCodeId=7515&lngWId=4), diakses 2 Desember 2002).
3. Ramamurthy, B. *File Management*, (Online), (<http://www.cse.buffalo.edu/faculty/bina/cse421/spring2005/FileSystemMar30.ppt>, diakses 14 November 2005).
4. Hall, Mike. *ASP File Operations*, (Online), (<http://brainjar.com/asp/fileops/>, diakses 12 Juni 2006).
5. (http://www.micro2000uk.co.id/hardware_glossary.htm, diakses 17 Juni 2006).
6. Jeffay, Kevin. *Secondari Storage Management*, (Online), ([www.cs.unc.edu/~jeffay/courses/ comp142/notes/ 15-SecondaryStorage.pdf](http://www.cs.unc.edu/~jeffay/courses/comp142/notes/15-SecondaryStorage.pdf), diakses 14 November 2005).
7. Nallipogu, Elsie et al. *Improving the Throughput of Remote Storage Access through Pipelining*. Proc. of the 3rd International Workshop on Grid Computing (GRID2002), Baltimore, MD, 18 November 2002 (Online), ([http://www.cse.ohio-state.edu/~lauria/ cse660/](http://www.cse.ohio-state.edu/~lauria/cse660/) - diakses 12 Juni 2006).
8. Free Online Dictionary of Computing. *State Transition Diagram*, (online), (<http://foldoc.org/foldoc/foldoc.cgi?state+transition+diagram>, diakses 8 Juli 2006).
9. Raishe, Thomas. *Black Box Testing*, (Online), (<http://www.cse.fau.edu/~maria/COURSES/CEN4010-SE/C13/black.html>, diakses 1 Juli 2006)
10. Combs, Gerald. *Ethereal Network Analyzer versi 0.99.0*, (Online), (<http://www.ethereal.com>, diakses 1 Oktober 2006).