

File Transfer Protocol

FAJAR YUSRAN ZEBUA

fajarzebua@yahoo.com

fajarzebua@gmail.com

Lisensi Dokumen:

Copyright © 2003-2006 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pengertian FTP

File Transfer Protocol (FTP) adalah suatu protokol yang berfungsi untuk tukar-menukar file dalam suatu *network* yang menggunakan TCP koneksi bukan UDP. Dua hal yang penting dalam FTP adalah FTP Server dan FTP Client.

FTP server adalah suatu server yang menjalankan software yang berfungsi untuk memberikan layanan tukar menukar file dimana server tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan (*request*) dari FTP client.

FTP client adalah computer yang merequest koneksi ke FTP server untuk tujuan tukar menukar file. Setelah terhubung dengan FTP server, maka client dapat men-*download*, meng-*upload*, me-*rename*, men-*delete*, dll sesuai dengan *permission* yang diberikan oleh FTP server.

Tujuan dari FTP server adalah sebagai berikut :

- Untuk tujuan *sharing* data
- Untuk menyediakan *indirect* atau *implicit remote computer*
- Untuk menyediakan tempat penyimpanan bagi user
- Untuk menyediakan transfer data yang reliable dan efisien

Keamanan FTP

FTP sebenarnya cara yang tidak aman dalam mentransfer suatu file karena file dikirimkan tanpa di-enkripsi terlebih dahulu tetapi melalui *clear text*. Mode text yang dipakai untuk transfer data adalah format ASCII atau format *binary*. Secara default, FTP menggunakan mode ASCII dalam transfer data. Karena pengirimannya tanpa enkripsi, *username*, *password*, data yang di transfer, maupun perintah yang dikirim dapat di *sniffing* oleh orang dengan menggunakan *protocol analyzer* (*sniffer*). Solusi yang digunakan adalah dengan menggunakan SFTP (SSH FTP) yaitu

FTP yang berbasis pada SSH atau menggunakan FTPS (FTP over SSL) sehingga data yang dikirim terlebih dahulu di enkripsi.

Mode Dalam FTP

FTP biasanya menggunakan dua buah port untuk koneksi yaitu port 20 dan port 21 dan berjalan *exclusively* melalui TCP bukan UDP. FTP server mendengar pada port 21 untuk *incoming connection* dari FTP client. Biasanya port 21 adalah *command port* dan port 20 adalah *data port*.

Pada FTP server, terdapat 2 mode koneksi yaitu aktif mode (*active mode*) dan pasif mode (*passive mode*).

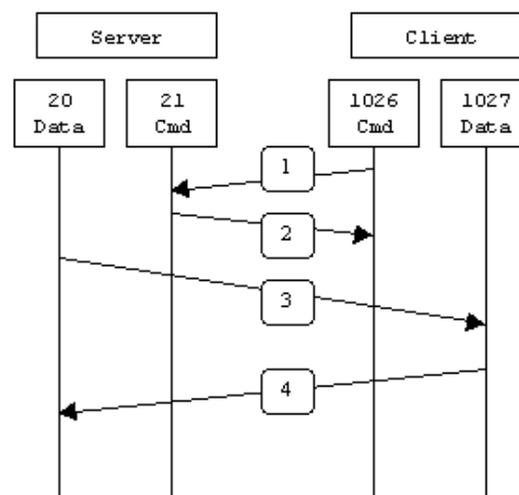
1. Active Mode

Pada aktif mode ini, server secara aktif terhubung dengan client. Untuk melakukan pengaturan aktif mode, client mengirimkan sebuah port command ke server, menentukan alamat dan nomor port dari client yang sedang mendengar. Bila satu koneksi diperlukan, server memulai suatu koneksi ke client di alamat ini. Secara umum, server bertanggung jawab untuk menutup koneksi-koneksi ini.

Untuk mendukung aktif mode, kanal komunikasi berikut perlu dibuka :

- Port server FTP 21 dimana pun (client memulai koneksi)
- Port server FTP 21 ke port >1023 (server memberikan respon ke port kontrol client)
- Port server FTP 20 ke port >1023 (server memulai koneksi data ke port data client)
- Port server FTP 20 dari port >1023 (client mengirimkan ACK ke port data server)

Prosesnya dapat dilihat pada gambar berikut :



Source : <http://slacksite.com>

Pada aktif mode, client terhubung dari suatu port yang acak (port N dimana $N > 1023$) ke FTP *command port* (port 21). Kemudian client mulai mendengarkan port $N+1$ dan mengirim FTP *command port* $N+1$ ke FTP server. Selanjutnya server akan terkoneksi ke client melalui data port (port 20).

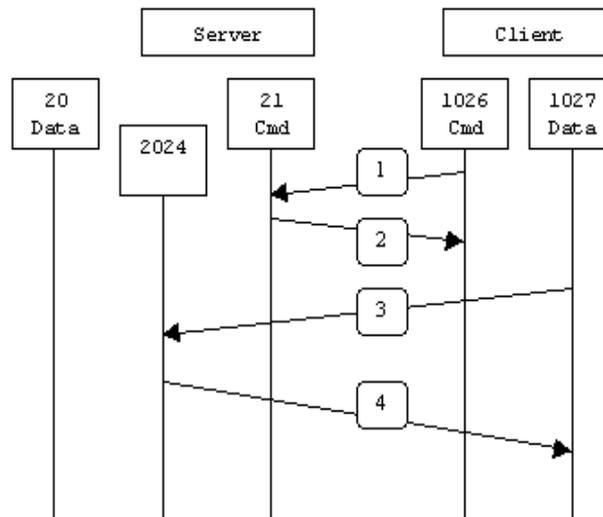
2. Passive Mode

Pada pasif mode, client memulai koneksi dengan server; memecahkan permasalahan dalam firewall-firewall penyaring koneksi port data ke client menuju server. Ketika koneksi FTP dibuka, client membuka dua buah port acak (port N dan N+1 dimana $N > 1023$). Port N akan kontak dengan port 21. Server akan membuka port acak (port P dimana $P > 1023$) dan kemudian mengirim command port P ke client. Kemudian client membuat koneksi dari port N+1 ke port P.

Untuk mendukung pasif mode, kanal komunikasi berikut perlu dibuka :

- Port server FTP 21 dimanapun (client memulai koneksi)
- Port server FTP 21 ke port > 1023 (server bereaksi terhadap port control client)
- Port-port server FTP > 1023 dari mana saja (Client memulai koneksi data ke port acak yang telah ditentukan oleh server)
- Port-port server FTP > 1023 ke port-port remote (Server mengirimkan ACK (dan data) ke port data client)

Prosesnya dapat dilihat pada gambar berikut :



Source : <http://slacksite.com>

Tahap pertama, client menghubungi server pada command port dan mengeluarkan perintah PASV. Server kemudian memberikan jawaban pada tahap kedua dengan menggunakan port 2024, memberitahu client bahwa port tersebut sedang mendengarkan untuk digunakan pada koneksi data. Tahap ketiga, client memulai koneksi data dari data port-nya ke data port yang telah ditentukan oleh server.

Perintah FTP

Perintah-perintah FTP yang dikirimkan terdiri atas string teks sederhana (mengikuti protokol Telnet). Sebagai contoh, untuk mendapatkan kembali suatu file, client mengirimkan "RETR filename" di koneksi kendali ke FTP server. Untuk mentransfer suatu file, client akan mengirimkan "STOR filename".

FTP server mengetahui setiap perintah pada setiap balasan FTP, dimana terdiri atas tiga digit. Digit pertama menandai adanya tanggapan yang baik, tidak baik, atau yang tidak sempurna. Jika satu kesalahan terjadi, maka digit kedua bisa digunakan untuk mendeteksi kesalahan yang terjadi. Dengan cara yang sama, digit ketiga dapat digunakan untuk lebih menspesifikasikan kesalahan yang terjadi.

Digit pertama merupakan digit yang paling utama, dan kemungkinan nilai yang muncul adalah sebagai berikut :

1yz	<i>Positive Preliminary reply.</i> Permintaan telah diketahui, namun balasan yang lain dari client tetap diharapkan
2yz	<i>Positive Completion reply.</i> Permintaan telah berhasil dilakukan, sehingga client dapat mengirimkan permintaan yang lain
3yz	<i>Positive Intermediate reply.</i> Perintah telah diterima, namun masih membutuhkan informasi yang lain. Client diharuskan untuk mengirimkan balasan perintah lainnya
4yz	<i>Transient Negative reply.</i> Perintah gagal, namun masih ada kesempatan untuk mencobanya lagi
5yz	<i>Permanent Negative Completion reply.</i> Perintah gagal, namun tidak harus diulangi lagi

Contoh perintah FTP :

Untuk mengubah direktori yang dikirimkan oleh client :

```
CWD dirname
```

Server akan merespon dengan :

```
250 CWD command successful
```

Sebagai balasan, dimulai dengan a '2', dimana urutan perintah diselesaikan.

Bila ingin merubah salah satu direktori dan itu tidak ada, maka perintahnya :

```
CWD nonexistentdir
```

Server akan merespon dengan :

```
550 nonexistentdir : The system cannot find the file spesified
```

Sebagai balasan adalah a '5', dimana perintah gagal, dan pasti akan gagal jika diulangi lagi (kecuali jika direktori yang hilang diciptakan di server).

Sesi Perintah (*Session Commands*)

Untuk memulai sesi perintah FTP, maka perintah dari USER yang dikirimkan ke server adalah :

```
USER javaftp
```

Server akan memberikan balasan :

```
331 Password required for javaftp
```

Client harus memberikan balasan dengan mengisi password :

```
PASS mypassword
```

Server akan memberikan balasan :

```
230 User javaftp logged in
```

Setelah login, maka pengguna dapat menggunakan perintah-perintah yang berhubungan dengan direktori dan file.

Untuk mengakhirinya, client mengirimkan perintah sebagai berikut :

QUIT

Server akan memberikan balasan :

221

Sesi akhirnya ditutup, sehingga perintah apapun yang dikirimkan sudah tidak diterima lagi.

Referensi :

[1] Wikipedia : <http://en.wikipedia.org>

[2] IETF : <http://ietf.org>

[3] Slacksite : <http://slacksite.com>

Biografi Penulis



Fajar Yusran Zebua. Dilahirkan di Nias pada tanggal 25 Juni 1984. Lulus dari SMU Negeri 1 Gunungsitoli, Kabupaten NIAS pada tahun 2002. Mengambil jurusan Teknik Informatika di Institut Sains dan Teknologi AKPRIND (ISTA) Yogyakarta, dan menyelesaikan studinya pada bulan Nopember 2006. Lulus dengan predikat *Cum Laude* dan menjadi mahasiswa terbaik mendorong penulis untuk mencari beasiswa melanjutkan studi. Akhirnya, pada awal bulan September 2007 penulis mendapatkannya, sehingga dapat melanjutkan studi S-2 di Universitas Gajah Mada (UGM) dengan program Magister Ilmu Komputer sampai saat ini. Dalam kesehariannya, penulis menambah ilmu pengetahuannya dengan mendalami beberapa database, jaringan komputer (khususnya *wireless connection*) dan juga beberapa bahasa pemrograman seperti Java, .NET *Technology*, PHP-Ajax.

Informasi lebih lanjut mengenai penulis ini bisa didapat melalui :

Blog :

<http://fajarz.wordpress.com> (*Knowledge Blog*)

<http://gudeg.wordpress.com> (*Lifestyle Blog*)

E-mail :

fajarzebua@yahoo.com

fajarzebua@gmail.com