

KOM334: KOMUNIKASI DATA & JARINGAN KOMPUTER

Materi Kuliah

Aplikasi-aplikasi di Internet

DEPARTEMEN ILMU KOMPUTER FMIPA IPB

Aplikasi-aplikasi Internet

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang)

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ

(Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan, Surat Al-‘Alaq: ayat 1, Makkiyyah)

Revisi: 1.0 (September 2005)

© Heru Sukoco 2005

Departemen Ilmu Komputer FMIPA IPB

Jl. Raya Pajajaran Baranangsiang III

Bogor – Jawa Barat 16144

Telp. +62-251-356653; 385961

Fax. +62-251-356653

Departemen Ilmu Komputer FMIPA IPB

Jl. Raya Darmaga Kampus IPB Darmaga

Bogor – Jawa Barat 16680

Telp./Fax. +62-251-625584

Kantor Pengembangan Sistem Informasi (KPSI) IPB

Jl. Raya Darmaga Gedung Rektorat Lt. 2 Kampus IPB Darmaga

Bogor – Jawa Barat 16680

Telp. +62-251-623936; Fax. +62-251-623936

Email: heru_sukoco@ilkom.fmipa.ipb.ac.id, hsrkom@ipb.ac.id, hsrkom@yahoo.com

URL: <http://www.ipb.ac.id/hsrkom> atau <http://www.ilkom.fmipa.ipb.ac.id/~hsrkom>

Daftar Isi

Aplikasi-aplikasi Internet	1
Tujuan Instruksional Khusus.....	1
Deskripsi Singkat	1
7. 1. Web Server (HTTP, <i>HyperText Transfer Protocol</i>)	1
7. 2. FTP (<i>File Transfer Protocol</i>) Server	2
7. 3. Email Server (SMTP, <i>Simple Mail Transfer Protocol</i> ; POP, <i>Post Office Protocol</i>)	5
7. 4. Remote Login / Terminal Access (TELNET).....	7

Daftar Gambar

7. 1. Aplikasi HTTP	1
7. 2. Aplikasi FTP	3
7. 3. Aplikasi Electronic Mail (Email)	5
7. 4. Aplikasi TELNET	7

Aplikasi-aplikasi Internet

Tujuan Instruksional Khusus

Mahasiswa diharapkan dapat:

- Mengetahui aplikasi-aplikasi apa saja yang populer digunakan di Internet
- memahami konsep dan implementasi aplikasi-aplikasi yang banyak digunakan di Internet

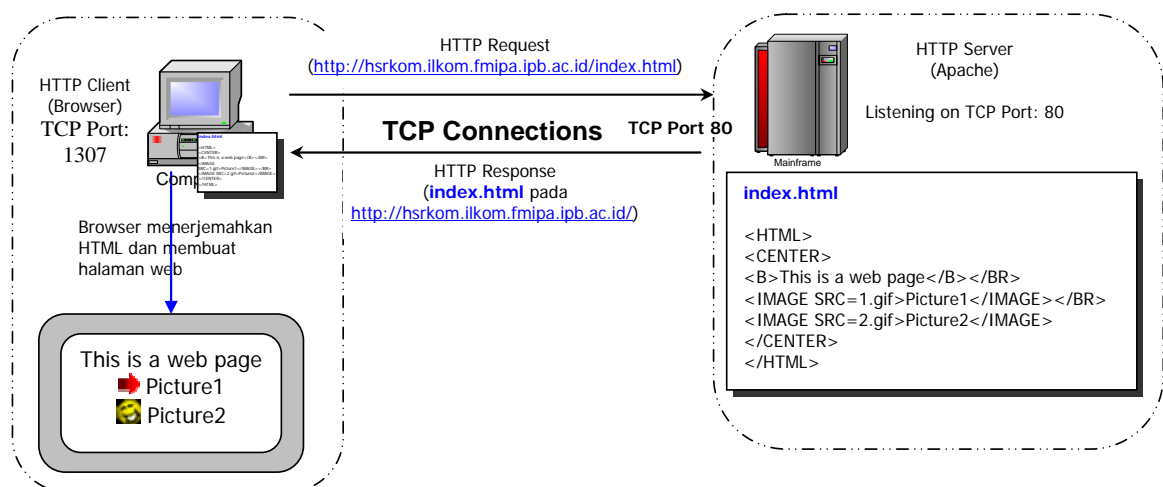
Deskripsi Singkat

- Aplikasi HTTP
- Aplikasi FTP
- Aplikasi Email dan Telnet

Pada bagian ini akan dibahas tentang perkembangan aplikasi-aplikasi di internet dan prinsip kerja masing-masing aplikasi berdasarkan protokol yang digunakannya.

7.1. Web Server (HTTP, *HyperText Transfer Protocol*)

Web server merupakan server yang menjadi penting seiring pesatnya perkembangan Internet karena berfungsi sebagai wadah penyimpanan dokumen-dokumen yang diakses melalui World Wide Web, yang merupakan aplikasi Internet paling sering digunakan. Komputer client mengakses dokumen yang tersimpan di web server melalui aplikasi web browser tanpa memperhatikan sistem operasi yang digunakan pada sisi server maupun client selama kedua belah pihak mengikuti protokol HTTP. Pada Gambar 7.1 dijelaskan secara singkat prinsip kerja aplikasi dan protokol HTTP.



Gambar 7.1. Aplikasi HTTP

Penjelasan proses pada Gambar 7.1 adalah sebagai berikut:

1. HTTP Client mengawali koneksi TCP ke HTTP Server (proses) pada `hsrkom.ilkom.fmipa.ipb.ac.id` di TCP Port = 80 (default port untuk HTTP Server).

HTTP Server pada host `hsrkom.ilkom.fmipa.ipb.ac.id` menunggu koneksi TCP di TCP Port = 80 (*Listening on TCP Port = 80*). Selanjutnya koneksi diterima (ACCEPT), kemudian HTTP Server memberikan peringatan ke HTTP Client, *TCP Connection terbentuk*.
2. HTTP Client mengirimkan *HTTP Request message* (berupa alamat URL) ke *TCP Connection Socket*.
3. HTTP Server menerima *HTTP Request message*, dan mengirimkan *HTTP Response message* yang terdiri dari halaman html yang di-request (referensi,obyek,dll) ke *TCP Connection Socket*, dalam hal ini `index.html` di server `hsrkom.ilkom.fmipa.ipb.ac.id`.
4. HTTP Server menutup koneksi TCP.
5. HTTP Client menerima HTTP Response message yang terdiri file html, kemudian HTTP Client (Browser) menerjemahkan tag-tag HTML dan menampilkannya. Parsing file HTML, cari referensi lain (yaitu 2 obyek gambar).
6. Ulangi langkah ke-1 s/d 5 untuk setiap n obyek jpg (dalam hal ini 2 obyek jpg).

7.2. FTP (*File Transfer Protocol*) Server

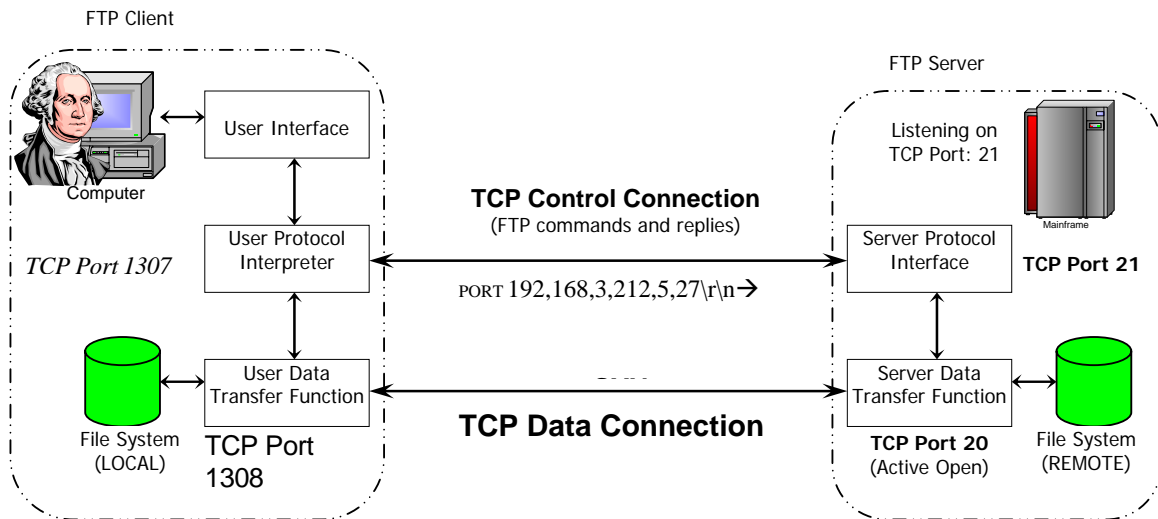
FTP server merupakan *server* yang juga erat kaitannya dengan Internet. Server ini menyediakan layanan *upload/download file* dari *client* melalui aplikasi *FTP Client* dengan protokol FTP (RFC 959).

FTP berbeda dengan aplikasi lainnya, karena memiliki 2 koneksi TCP untuk melakukan transfer sebuah file, yaitu:

1. **Control Connection**, yang digunakan pada koneksi antara klien-server yang normal. Server membuka diri secara pasif di sebuah port khusus (*well-known port*) yaitu TCP Port 21 dan selanjutnya menunggu koneksi yang akan dilakukan klien (*client connection*). Klien secara aktif membuka diri ke TCP port 21 untuk membangun *control connection*. *Control connection* ini akan dipertahankan sepanjang waktu selama klien masih berkomunikasi dengan server. Koneksi ini digunakan oleh klien untuk mengirim perintah-perintah ke server dan sebaliknya.

Koneksi ini bersifat “mengurangi delay”, karena perintah-perintah biasanya diketik oleh user (manusia) yang tentunya butuh kecepatan respon yang tinggi.

2. **Data Connection**, yang dibangun setiap kali sebuah file ditransfer antara klien-server. Koneksi ini bersifat “memaksimalkan throughput (ukuran data yang ditransfer)”, karena koneksi ini untuk transfer file.



Gambar 7.2. Aplikasi FTP

Penjelasan proses pada Gambar 7.2 adalah sebagai berikut:

1. Pengguna di FTP Client melakukan aktivitas FTP melalui *user interface* (misalnya: LeechFTP, SecureFX) dan tidak berhubungan langsung dengan *FTP commands* dan *FTP replies*.

User interface akan menjembatani pengguna berinteraksi dengan perintah-perintah FTP yang nantinya akan dikirimkan ke FTP Server melalui *TCP control connection*.

2. *User protocol interpreter* selanjutnya yang akan melakukan koneksi *TCP control connection* (*FTP commands*) ke FTP Server pada *TCP Port = 21*. Pada saat ini *TCP control connection* terbentuk dan akan dipertahankan sepanjang klien masih berkomunikasi dengan server. Perintah-perintah FTP standar akan dikirimkan oleh interpreter ini ke FTP Server melalui koneksi *TCP control connection*.

FTP Server (*server protocol interface*) selanjutnya mengirimkan *FTP replies* dari setiap perintah-perintah FTP Client melalui koneksi *TCP control connection*.

3. Selanjutnya, pengguna dapat melakukan perintah FTP lainnya (seperti pengiriman/pengambilan file/direktori).

Pada saat pengguna akan mengambil file dari FTP Server, *user protocol interpreter* akan menerjemahkan perintah tersebut dan mengirimkannya ke FTP Server berupa PORT $n1, n2, n3, n4, n5, n6$ (IP klien $n1.n2.n3.n4$ dan port: $n5 \times 256 + n6$) melalui TCP *control connection*. Selanjutnya klien secara pasif akan membuka port lainnya untuk koneksi TCP *data connection*.

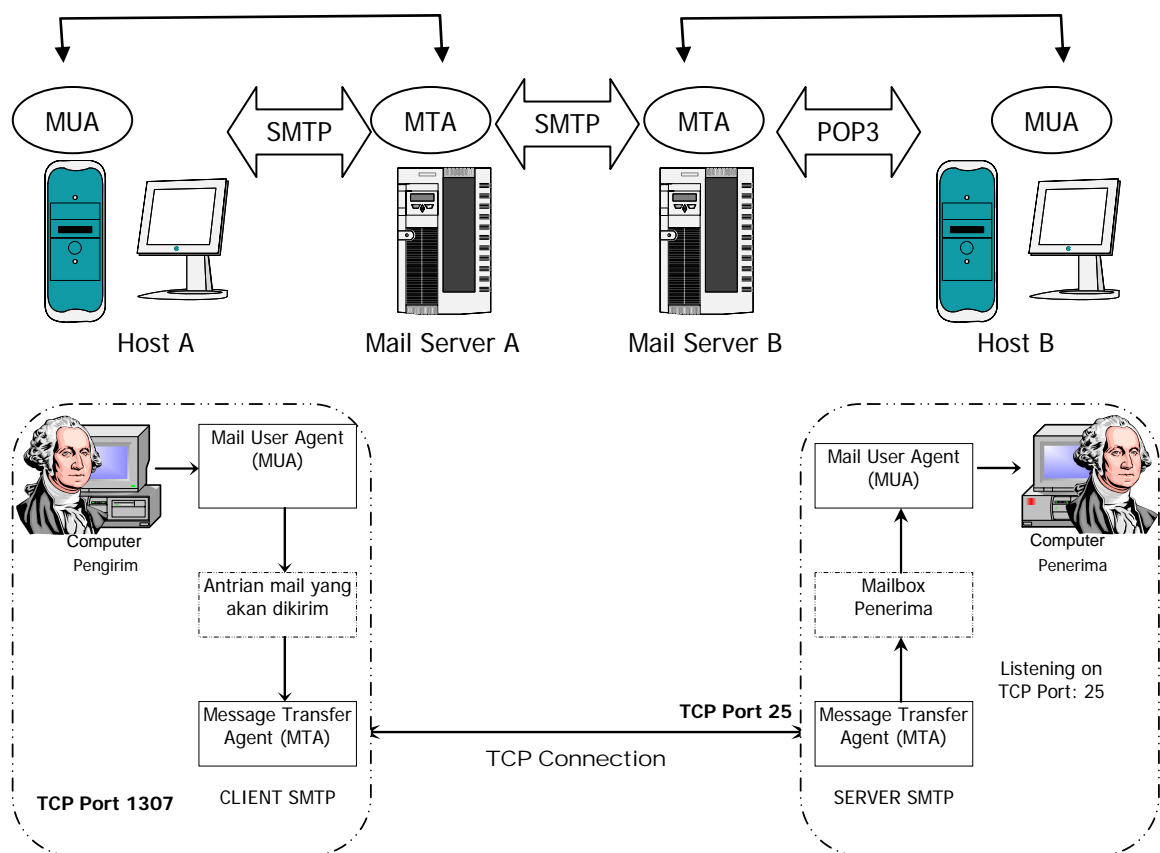
4. FTP Server menerima nomor port tersebut dari TCP *control connection* dan mengirimkan balasan secara aktif ke port FTP Client melalui TCP *data connection*. Nomor TCP Port untuk TCP *data connection* di sisi FTP Server = 20 sedangkan di sisi klien sembarang nomor port yang belum digunakan.
5. FTP Server menutup koneksi TCP *data connection*.
FTP Client menerima seluruh data secara lengkap.
6. Ulangi langkah ke-1 s/d 5 untuk setiap perintah FTP yang diberikan.
7. Selesai, pengguna memberikan perintah **QUIT**, lakukan langkah ke-1 dan 2, kemudian FTP Server akan memberikan balasan melalui TCP *control connection* sekaligus menutupnya.

Contoh FTP:

Aktivitas	Deskripsi
[hsrkom@ilkom/]ftp -d hsrkom.ilkom.fmipa.ipb.ac.id Connected to hsrkom.ilkom.fmipa.ipb.ac.id. 220-Serv-U FTP-Server v2.3 for WinSock ready... 220-Selamat Datang di Server FTP hsrkom 220-Anda dapat mengakses dan mengambil resource 220 pada server ini secara gratis. User (hsrkom.ilkom.fmipa.ipb.ac.id:(none)): hsrkom ---> USER hsrkom 331 User name OK, send password Password: ---> PASS ***** 230 User HSRKOM logged in ---> SYST 215 UNIX Type: L8 Remote system type is UNIX. Using ASCII mode to transfer files. ftp> dir Ckinfo.txt ---> PORT 192,168,3,212,4,179 200 PORT Command OK ---> LIST Ckinfo.txt 150 Opening data connection -rwxr-xr-x 1 User Group 3246 Jun 4 06:19 CKINFO.TXT 226 Transfer complete remote: Ckinfo.txt ftp: 67 bytes received in 0.05Seconds 1.34Kbytes/sec. ftp> quit ---> QUIT 221 Sampai Jumpa ... Merdeka!	-d option untuk output debug Klien secara aktif membuka TCP <i>control connection</i> (TCC) Server merespon: SIAP (melalui TCC) Klien menampilkan prompt LOGIN Klien mengirimkan nama pengguna (melalui TCC) Server meminta PASSWORD Pengguna memasukkan password Klien mengirimkan password (melalui TCC) Server mengenal pemakai. Status: OK Server memberitahu tipe sistemnya (melalui TCC) Pengguna mengetikkan perintah DIR Klien mengirimkan perintah PORT (melalui TCC) Server merespon: OKE (melalui TCC) Klien mengirim perintah LIST (melalui TCC) Server menjawab perintah melalui TCP <i>data connection</i> (TDC) Server menutup TDC Pengguna mengetikkan perintah QUIT Klien mengirimkan perintah QUIT (melalui TCC) Server merespons: BYE dan menutup TCC

7.3. Email Server (SMTP, *Simple Mail Transfer Protocol*; POP, *Post Office Protocol*)

Electronic mail atau lebih dikenal dengan *e-mail* merupakan aplikasi Internet yang paling banyak digunakan setelah *World Wide Web*. Pengiriman dan penerimaan *e-mail* ditangani oleh *mail server* atau *mail transport agent* (MTA). Selain *mail server* sebenarnya banyak aplikasi lain yang terlibat dalam mekanisme pengiriman *e-mail* yang secara ringkas terlihat dalam gambar berikut.



Gambar 7.3. Aplikasi Electronic Mail (Email)

Penjelasan proses pada Gambar 7.3 adalah sebagai berikut:

1. Pengguna berinteraksi dengan MUA (misalnya: pine, Elm, Eudora, dll) dan MUA mengawali koneksi TCP ke SMTP Server di TCP Port = 25 (default port untuk SMTP Server).

Email yang dikirim oleh MUA, sementara ditampung dalam antrian sebelum dikirim oleh MTA sambil menunggu waktu pengiriman.

2. MTA (misalnya: sendmail, qmail, dll) mengirimkan email ke MTA lain menggunakan SMTP (koneksi TCP). Komunikasi antara 2 MTA menggunakan NVT ASCII (7-bit). Pengguna biasanya tidak berhubungan dengan MTA, karena hal ini merupakan tanggung jawab administrator untuk mengatur MTA lokal.
3. SMTP Client mengirimkan *request message* ke SMTP Server ke TCP port = 25.
4. SMTP Server merespons dengan kode balasan numerik dan beberapa string yang dapat dibaca di SMTP Client.
5. SMTP Server menutup koneksi TCP.
6. Ulangi langkah ke-1 s/d 5 untuk setiap email.

Catatan:

Untuk kasus *Mail Access Protocol* (seperti: POP, IMAP, HTTP-Mail) hampir mirip dengan proses SMTP. Perbedaannya adalah SMTP sebagai protokol untuk *delivery/storage to receiver's server*, sedangkan *Mail Access Protocol* (POP,IMAP,HTTP-Mail) sebagai protokol untuk *retrieval from server*.

Contoh komunikasi SMTP:

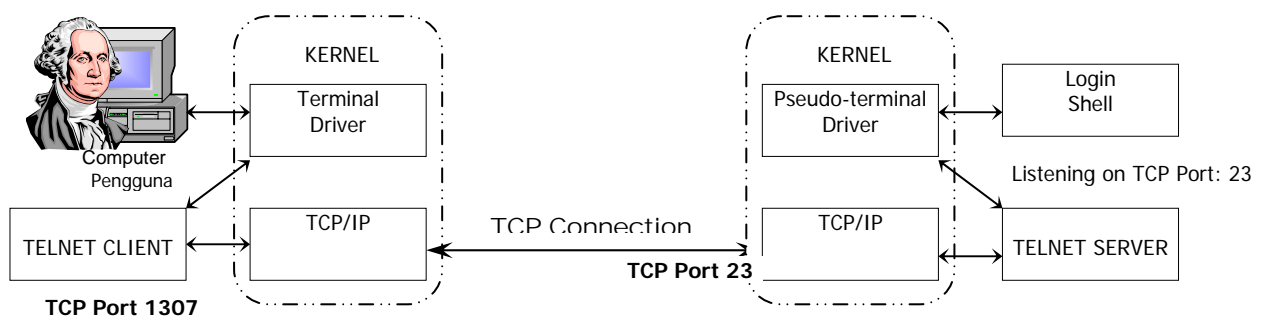
Aktivitas	Deskripsi
<pre>[hsrkom@ilkom/] telnet hsrkom.ilkom.fmipa.ipb.ac.id 25 Trying 192.168.3.212... Connected to hsrkom.ilkom.fmipa.ipb.ac.id Escape character is '^]' 220 hsrkom.ilkom.fmipa.ipb.ac.id qmail 1.3 SMTP server ready helo hsrkom 250 hsrkom.ilkom.fmipa.ipb.ac.id, ok! MAIL From:heru_sukoco@ilkom.fmipa.ipb.ac.id 250 Ok RCPT To:hsrkom@ilkom.fmipa.ipb.ac.id 250 Ok DATA 254 Enter your mail, end with . Subject: File Jarkom Hi, udah buka email belum? Aku udah kirim file2 jarkom ke kamu . 250 Sent Closing connection to ilkom.fmipa.ipb.ac.id. 221 Close</pre>	<p>Pengguna mengaktifkan telnet untuk mengakses ke SMTP Server Klien secara aktif membuka TCP <i>connection</i> (Port: 25) Server terkoneksi dan memberikan status: SIAP</p> <p>Status server dan versi SMTP yang digunakan Klien mengirimkan pesan pembukaan: HELO Server merespons: OK Pengguna mengetikkan alamat email pengirim Server merespons: OK Pengguna mengetikkan alamat email penerima Server merespons: OK Pengguna memulai email Server merespons: Masukkan Data dan Akhiri dengan “.” Pengguna mengetikkan Header Subject</p> <p>Pengguna mengetikkan isi surat (body)</p> <p>Akhiri isi surat</p> <p>Klien mengirimkan email ke alamat penerima Server menutup koneksi TCP</p>

7. 4. Remote Login / Terminal Access (TELNET)

Salah satu aplikasi *client-server* yang menarik adalah *remote login*. Dengan aplikasi ini seorang pengguna dapat login ke *server* dan melakukan operasi-operasi di *server* seakan-akan ia menggunakan *server* tersebut secara langsung. Operasi yang dilakukan mulai dari operasi *file* di *server* hingga mematikan *server*, tentunya tergantung pada hak akses yang dimiliki pengguna tersebut.

Remote login dapat dilakukan dengan beberapa jenis protokol, di antaranya yang populer adalah *telnet* dan *secure shell* (SSH). Keduanya menyediakan fungsi yang serupa namun SSH memiliki kelebihan dari segi keamanan dengan melakukan enkripsi terhadap komunikasi data yang terjadi antara *client* dengan

Contoh : telnet hsrkom.ikom.fmipa.ipb.ac.id (default=23)
(Alamat IP: 192.168.3.212)



Gambar 7.4. Aplikasi TELNET

Penjelasan proses pada Gambar 7.4 adalah sebagai berikut:

1. Telnet Client mengawali koneksi TCP ke Telnet Server di host hsrkom.ikom.fmipa.ipb.ac.id pada TCP Port = 23 (Default Port untuk Telnet Server).

Telnet Server pada host hsrkom.ikom.fmipa.ipb.ac.id menunggu koneksi TCP dari Telnet Client di TCP Port = 23 (*Listening on TCP Port = 23*). Selanjutnya koneksi diterima (ACCEPT), kemudian Telnet Server memberikan peringatan ke Telnet Client, *TCP connection* terbentuk.

2. Telnet Client mengirimkan *Request message* ke *TCP Connection Socket*. Telnet Client berinteraksi dengan pengguna di terminal dan protokol TCP/IP. Secara normal, apa pun yang diketikkan oleh pengguna dikirim ke Telnet Server melalui koneksi TCP, dan apa pun yang diterima dari koneksi TCP adalah output bagi terminal pengguna.
3. Telnet Server menerima Request message, dan mengirimkan Response message melalui koneksi TCP ke Telnet Client (Terminal/Monitor) sebagai outputnya. Koneksi TCP akan dipertahankan sepanjang klien masih berkomunikasi dengan server selama tidak melewati batas waktu *idle* yang diizinkan.

4. Ulangi langkah ke-1 s/d 3 untuk setiap perintah yang diketikkan.
5. Selesai, pengguna memberikan perintah **QUIT/EXIT**, dan kemudian Telnet Server akan memberikan balasan sekaligus menutup *TCP connection*.

Catatan:

1. Kotak terputus-putus sebenarnya untuk memberikan catatan bahwa *terminal driver* dan *pseudo-terminal driver* selama implementasi TCP/IP merupakan bagian dari kernel sistem operasi, sedangkan Telnet Client dan Server biasanya merupakan aplikasi pengguna (*user applications*).
2. Login shell pada host Server untuk menegaskan bahwa pengguna harus melakukan login ke server, sehingga seseorang harus memiliki *account* pada host Server jika akan melakukan Telnet.

Contoh Telnet:

Aktivitas	Deskripsi
<pre>[hsrkom@ilkom/] telnet hsrkom.ilkom.fmipa.ipb.ac.id Trying 192.168.3.212... Connected to hsrkom.ilkom.fmipa.ipb.ac.id Escape character is '^]' Linux Topimere 7.2 [kernel 2.2.14] at myServer Login: hsrkom Password: ***** [hsrkom@hsrkom/] _</pre>	<p>Pengguna mengaktifkan telnet Klien secara aktif membuka <i>TCP connection</i> (Port: 23) Server terkoneksi dan memberikan status: SIAP</p> <p>Klien menampilkan prompt LOGIN Server meminta PASSWORD dan engguna memasukkan password</p> <p>Server mengenal pengguna. Status: OK dan menerima perintah yang diketikkan pengguna.</p>

Referensi

- Comer, D. E. 2004. Computer Networks and Internet with Internet Applications. 4th Ed. Int'l Ed. Pearson Education, Inc. Pearson Prentice Hall. New Jersey.
- Garcia, L. and I. Widjaja, 2004. Communication Networks: Fundamental Concepts and Key Architectures. 2nd Ed. Int'l Ed. McGraw Hill, Inc.
- Halsall, F. 2002. Data Communications, Computer Networks and Open Systems. Addison Wesley.
- Stalling, W. 2004. Data and Computer Networks. 7th Ed. Int'l Ed. Pearson Education, Inc. Pearson Prentice Hall. New Jersey.