Pembuatan Aplikasi
Pertukaran File Pada Jaringan Dengan
Memperhatikan Aspek Keamanan dan Efisiensi

Nama : Chandra Wijaya, S.T., M.T.

NIK : 21200

Program Studi Teknik Informatika

Fakultas Teknologi Informasi dan Sains

UNIVERSITAS KATOLIK PARAHYANGAN

2012

ABSTRAK

Dalam kegiatan sehari-hari yang menggunakan alat bantu komputer, data-data yang dipergunakan membutuhkan tempat penyimpanan di dalam komputer. Data-data yang berada pada satu komputer, terkadang dibutuhkan juga di komputer yang lainnya. Untuk keperluan tersebut, dapat digunakan flashdisk sebagai media penyimpanan sementara. Namun, bila komputer sudah terhubung dengan jaringan, komputer tersebut dapat langsung berbagi data tanpa menggunakan media tambahan lainnya. Kecepatan pertukaran data dipengaruhi oleh sumber daya yang tersedia. Semakin besar kapasitas bandwidth yang dimiliki, semakin cepat proses pertukaran file.

Untuk memperkecil ukuran data diperlukan sebuah aplikasi yang dapat memperkecil atau mengompresi data. Salah satu algoritma yang dapat dipergunakan untuk mengkompresi data adalah algoritma ZLIB. Secara spesifik, kompresi data bertujuan untuk mereduksi tempat penyimpanan data dan mereduksi waktu untuk mentransmisikan data yang memiliki kapasitas besar. Dengan memanfaatkan teknik kompresi ini, maka proses pengiriman data akan menjadi lebih maksimal dan mereduksi waktu transfer file. Dengan adanya kejahatan melalui internet, data pengguna akan semakin tidak aman dan menjadi intaian para penjahat yang menggunakan media internet. Data-data pribadi yang dikirimkan melalui internet, tidak seharusnya dapat dibaca dan dimiliki oleh orang lain yang tidak berkepentingan. Agar kerahasiaan data tetap terjamin saat dipertukarkan dibutuhkan suatu mekanisme untuk menjaga keamanan data tersebut.

Dalam penelitian ini, penulis akan membuat sebuah aplikasi yang dapat mempertukarkan file melalui internet, dengan memperhatikan aspek keamanan dan efisiensi bandwidth. Penulis akan menggunakan metode SSH dan library zlib untuk memenuhi kebutuhan tersebut.

ABSTRACT

In everyday activities using computer, data requires storage device in a computer. Data saved on a single computer, sometimes also needed when using the other computers. For this purpose, the usb-disk can be used as a temporary storage medium. However, if the computer is connected to the network, the computer can instantly share data without using any additional media.The speed of data exchange in the daily needs are influenced by available resources. For example, if the exchange of data is using the Internet, the bandwidth capacity is the main concern. The greater the bandwidth capacity, the faster file exchange process happened.

To minimize the size of the data being exchanged, the data needs to be compressed first. So the time needed for the delivery process is minimized. One algorithm that can be used to compress data is zlib algorithm. Specifically, data compression aims to reduce data storage and reduce the time to transmit data. By utilizing this compression technique, the data exchange process will be more leverage and reduce file transfer time. The data confidentiality becomes very important, when we use internet. The cybercrime through the Internet, will make users become unsafe. Personal data transmitted over the internet, should not be read and shared by others who are not related. To assured confidentiality of data exchanged, we need a mechanism to maintain data security. One proposed solution is to use an encryption method, a method used to secure data by changing the original data into a form that can not be read without having the key to turning that data into the form of the original data. There are several methods of encryption that can be used include Secure Shell (SSH) and Secure Socket Layer (SSL).

In this research, the author will make an application which can exchange data securely and efficiently in a network environment. The application will use zlib and SSH as the secure and efficient data exchange mechanism.

Daftar Isi

BAB I
PENDAHULUAN

1.1. Latar Belakang

Dalam kegiatan sehari-hari yang menggunakan alat bantu komputer, data-data yang dipergunakan membutuhkan tempat penyimpanan di dalam komputer. Data-data yang berada pada satu komputer, terkadang dibutuhkan juga di komputer yang lainnya. Untuk keperluan tersebut, dapat digunakan flashdisk sebagai media penyimpanan sementara. Namun, bila komputer sudah terhubung dengan jaringan, komputer tersebut dapat langsung berbagi data tanpa menggunakan media tambahan lainnya.

Kecepatan pertukaran data yang berukuran semakin besar dalam kebutuhan sehari-hari dipengaruhi oleh sumber daya yang tersedia. Misalkan, jika pertukaran data menggunakan internet, maka kapasitas bandwidth adalah sumber daya utama saat pertukaran tersebut. Semakin besar kapasitas bandwidth yang dimiliki, semakin cepat proses pertukaran file.

Untuk memperkecil ukuran data diperlukan sebuah aplikasi yang dapat memperkecil atau mengompresi ukuran data atau file tersebut pada saat pengiriman agar dapat menghemat penggunaan waktu kompresi. Salah satu algoritma yang dapat dipergunakan untuk mengkompresi data adalah algoritma ZLIB. Secara spesifik, kompresi data bertujuan untuk mereduksi tempat penyimpanan data dan mereduksi waktu untuk mentransmisikan data yang memiliki kapasitas besar. Dengan memanfaatkan teknik kompresi ini, maka proses pengiriman data akan menjadi lebih maksimal dan mereduksi waktu transfer file.

Kerahasiaan data yang dimilikipun menjadi sangat penting, karena ketergantungan kita dalam menggunakan komputer. Dengan adanya kejahatan melalui internet, para pengguna akan semakin tidak aman dan menjadi intaian para penjahat yang menggunakan media internet. Data-data pribadi yang dikirimkan melalui internet, tidak seharusnya dapat dibaca dan dimiliki oleh orang lain yang tidak berkepentingan.

Agar kerahasiaan data tetap terjamin saat dipertukarkan dibutuhkan suatu mekanisme untuk menjaga keamanan data tersebut. Untuk kepentingan pertukaran data antar komputer tersebut, penulis akan membuat perangkat lunak yang mementingkan aspek keamanan file yang dipertukarkan dan juga efisiensi dalam penggunaan sumber daya (bandwidth) yang dimiliki.

Salah satu solusi yang ditawarkan adalah dengan menggunakan metode enkripsi, yaitu sebuah metode yang digunakan untuk mengamankan data dengan mengubah data asli ke dalam bentuk yang tidak dapat dibaca tanpa memiliki kunci untuk mengubah data tersebut ke bentuk data asli. Ada beberapa metode enkripsi yang dapat dipergunakan diantaranya adalah Secure Shell (SSH) dan Secure Socket Layer (SSL)

1.2. Tujuan

Berdasarkan rumusan masalah diatas, maka tujuan dari penelitian ini adalah :

1. Membuat perangkat lunak untuk menunjang pertukaran file antar komputer yang mementingkan aspek keamanan data yang dikirimkan melalui jaringan.
2. Membuat perangkat lunak untuk menunjang pertukaran file antar komputer dengan mementingkan efisiensi sumber daya terutama bandwidth.

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Perangkat lunak yang dibuat menggunakan bahasa pemrograman java.
2. Sistem operasi yang digunakan adalah Microsoft Windows 7 dan FreeBSD.
3. Jaringan yang digunakan untuk pengujian adalah jaringan komputer internal Unpar.
4. Kecepatan transfer data dibatasi menjadi 1 MBps untuk upload dan download.

1.4. Metodologi Penelitian

Tahap – tahap yang dilakukan dalam penelitian ini adalah :

1. Melakukan studi pustaka mengenai kompresi data dan secure shell.
2. Membuat spesifikasi perangkat lunak.
3. Merancang perangkat lunak sesuai spesifikasi yang telah dirancang sebelumnya.
4. Melakukan pengujian terhadap perangkat lunak yang telah dibuat.
5. Penarikan kesimpulan.

1.5. Sistematika Pembahasan

Sistematika dalam penelitian ini adalah :

BAB 1. PENDAHULUAN

Pada bab ini akan dibahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

BAB 2. DASAR TEORI

Pada bab ini akan dibahas mengenai kompresi data dan secure shell yang digunakan untuk melakukan pengiriman data melalui jaringan.

BAB 3. PERANCANGAN DAN PENGUJIAN

Pada bab ini akan dibahas mengenai perancangan perangkat lunak yang sesuai dengan kebutuhan, kemudian dilakukan pengujian terhadap perangkat lunak yang telah dibangun.

BAB 4. KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan dan saran yang dapat digunakan untuk pengembangan penelitian selanjutnya.

BAB 2
TEORI DASAR

2.1. Secure Shell (SSH)

2.1.1. Pengertian

Pada awalnya SSH dikembangkan oleh Tatu Yl nen di Helsinki University of Technology. SSH memberikan alternatif yang secure terhadap remote session tradisional dan file transfer protocol seperti telnet dan rlogin. Protokol SSH mendukung otentikasi terhadap remote host, yang dapat meminimalkan ancaman pemalsuan identitas client lewat IP address spoofing. Selain itu SSH mendukung beberapa protokol enkripsi secret key (DES, TripleDES, IDEA, dan Blowfish) untuk membantu memastikan privacy dari keseluruhan komunikasi, yang dimulai dengan username/password.

SSH menyediakan suatu virtual private connection pada application layer, mencakup interactive logon protocol (ssh dan sshd) serta fasilitas untuk secure transfer file (scd). Implementasi SSH pada linux diantaranya adalah OpenSSH.

SSH merupakan paket program yang digunakan sebagai pengganti yang aman untuk rlogin, rsh dan rcp. SSH menggunakan public-key cryptography untuk mengenkripsi komunikasi antara dua host. SSH dapat digunakan untuk login secara aman ke remote host atau menyalin data antar host, sementara mencegah man-in-the middle attacks (pembajakan sesi) dan DNS spoofing.

2.1.2. Kegunaan SSH

Adapun SSH merupakan produk serbaguna yang dirancang untuk melakukan banyak hal. Kebanyakan adalah pembuatan tunnel antar host. Beberapa implementasi SSH tergantung pada library SSL karena SSH dan SSL menggunakan banyak menggunakan algoritma enkripsi yang sama (misalnya TripleDES), Algoritma enkripsi lain yang didukung oleh SSH di

antaranya BlowFish (BRUCE SCHNEIER), IDEA (The International Data Encryption Algorithm), dan RSA (The Rivest-Shamir-Adelman). Dengan berbagai metode enkripsi yang didukung oleh SSH, Algoritma yang digunakan dapat diganti secara cepat jika salah satu algoritma yang diterapkan mengalami gangguan.

Dua hal penting SSH adalah console login (menggantikan telnet) dan secure filetransfer (menggantikan FTP), tetapi dengan SSH anda juga memperoleh kemampuan membentuk source tunnel untuk melewatkan HTTP, FTP, POP3, dan apapun lainnya melalui SSH tunel.

Tanpa adanya traffic dari suatu aplikasi, SSL tidak melakukan apa-apa, tetapi SSH sudah membentuk encrypted tunel antara dua host yang memungkinkan untuk melakukan login shell, file transfer, dan lain sebagainya.

2.1.3. Cara Kerja SSH

Misalkan suatu client mencoba mengakses suatu linux server melalui SSH. SSH daemon yang berjalan baik pada linux server maupun SSH client telah mempunyai pasangan public/private key yang masing-masing menjadi identitas SSH bagi keduanya.

Langkah-langkah koneksinya adalah sebagai berikut :

1. Client bind pada local port nomor besar dan melakukan koneksi ke port 22 pada server.
2. Client dan server setuju untuk menggunakan sesi SSH tertentu. Hal ini penting karena SSH v.1 dan v.2 tidak kompatibel.
3. Client meminta public key dan host key milik server.
4. Client dan server menyetujui algoritma enkripsi yang akan dipakai (misalnya TripleDES atau IDEA).
5. Client membentuk suatu session key dan mengenkripsinya menggunakan public key milik server.

6. Server men-decrypt session key yang didapat dari client, meng-re-encrypt-nya dengan public key milik client, dan mengirimkannya kembali ke client untuk verifikasi.
7. Pemakai mengotentikasi dirinya ke server di dalam aliran data terenkripsi dalam session key tersebut.

Sampai disini koneksi telah terbentuk, dan client dapat selanjutnya bekerja secara interaktif pada server atau mentransfer file ke atau dari server. Langkah ketujuh diatas dapat dilaksanakan dengan berbagai cara (username/password, kerberos, RSA dan lain-lain).

2.2. Kompresi Data

2.2.1. Pengertian

Kompresi data (pemampatan data) merupakan suatu teknik untuk memperkecil jumlah ukuran data (hasil kompresi) dari data aslinya. Pemampatan data umumnya diterapkan pada mesin komputer, hal ini dilakukan karena setiap simbol yang muncul pada komputer memiliki nilai bit – bit yang berbeda.

Pengiriman data hasil kompresi dapat dilakukan jika pihak pengirim atau yang melakukan kompresi dan pihak penerima memiliki aturan yang sama dalam hal kompresi data. Pihak pengirim harus menggunakan algoritma kompresi data yang sudah dipilih dan pihak penerima juga menggunakan teknik dekompresi data yang sama dengan pengirim sehingga data yang diterima dapat dibaca kembali dengan benar.

Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan bandwidth.

2.2.2. Jenis Kompresi Data

Jenis kompresi data berdasarkan outputnya dibagi menjadi dua bagian yaitu:

1. Lossy Compression

   Yaitu teknik kompresi dimana data hasil dekompresi tidak sama dengan data sebelum kompresi namun sudah "cukup" untuk digunakan. Contoh : Mp3, streaming media, JPEG, MPEG, dan WMA.

   Kelebihan dari lossy compression yaitu ukuran file lebih kecil dibanding lossless namun masih tetap memenuhi syarat untuk digunakan. Biasanya teknik ini membuang bagian – bagian data yang sebenarnya tidak berguna, tidak begitu dirasakan, tidak begitu dilihat oleh manusia sehingga manusia masih beranggapan bahwa data tersebut masih bisa digunakan walaupun sudah dikompresi.

   Misal terdapat image asli berukuran 12,249 bytes, kemudian dilakukan kompresi dengan JPEG kualitas 30 dan berukuran 1,869 bytes berarti image tersebut 85% lebih kecil dengan ratio kompresi 15%.

2. Lossless Compression

   Yaitu teknik kompresi dimana data hasil kompresi dapat didekompres lagi dan hasilnya tepat sama seperti data sebelum proses kompresi. Teknik ini digunakan jika data setelah dikompresi harus diekstrak/dekompres lagi tanppa adanya kerusakan terhadap data yang telah dikompres.

   Teknik kompresi lossless ini banyak digunakan untuk berkas – berkas dokumen, dimana sama sekali tidak diperkenankan adanya perbedaan antara informasi awal (sebelum proses kompresi) dan informasi yang diperoleh setelah proses dekompresi.

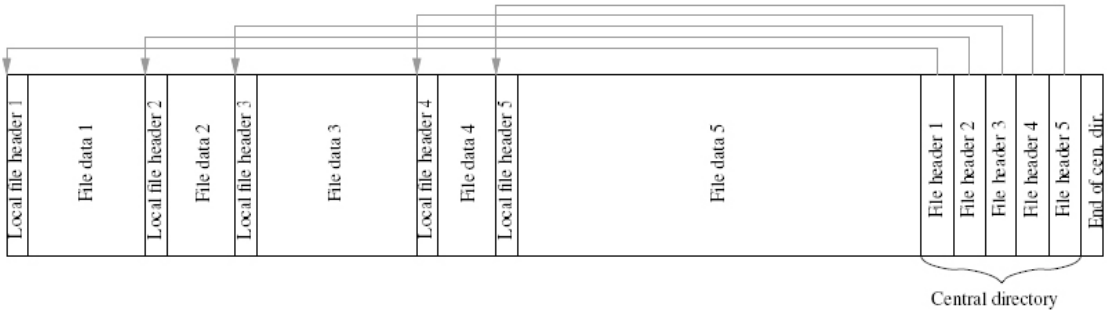
   Teknik kompresi ini mempertahankan kebutuhan informasi yang dikandung oleh data, sehingga informasi yang terkandung pada file yang telah terkompresi tetap

terjaga meskipun ukurannya telah berubah dari ukuran data aslinya.

2.2.3. Algoritma kompresi ZLIB

Algoritma ZLIB merupakan turunan dari Agoritma kompresi Deflate. Algortima ini diciptakan oleh Jean-Loup Gailly yang menciptakan proses kompresi data dan Mark Adler yang menciptakan proses dekompresi data. Algoritma Zlib melakukan kompresi dengan pengkompresian data yang terdiri dari serangkaian blok, sesuai dengan blok inputan data tersebut. Setiap blok pada data tersebut dikompresi dengan menggunakan algoritma kompresi data Deflate sebagai kompressor yang merupakan variasi dari Algoritma LZ77 dikombinasikan dengan Huffman Coding.

Data yang dikompresi oleh algoritma ZLIB akan diberikan pembungkus berupa data header setelah data tersebut dikompresi oleh algoritma Deflate. Header file yang diberikan pada data yang telah dikompresi tersebut merupakan bagian dari Algoritma ZLIB. Adapun struktur file dari algoritma ZLIB beserta header file yang telah diberikan pada data yeng telah dikompresi ditunjukkan pada gambar 2.1.



Gambar 2.1. struktur file dari algoritma ZLIB

BAB 3
PERANCANGAN DAN PENGUJIAN

3.1 Perancangan Jaringan

Jaringan yang digunakan untuk pengujian terhadap perangkat lunak yang dibuat adalah menggunakan jaringan internal Unpar. Pada jaringan internal Unpar, terdapat beberapa subnet yang terhubung satu sama lain. Antara komputer client dan server, terdapat 3 hop dengan kondisi kepadatan jaringan yang berbeda-beda setiap detiknya.

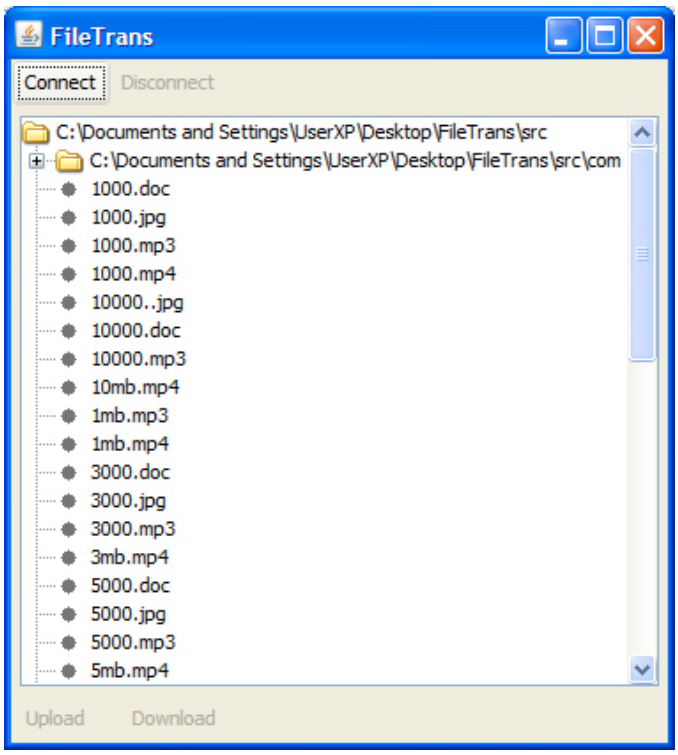
3.2 Perancangan Perangkat Lunak

Perangkat lunak yang dibuat menggunakan bahasa pemrograman java. Perangkat lunak tersebut dapat berjalan di berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat java runtime environment.

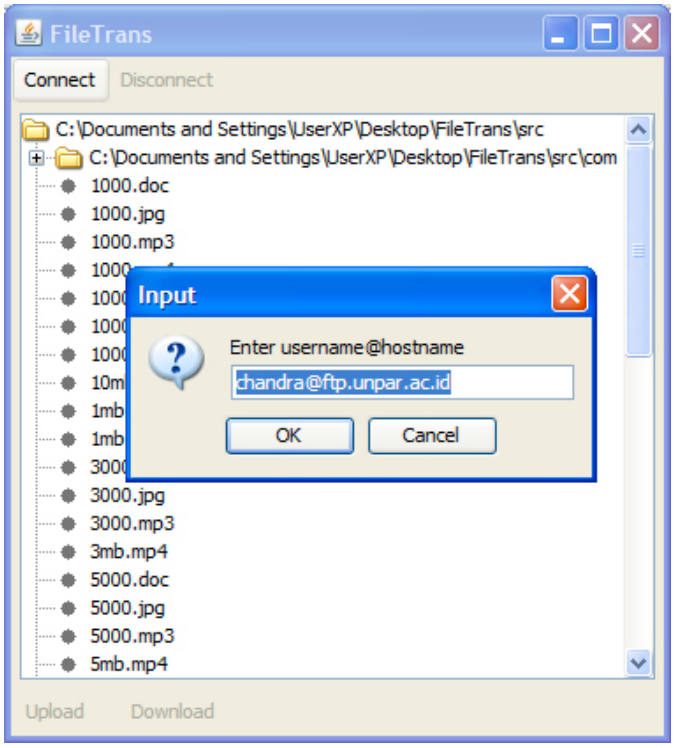
Perangkat lunak yang digunakan akan terdiri dari 2 jenis, yaitu client dan server. Perangkat lunak client akan dijalankan oleh user yang akan melakukan pemindahan file, sedangkan untuk server akan menggunakan openSSH yang sudah terdapat di sistem operasi UNIX.

3.3 Perancangan antar muka

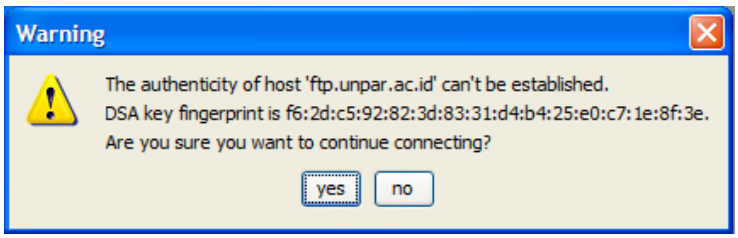
Berikut ini adalah antar muka dari perangkat lunak yang dibuat.



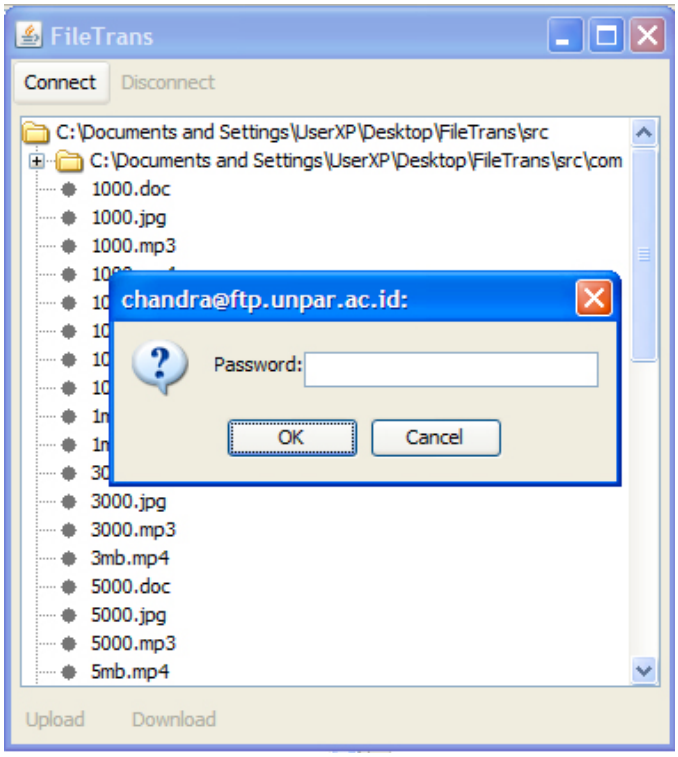
Gambar 3.1 Halaman awal dari perangkat lunak



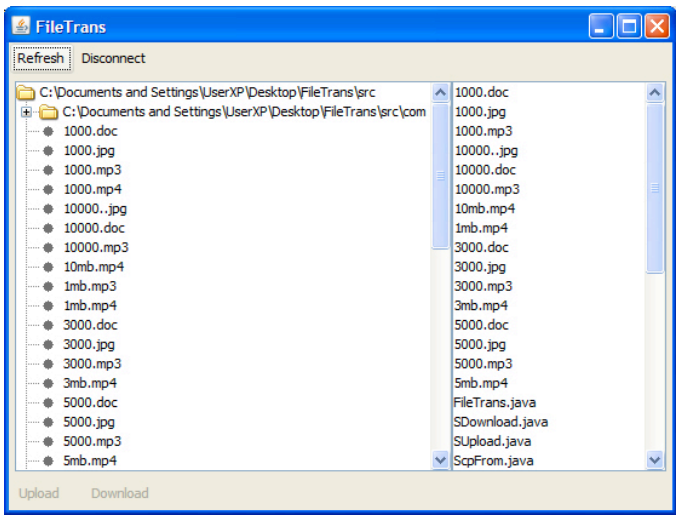
Gambar 3.2 User memasukan username dan alamat server yang dituju



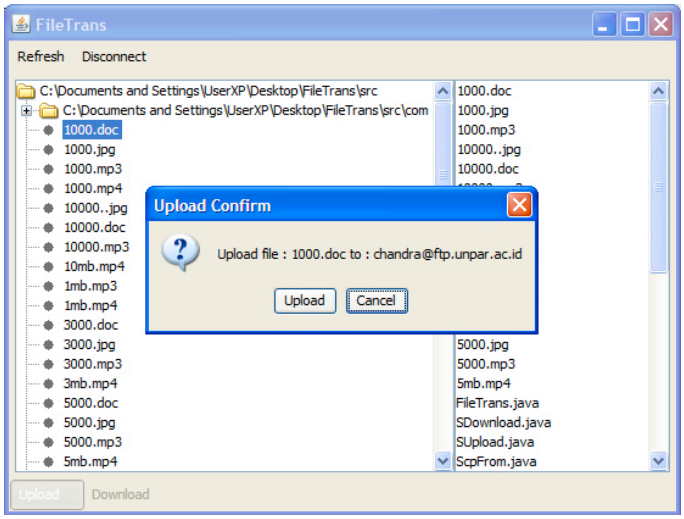
Gambar 3.3 Perangkat lunak mempertukarkan public key antara client dan server



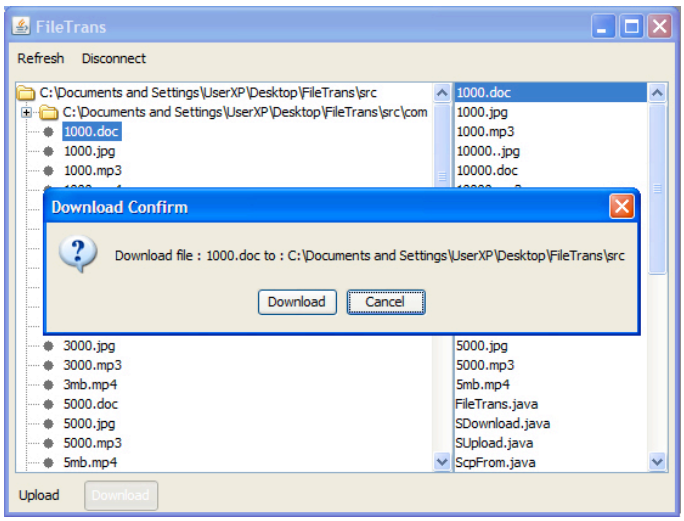
Gambar 3.4 User memasukan password



Gambar 3.5 Daftar file antara client dan server



Gambar 3.6 User melakukan upload file ke server



Gambar 3.7 User melakukan download file dari server

3.4 Hasil Pengujian

Berikut ini adalah hasil pengujian terhadap perangkat lunak yang telah dibuat. Satuan yang digunakan adalah ms (milisecond).

Tabel 3.1 Pengujian delay terhadap file bertipe .doc

DOC	1 MB	3 MB	5 MB	10 MB
Level 0	10233	24368	41672	86890
Level 1	10453	26506	41246	86687
Level 5	10211	26478	41234	86422
Level 9	9918	24212	40763	84346

Tabel 3.2 Pengujian delay terhadap file bertipe .jpg

JPG	1 MB	3 MB	5 MB	10 MB
Level 0	8968	26281	41672	91078
Level 1	8948	26265	41246	91068
Level 5	8906	26162	41234	90678
Level 9	8878	25987	40763	90578

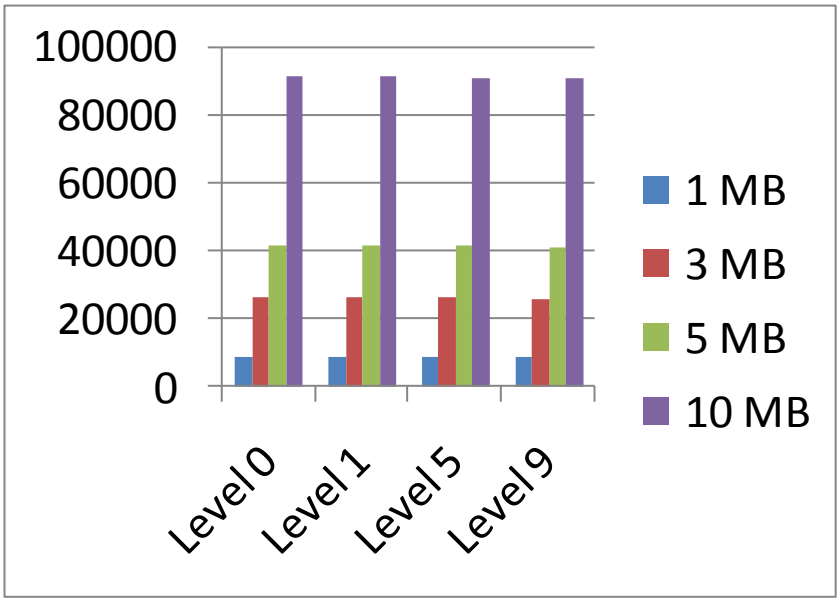
Tabel 3.3 Pengujian delay terhadap file bertipe .mp3

MP3	1 MB	3 MB	5 MB	10 MB
Level 0	8968	26281	41672	91235
Level 1	8948	26265	41246	91218
Level 5	8906	26162	41234	91068
Level 9	8878	25987	40763	90116

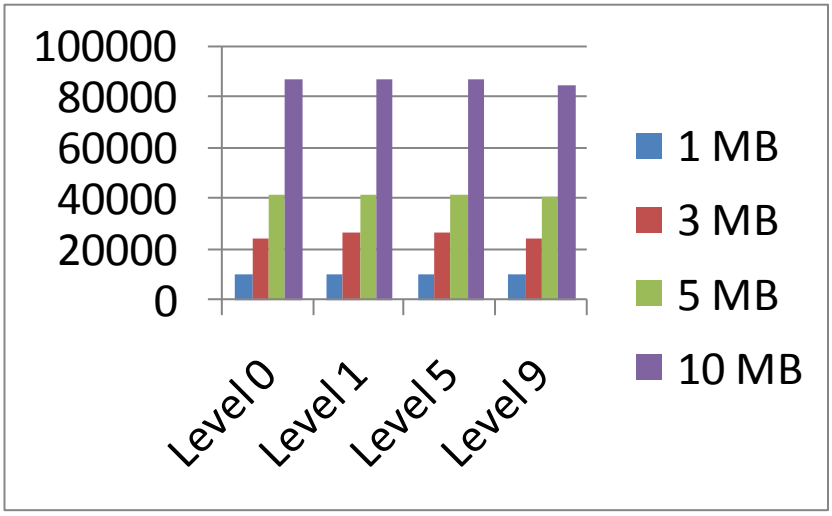
Tabel 3.4 Pengujian delay terhadap file bertipe .mp4

MP4	1 MB	3 MB	5 MB	10 MB
Level 0	8322	28990	42561	84837
Level 1	8406	29117	43226	84859
Level 5	8389	29250	43250	84844
Level 9	8348	28782	42869	83738

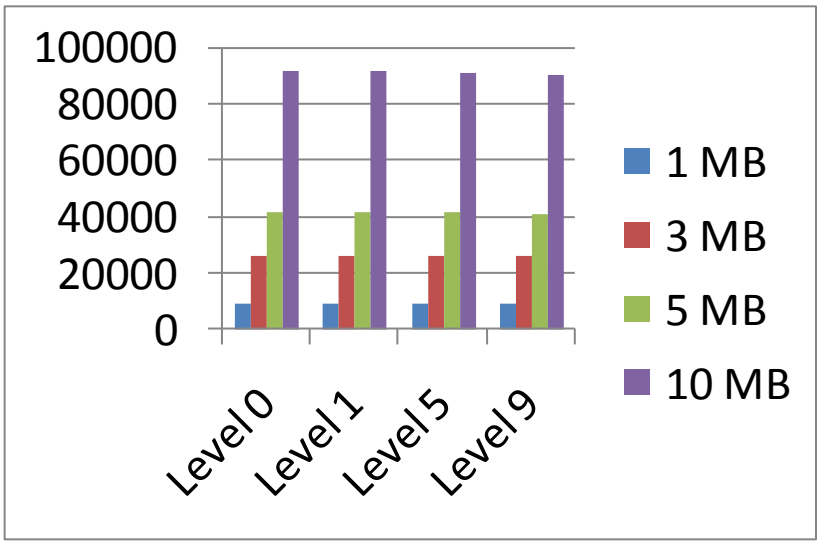
Berikut ini adalah gambar grafik dari pengujian file-file tersebut.



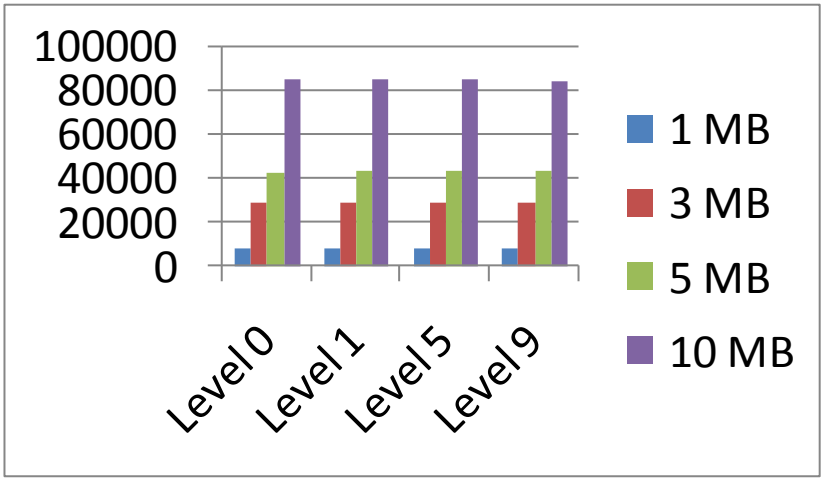
Gambar 3.8 Pengujian terhadap file bertipe .jpg



Gambar 3.9 Pengujian terhadap file bertipe .doc



Gambar 3.10 Pengujian terhadap file bertipe .mp3



Gambar 3.11 Pengujian terhadap file bertipe .mp4

BAB 4
KESIMPULAN DAN SARAN

Dari penelitian yang telah dilakukan, penulis akan menyimpulkan beberapa hal berikut:

1. Penggunaan SSH dapat meningkatkan keamanan data. Hal ini disebabkan karena data yang dikirimkan menggunakan SSH telah dienkripsi sehingga tidak dapat dibaca oleh sembarang pihak.
2. Penggunaan ZLIB membuat file menjadi berukuran lebih kecil saat akan dipertukarkan. Hal ini dapat dilihat dari hasil pengujian di bab 3, bahwa dengan menggunakan kompresi, delay menjadi lebih kecil dibandingkan tanpa menggunakan kompresi.
3. Ada tipe file tertentu yang dapat dikompres agar ukurannya dapat lebih kecil, dan ada tipe file tertentu yang tidak terlalu signifikan hasilnya saat dikompres. Hal ini disebabkan karena file tersebut sudah merupakan hasil kompresi dari program tertentu.

Saran penulis untuk penelitian selanjutnya adalah:

1. Tipe file yang diujicobakan lebih beragam.
2. Perangkat lunak diujicobakan pada jaringan yang lebih besar, sehingga lebih real hasilnya.

DAFTAR REFERENSI

[SAL-04] Salomon, David, 2004, "Data Compression The Complete Reference 3rd Edition", Springer.

[SAL-10] Salomon, David, 2010, "Handbook of Data Compression 5th Edition", Springer.

[NEL-05] Nelson, Mark, 2005, "The Data Compression Book 2nd Edition", IDG Books Worldwide Inc.

[URL-01] "The Secure Shell (SSH) Protocol Architecture" , http://tools.ietf.org/html/rfc4251

[URL-02] "The Secure Shell (SSH) Protocol Architecture", http://tools.ietf.org/html/rfc4251

[URL-03] "The Secure Shell (SSH) Authentication Protocol", http://tools.ietf.org/html/rfc4252