Jurnal Generic, Vol. 9, No. 2, September 2014, pp. 348~355
ISSN: 1907-4093 (print), 2087-9814 (online)

Implementasi FTP Server dengan Metode Transfer Layer Security untuk Keamanan Transfer Data Menggunakan CentOS 5.8

Ahmad Fali Oklilas[1], dan Budi Irawan[2]
[1]Jurusan Sistem Komputer Fakultas Ilmu Komputer, Universitas Sriwijaya,
[2]Program Studi Teknik Komputer Program Diploma Komputer, Universitas Sriwijaya
Jln. Raya Palembang Prabumulih Km.32 Inderalaya, Ogan Ilir, Indonesia
E-mail: [1]faliunsri@gmail.com, dan [2]budi_irawan6@yahoo.com

Abstrak

Dalam kehidupan sehari-hari pertukaran informasi sangatlah dibutuhkan, untuk mendapatkan informasi atau bertukar informasi dapat dilakukan kapan saja dan dimana saja. Informasi bisa berupa file atau data, jaringan komputer merupakan salah satu media untuk melakukan pertukaran sharing data. Sebagai contoh salah satu media yang dapat digunakan adalah FTP (File Transfer Protocol), dengan cara file yang di-upload oleh user tersimpan di hard disk server yang dapat diakses kapan saja selama user terhubung dengan jaringan internet. Semakin berkembangnya teknologi pada zaman modern sekarang file-file penting yang di-upload atau yang di-download sangatlah rentan dengan kejahatan dunia maya seperti digunakanya tool wireshark yang dapat mengetahui username, password, dan file yang di-upload atau di-download oleh user. Untuk itu dibangunlah FTP server dengan system keamanan dalam transfer data. FTP server yang telah diaktifkan fitur secure socket layer dan secure shell dapat mengamankan proses informasi username, password, dan file atau data yang di-upload dan di-download oleh user tidak dapat terbaca oleh tool wireshark. Untuk mencegah penuhnya kapasitas hard disk server, maka digunakanlah penerapan Disk Quota yang berfungsi memberi batasan kuota user dalam melakukan upload. Untuk pembuktian keamanan FTP server dapat dilakukan pengujian menggunakan metode transfer layer security (TLS). Dengan demikian FTP server yang belum menggunakan secure socket layer dan secure shell username, password dan file dapat terbaca oleh tool wireshark. Sedangkan FTP server yang telah menggunakan secure socket layer dan secure shell lebih aman karena username, password dan file tidak terbaca oleh tool wireshark.

Kata kunci: *Disk Quota, File Transfer Protocol, Secure Shell, Secure Socket Layer,Transfer Layer Security.*

Abstract

In daily life the exchange of information is needed to get the information or information exchange, it can be done anytime and anywhere. Information can be sharing the data. For example, one of media that can be used is a FTP (File Transfer Protocol), by the files that have been uploaded by the user will be stored in the server's hard disk which is can be accessed at any time during the user connects to the Internet network.The developing of technology in the modern era, the important files that are uploaded or downloaded are vulnerable to cybercrime like using wire shark tool to

know the username, password, and the files that are uploaded or downloaded by the user So, because of that the FTP server with the security system of the data transfer have been made. FTP server active by secure socket layer and shell user features that can secure the information process of username, password and files or data that have uploaded and downloaded by the user cannot be read by the wire shark tool. To prevent the full capacity of the hard disk server, so that's why it used Disk Quota application that serves to limit the user's quota while uploading. For the proven of FTP security server can be tested by using the method of transfer layer security (TLS). Thus, the FTP server that is not using secure socket layer and the secure shell username, password and the file can be read by wire shark tool. While the FTP server has been used secure socket layer and secure shellit is more secure because of the username, password, and the file cannot be read by wire shark tool.

Keywords: *Disk Quota, File Transfer Protocol, Secure Shell, Secure Socket Layer, Transfer Layer Security.*

1. Pendahuluan

File Transfer Protocol (FTP) sampai saat ini masih menjadi media favorit yang digunakan untuk melakukan *transfer file* melalui jaringan internet terutama *file-file* yang berukuran besar. Hal ini disebabkan media komunikasi seperti email memiliki keterbatasan untuk melewatkan ukuran *file* yang besar [1]. FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi. Pengguna terdaftar dapat menggunakan *username* dan *password*-nya untuk mengakses, men-*download*, dan meng-*upload* berkas-berkas yang ia kehendaki. Umumnya, para pengguna terdaftar memiliki akses penuh terhadap beberapa direktori, sehingga mereka dapat membuat berkas, membuat direktori, dan bahkan menghapus berkas. Pengguna yang belum terdaftar dapat juga menggunakan metode *anonymous login,* yakni menggunakan nama pengguna *anonymous* dan *password* yang diisi dengan alamat *e-mail*.

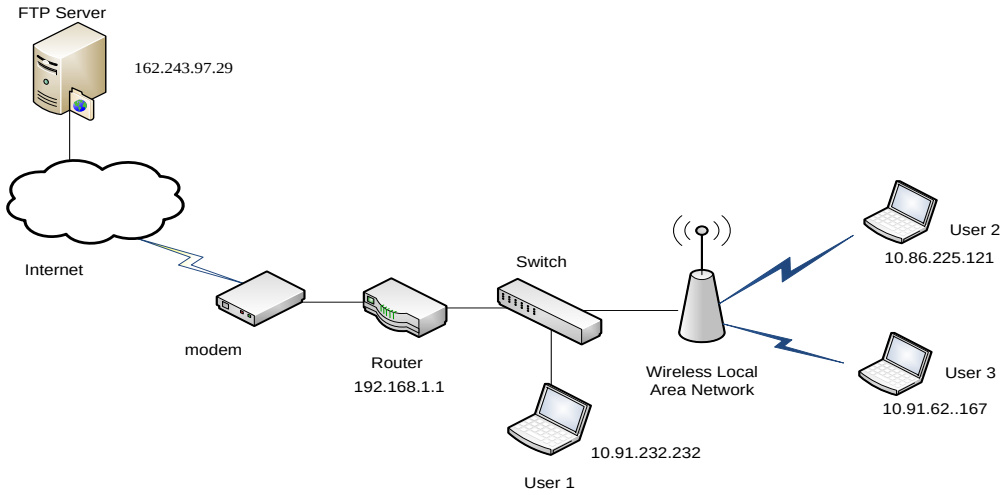
Oleh karena itu diperlukan penerapan *transfer* data dan autentikasi pada FTP *server* dengan menggunakan *secure socket layer* dan *secure shell*, sehingga proses autentikasi dan proses *transfer* data terlebih dahulu dienskripsi menjadi *ciphertext*. Apabila tidak ada batasan dalam melakukan *upload file* maka *user* yang berada dalam jaringan FTP *server* akan terus menyimpan data pada computer FTP *server* dan mengakibatkan penuhnya kapasitas *hard disk* yang ada. Untuk mengatasi setiap *user* dalam jaringan FTP server agar tidak menyimpan data melebihi kapasitas yang ada maka dipergunakanlah pembatasan *disk quota* pada setiap *user*, sehingga *user* pada jaringan FTP *server* tidak sembarangan menyimpan data atau *file* pada komputer *FTP server*.

Penelitian ini berdasarkan pengembangan dari penilitian yang telah dilakukan oleh Mohammad Martin Ruswanda yang berjudul implementasi FTP *server* dengan *secure socket layer* dan *secure shell* untuk keamanan transfer data [2] dan dengan sedikit pengembangan yang sebelumnya menggunakan Linux Ubuntu maka pada penilitan ini digunakan CentOS 5.8**.** Dimana dalam pengembangan ini telah menggunakan sistem

operasi CentOS, *tools pure-ftp*, penambahan *database* (*MySQL*) sebagai tempat penyimpanan data *client*, dan pengujian dilakukan dengan menggunakan metode *transfer layer security* (TLS).

2. Perancangan Sistem

2.1 PerancanganTopologi FTP Server



Gambar 1: Topologi FTP Server

Topologi yang digunakan dalam penelitian ini adalah topologi star, dimana FTP server yang terhubung ke internet, kemudian dapat diakses oleh *client* melalui modem kemudian terhubung ke *router* dengan IP 192.168.1.1. Selanjutnya dihubungkan ke *switch* dan ada LAN kabel ke *user* 1 dengan IP 10.91.232.232. Dari *switch* juga ada yang terhubung ke *wireless* LAN sehingga dapat melayani *user* 2 dengan IP 10.86.225.121, dan *user* 3 dengan IP 10.91.622.167. *Wireless* LAN dapat juga ditambah beberapa *user* yang lain begitu juga *switch* dapat ditambah Kabel LAN untuk beberapa *user* yang lain. Dapat dilihat secara detil pada Gambar 1.

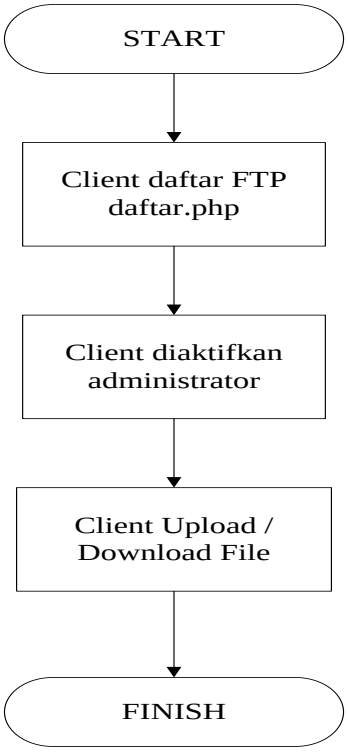
2.2 Tahap Pendaftaran dan Pembuatan FTP Server

Penelitian ini menggunakan system operasi *CentOS versi 5.8* pada VPS (*Virtual Private Server*).Tahap-tahap pendaftaran dan pembuatan FTP server sebagai berikut:

1. Daftar terlebih dahulu di *digitalocean.com;*
2. Pilih jenis OS yang akan digunakan yaitu CentOS versi 5.8;
3. Setelah itu didapatkan sebuah informasi berupa *username*, *password*, IP *address* VPS;
4. Login server VPS menggunakan informasi yang telah didapat menggunakan *putty*;
5. Terakhir install paket-paket FTP server.

2.3 Sistem Manajemen FTP Server

Setiap pengguna melakukan proses *upload* (unggah) dan *download* (unduh) menggunakan FTP server diharuskan daftar terlebih dahulu dan akan diaktifkan oleh administrator, dengan cara masuk ke *browser* dan menggunakan IP akses yakni 162.243.97.29/budi.



Gambar 2: Diagram alir proses kerja FTP server

Pada Gambar 2 merupakan diagram alir dari proses kerja FTP server, pada proses awal dimulai dengan *star*t, dimana *client* mendaftar ke FTP *server* melalui akses web di daftar.php. Setelah proses pendaftaran maka harus ada aktivasi dari administrator sehingga *client* diperbolehkan untuk akses selanjutnya. Setelah *client* mendapat hak akses maka dia boleh melakukan *upload* dan atau *download file* di FTP *server*. Proses ini dapat berlanjut ke *client* lain atau bisa selesai.

3. Hasil dan Pembahasan

Dalam membangun FTP server ini dilakukan beberapa tahap pengujian seperti keamanan authentikasi, keamanan transfer data, dan pengujian *disk quota*.

3.1 Pengujian Keamanan Authentikasi FTP Server

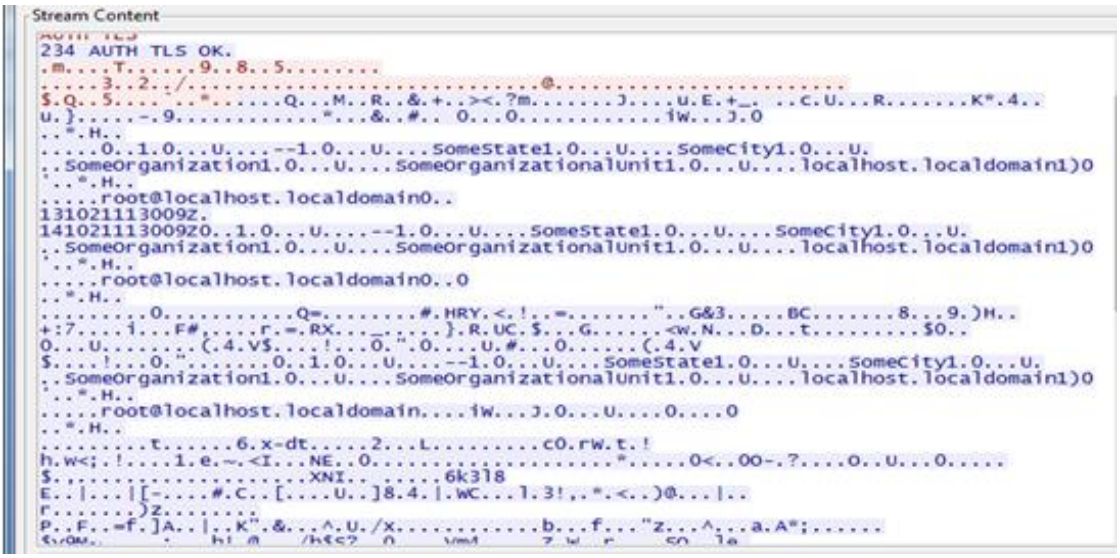
Pengujian keamanan authentikasi FTP server dilakukan dengan beberapa tahap seperti berikut:

1. Tujuan pengujian : Mengamankan proses authentikasi FTP *server*.
2. Yang akan diuji : Proses authentikasi FTP *server* menggunakan protokol TLS.
3. Parameter Keberhasilan:
   a) Berhasil apabila *username* dan *password* yang dikirim tidak dapat dibaca oleh *tool wire shark*;
   b) Gagal apabila *username* dan *password* yang dikirim dapat dibaca oleh *tool wire shark*.

FTP *user* melakukan proses *login* ke FTP *server* menggunakan *username* irawan dan *password* irawan. Jika *user* berhasil maka dapat dilihat pada *tool wire shark* dimana *username* dan *password* tersebut berhasil *login* terlihat pada Gambar 3.

Gambar 3:*Username* dan *password* berhasil dibaca oleh *tool wire shark*

Ketika FTP *user* dengan *username* irawan dan *password* irawan melakukan *login* menggunakan protokol TLS maka akan tampil seperti Gambar 4 seperti berikut.

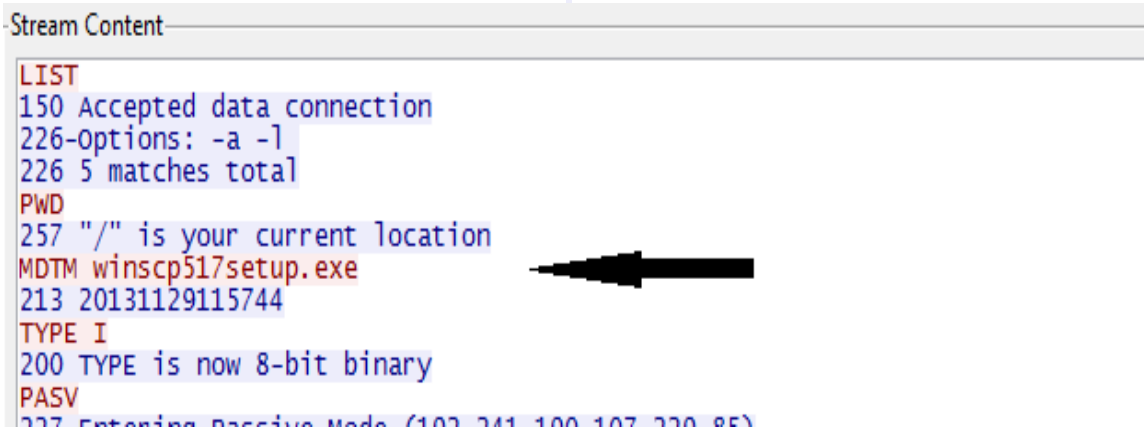


Gambar 4: *Username* dan *password* tidak terbaca oleh *tool wire shark*

Gambar 4 menjelaskan bahwa ketika FTP *user* melakukan *login* menggunakan *protocol* TLS, maka paket data yang dikirim berupa *username* irawan dan *password* irawan tidak dapat dibaca oleh *tool wire shark.*

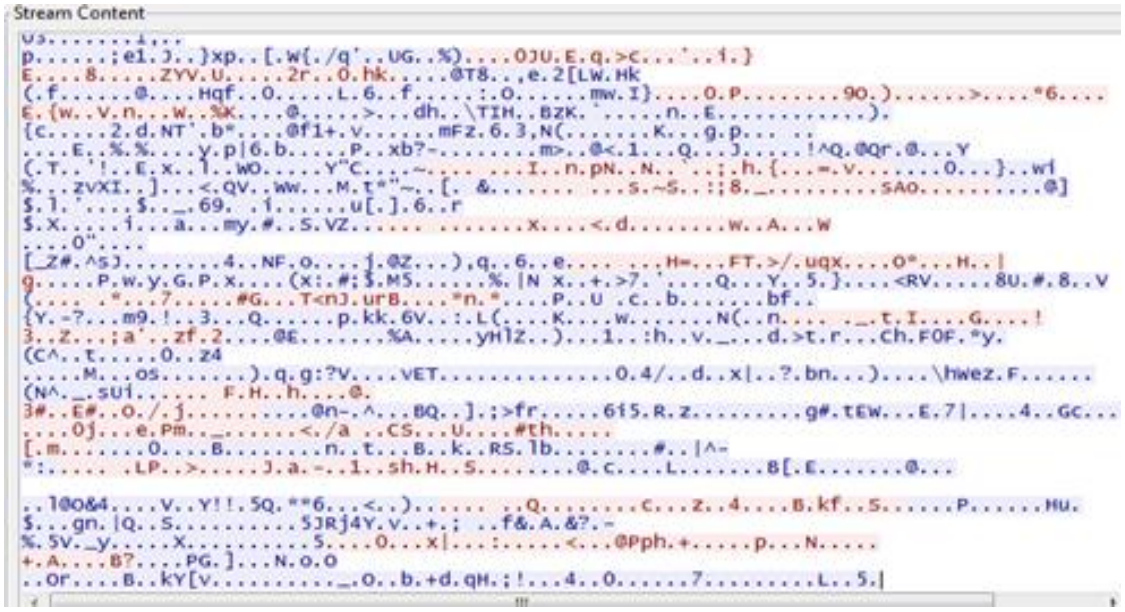
3.1.1 Pengujian Keamanan Transfer Data FTP server

Hasil pengujian keamanan transfer data pada FTP *server* menggunakan *file* winscp517setup.exe diperlihatkan Gambar 5.



Gambar 5: Isi *file* yang berhasil dibaca oleh tool wireshark

Gambar 5 menjelaskan bahwa *file* winscp517setup.exe yang ditransfer ke FTP server dapat dibaca oleh *tool wire shark.*



Gambar 6: Isi file tidak dapat dibaca oleh *tool wire shark*

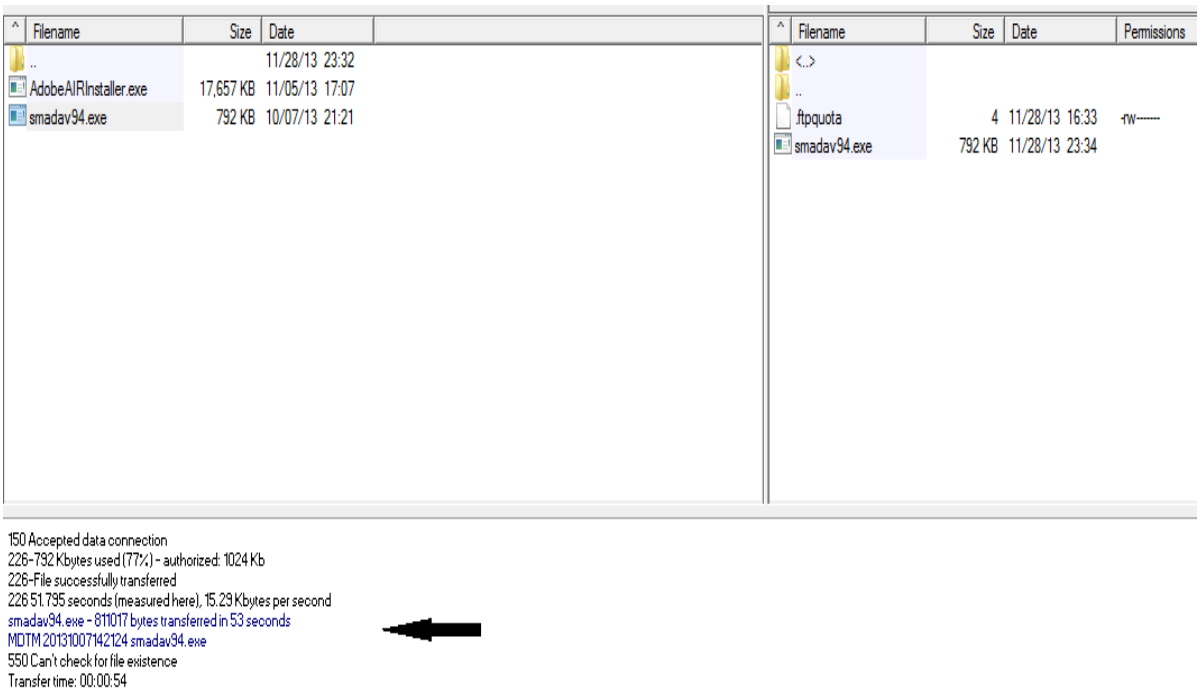
Gambar 6 menjelaskan bahwa *file* winscp517setup.exe yang ditransfer ke FTP *server* tidak dapat dibaca oleh *tool wire shark*. Hal ini dikarenakan sudah menggunakan *protocol* TLS dalam autentikasi dan transfer data antara FTP *server* menggunakan *secure socket layer* dan *secure shell*. Ini berarti sudah berhasil melalukan pengamanan akses pada FTP *server*.

3.1.2 Pengujian Disk Quota

Dalam pengujian Disk Quota pada FTP *server* dilakukan dengan beberapa tahap seperti berikut.

1. Tujuan pengujian : Membatasi *quota user* untuk melakukan unggah agar tidak melebihi batas kapasitas hard disk.
2. Yang akan diuji : Proses unggah data *user* ke FTP *server*.
3. Parameter Keberhasilan :
   a) Berhasil apabila *user* FTP gagal untuk melakukan unggah *file* atau data melebihi batas quota yang telah diberikan.
   b) Gagal apabila *user* FTP tetap dapat melakukan unggah meskipun melewati batas quota yang telah diberikan.

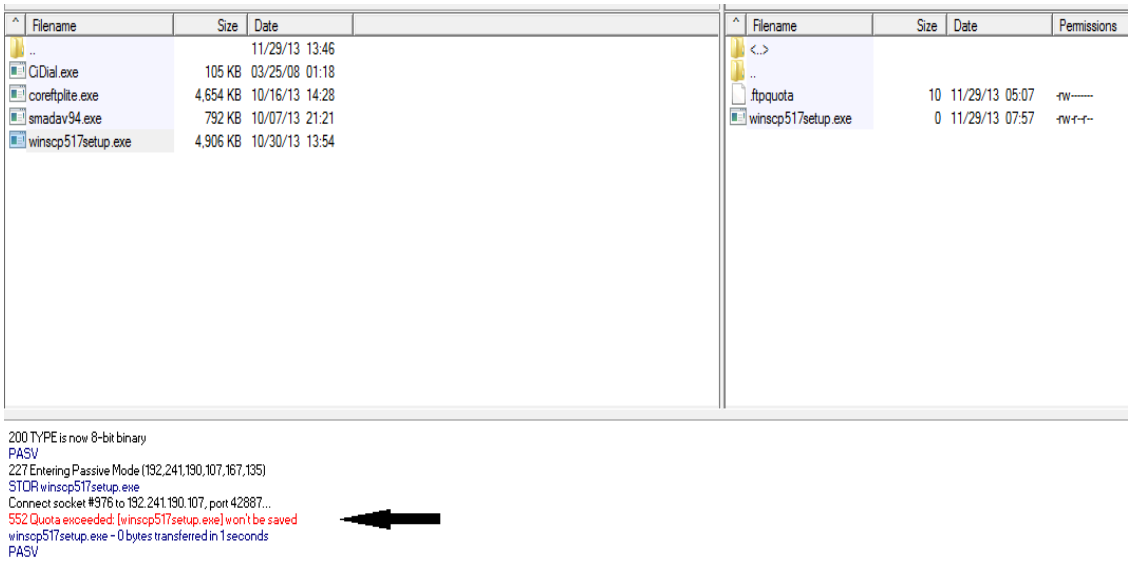
Admin membatasi quota untuk *user* irawan sebesar satu megabyte.



Gambar 7: User FTP berhasil melakukan *upload file*

Gambar 7 menjelaskan bahwa *user* irawan berhasil melakukan unggah *file* sebesar 729 KB yang berarti masih di bawah batas quota yang telah diberikan oleh admin.

Ketika user irawan melakukan unggah *file* sebesar 4906 KB berarti telah melewati batas kuota yang telah diberikan maka proses unggah file akan gagal seperti pada Gambar 8 berikut.



Gambar 8: Proses unggah gagal

5. Kesimpulan

Hasil penelitian implementasi FTP *server* dengan metode *Transfer Layer Security* (TLS) untuk keamanan transfer data menggunakan CentOS 5.8 dapat dismpulkan sebagai berikut.

1. FTP *server* tidak menggunakan *secure socket layer* dan *secure shell* dapat terbaca oleh *tool wireshark* nama *user*, *password*, maupun *file* yang di-unggah atau di-unduh oleh *user* FTP. Sebaliknya, user FTP tidak dapat terbaca oleh *tool wireshark*.
2. FTP *server* yang telah diterapkan *disk quota* berfungsi sebagai pembatasan penggunaan kapasitas *hard disk* untuk mencegah penuhnya *hard disk* pada *server*.

Referensi

[1] Athailah, "Ubuntu Server**"**, Batam: Jasakom, 2013.

[2] M. M. Ruswanda, P. D. Ibnu Graha, & T. Zani, "Implementasi FTP Server dengan Secure Sockets Layer dan Secure Shell untuk Keamanan Transfer Data**"**, Politeknik Telkom Bandung, 2011.

[3] Y. Hariyanto, "Sistem Operasi Linux Centos**",** STMIK Teknokrat Bandar Lampung, 2010.

[4] I. Cartealy, "Linux Networking**"**, Batam: Jasakom, 2013.

[5] A. Kadir, "Dasar Pemrograman Web Dinamis Menggunakan PHP", Yogyakarta: CV. Andi Offset, 2008.