**PENGAMANAN DAN PENGENDALIAN SISTEM INFORMASI**

Dalam pelaksanaan suatu sistem informasi tentu tidak bisa terlepas dari keamanan data. Terlebih bagi organisasi besar yang tidak go public, sistem informasi yang dijalankan tentu sudah dibuat sedemikian rupa agar data-data perusahaan tidak dapat diakses oleh pihak yang tidak berkompeten.

Beberapa ancaman dan gangguan yang mungkin terjadi dan berpengaruh terhadap sistem informasi, adalah sebagai berikut:

1. Kerusakan perangkat keras.

2. Perangkat lunak tidak berfungsi.

3. Tindakan-tindakan personal.

4. Penetrasi akses ke terminal.

5. Pencurian data atau peralatan.

6. Kebakaran.

7. Permasalahan listrik.

8. Kesalahan-kesalahan pengguna.

9. Program berubah.

10. Permasalahan-permasalahan telekomunikasi.

**Tujuan Keamanan Sistem Informasi**

Keamanan sistem dimaksudkan untuk mencapai tiga tujuan utama yaitu;

kerahasiaan, ketersediaan dan integritas.

1. Kerahasian. Setiap organisasi berusaha melindungi data dan informasinya dari pengungkapan kepada pihak-pihak yang tidak berwenang. Sistem informasi yang perlu mendapatkan prioritas kerahasian yang tinggi mencakup; sistem informasi eksekutif, sistem informasi kepagawaian (SDM), sistem informasi keuangan, dan sistem informasi pemanfaatan sumberdaya alam.
2. Ketersediaan. Sistem dimaksudkan untuk selalu siap menyediakan data dan informasi bagi mereka yang berwenang untuk menggunakannya. Tujuan ini penting khususnya bagi sistem yang berorientasi informasi seperti SIM, DSS dan sistem pakar (ES).
3. Integritas. Semua sistem dan subsistem yang dibangun harus mampu memberikan gambaran yang lengkap dan akurat dari sistem fisik yang diwakilinya.

**Keamanan informasi**

Aspek **keamanan informasi** adalah aspek-aspek yang dilingkupi dan melingkupi keamanan informasi dalam sebuah sistem informasi. Aspek-aspek ini adalah :

* privasi/kerahasiaan, menjaga kerahasiaan informasi dari semua pihak, kecuali yang memiliki kewenangan;
* integritas, meyakinkan bahwa data tidak mengalami perubahan oleh yang tidak berhak atau oleh suatu hal lain yang tidak diketahui (misalnya buruknya transmisi data);
* otentikasi/[identifikasi](http://id.wikipedia.org/w/index.php?title=Identifikasi&action=edit&redlink=1), pengecekan terhadap identitas suatu entitas, bisa berupa orang, kartu kredit atau mesin;
* tanda tangan, mengesahkan suatu informasi menjadi satu kesatuan di bawah suatu otoritas;
* otorisasi, pemberian hak/kewenangan kepada entitas lain di dalam sistem;
* validasi, pengecekan keabsahan suatu otorisasi;
* kontrol akses, pembatasan akses terhadap entitas di dalam sistem;
* sertifikasi, pengesahan/pemberian kuasa suatu informasi kepada entitas yang terpercaya;
* pencatatan waktu, mencatat waktu pembuatan atau keberadaan suatu informasi di dalam sistem;
* persaksian, memverifikasi pembuatan dan keberadaan suatu informasi di dalam sistem bukan oleh pembuatnya
* tanda terima, pemberitahuan bahwa informasi telah diterima;
* konfirmasi, pemberitahuan bahwa suatu layanan informasi telah tersedia;
* kepemilikan, menyediakan suatu entitas dengan sah untuk menggunakan atau mengirimkan kepada pihak lain;
* anonimitas, menyamarkan identitas dari entitas terkait dalam suatu proses transaksi;
* nirpenyangkalan, mencegah penyangkalan dari suatu entitas atas kesepakatan atau perbuatan yang sudah dibuat;

Dalam keamanan terdapat dua masalah penting, yaitu :

**1. Kehilangan Data (Data Loss)**

Kehilangan data dapat disebabkan karena beberapa hal, yaitu :

a. Bencana

- Kebakaran

- Banjir

- Gempa Bumi

- Perang

- Kerusuhan

- Binatang

b. Kesalahan perangkat keras dan perangkat lunak

- Tidak berfungsinya pemroses

- Disk atau tape yang tidak terbaca

- Kesalahan telekomunikasi

- Kesalahan program (bugs)

c. Kesalahan / kelalaian manusia

- Kesalahan pemasukan data

- Memasang tape atau disk yang salah

- Eksekusi program yang salah

- Kehilangan disk atau tape

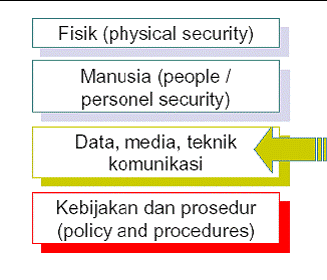
**2. Penyusup (hacker)**

Penyusup atau hacker terdiri dari :

a. Penyusup pasif (Membaca data yang tidak diotorisasi)

b. Penyusup aktif (Mengubah data yang tidak diotorisasi)

Klafisifikasi Keamanan Sistem Informasi menurut David Icove



Untuk meminimalkan kemungkinan terjadinya bencana (*disaster*), kesalahan (*errors*), interupsi pelayanan, kejahatan terhadap pemanfatan komputer, dan pelanggaran sistem pengamanan komputer, perlu dibangun kebijakan dan prosedur khusus ke dalam desain dan implementasi sistem informasi. Perlu dibangun pengendalian sistem informasi yang terdiri dari seluruh metode, kebijakan, dan prosedur organisasi yang dapat memastikan keamanan aset organisasi, keakuratan dan dapat diandalkannya catatan dan dokumen akuntansi, dan aktivitas operasional mengikuti standar yang ditetapkan manajemen. Pengendalian atas sistem informasi harus menjadi bagian yang terintegrasi sejak sistem informasi ini dirancang.

Pengendalian umum sistem informasi berhubungan dengan risiko-risiko yang berkaitan di berbagai area kegiatan, seperti: sistem operasi, sumber daya data, pemeliharaan sistem, pusat komputer, komunikasi data, pertukaran data elektronik (*electronic data interchange*-EDI), komputer mikro, dan sebagainya.

**Jenis Ancaman pada sistem informasi  
  
1. Ancaman dari Alam**Ancaman dari alam bisa menjadi hal yang amat sangat fatal bila tidak diantisipasi sejak mendesign sistem yang akan dijalankan. Karena kerusakan atau ancaman yang berasal dari alam akan merusak sistem tersebut dengan kerusakan yang sangat parah. Contohnya adalah banjir, tsunami, gempa, dan bencana lainnya. Walaupun tidak semua fenomena alam yang bersifat merusak tidak dapat diprediksikan, namun sekiranya perlu untuk mendesign sistem agar terhindar dari segala kemungkinan bencana alam yang terjadi.  
  
**2. Ancaman dari Manusia**Ancaman yang berasal dari manusia bisa dilakukan baik dari pihak dalam maupun luar, contoh dari ancaman yang berasal dari manusia adalah :

1. Pencurian

2. Penyadapan

3. Pembocoran info internal kepada konpetitor

4. Hacking

5. Human error (Kesalahan yang tidak disengaja)

**PEMBANGUNAN SISTEM KEAMANAN INFORMASI**

**Terminologi**

**Kriptografi** (*cryptography*) merupakan ilmu dan seni untuk menjaga pesan agar aman. (*Cryptography is the art and science of keeping messages secure.* [40]) “*Crypto*” berarti “*secret*” (rahasia) dan “*graphy*” berarti “*writing*” (tulisan) [3]. Para pelaku atau praktisi kriptografi disebut

***cryptographers***. Sebuah algoritma kriptografik (*cryptographic algorithm*), disebut **cipher**, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat.

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah **enkripsi** (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat

digunakan adalah “*encipher*”. Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut **dekripsi** (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah “*decipher*”. *Cryptanalysis* adalah seni dan ilmu untuk memecahkan *ciphertext* tanpa

bantuan kunci. *Cryptanalyst* adalah pelaku atau praktisi yang menjalankan *cryptanalysis*. *Cryptology* merupakan gabungan dari *cryptography* dan *cryptanalysis*.

**Enkripsi**

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi data anda disandikan (*encrypted*) dengan menggunakan sebuah kunci (*key*). Untuk membuka (*decrypt*) data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi (untuk kasus *private key cryptography*) atau dengan kunci yang berbeda (untuk kasus *public key cryptography*). enkripsi dan dekripsi dengan dua kunci yang berbeda. Secara matematis, proses atau fungsi enkripsi (*E*) dapat dituliskan sebagai:

**(1)** dimana: *M* adalah *plaintext* (*message*) dan *C* adalah *ciphertext*. Proses atau fungsi dekripsi (*D*) dapat dituliskan sebagai:

**(2) Elemen dari Enkripsi**

Ada beberapa elemen dari enkripsi yang akan dijabarkan dalam beberapa paragraf di bawah ini.

**Algoritma dari Enkripsi dan Dekripsi.**

Algoritma dari enkripsi adalah fungsi-fungsi yang digunakan untuk melakukan fungsi enkripsi dan dekripsi. Algoritma yang digunakan menentukan kekuatan dari enkripsi, dan ini biasanya dibuktikan dengan basis matematika. Berdasarkan cara memproses teks (*plaintext*), cipher dapat dikategorikan menjadi dua jenis: *block cipher* and *stream cipher*. Block cipher bekerja

dengan memproses data secara blok, dimana beberapa karakter / data digabungkan menjadi satu blok. Setiap proses satu blok menghasilkan keluaran satu blok juga.

**Data Encryption Standard (DES)**

DES, atau juga dikenal sebagai *Data Encryption Algorithm* (DEA) oleh ANSI dan DEA-1 oleh ISO, merupakan algoritma kriptografi simetris yang paling umum digunakan saat ini. Sejarahnya DES dimulai dari permintaan pemerintah Amerika Serikat untuk memasukkan proposal enskripsi. DES memiliki sejarah dari Lucifer1, enkripsi yang dikembangan di IBM kala itu.

Horst Feistel merupakan salah satu periset yang mula-mula mengembangkan DES ketika bekerja di IBM Watson Laboratory di Yorktown Heights, New York. DES baru secara resmi digunakan oleh pemerintah Amerika Serikat (diadopsi oleh National Bureau of Standards)

di tahun 1977. Ia dikenal sebagai Federal Information Processing Standard 46 (FIPS PUB46).

Aplikasi yang menggunakan DES antara lain:

**•** enkripsi dari password di sistem UNIX

**•** berbagai aplikasi di bidang perbankan

**Memecahkan DES**

DES merupakan block chiper yang beroperasi dengan menggunakan blok berukuran 64-bit dan kunci berukuran 56-bit. Brute force attack dengan mencoba segala kombinasi membutuhkan 256 kombinasi atau sekitar 7x 1017 atau 70 juta milyar kombinasi. DES dengan penggunaan yang biasa (*cookbook mode*) dengan panjang kunci 56 bit saat ini sudah dapat dianggap tidak aman karena sudah berhasil dipecahkan dengan metoda coba-coba (*brute force attack*). Ada berbagai group yang mencoba memecahkan DES dengan berbagai cara. Salah satu group yang bernama ***distributed.net*** menggunakan teknologi Internet untuk memecahkan problem ini menjadi sub-problem yang kecil (dalam ukuran blok). Pengguna dapat menjalankan sebuah program yang khusus dikembangkan oleh tim ini untuk mengambil beberapa blok, via Internet, kemudian memecahkannya di computer pribadinya. Program yang disediakan meliputi berbagai operating system seperti Windows, DOS, berbagai variasi Unix, Macintosh. Blok yang sudah diproses dikembalikan ke *distributed.net* via Internet. Dengan cara ini puluhan ribu orang, termasuk penulis, membantu memecahkan DES. Mekanisme ini dapat memecahkan DES dalam waktu 30 hari. Sebuah group lain yang disebut *Electronic Frontier Foundation* (EFF) membuat sebuah komputer yang dilengkapi dengan *Integrated Circuit chip DES cracker*. Dengan mesin seharga US$50.000 ini mereka dapat memecahkan DES 56-bit dalam waktu rata-rata empat (4) sampai lima (5) hari. DES cracker yang mereka kembangkan dapat melakukan eksplorasi keseluruhan dari 56-bit *keyspace* dalam waktu sembilan (9) hari. Dikarenakan 56-bit memiliki 216 (atau 65536) *keyspace* dibandingkan DES dengan 40-bit, maka untuk memecahkan DES 40-bit hanya dibutuhkan waktu sekitar 12 detik1. Dikarenakan hukum average, waktu rata-rata untuk memecahkan DES 40-bit adalah 6 detik.

**Hash function - integrity checking**

Salah satu cara untuk menguji integritas sebuah data adalah dengan memberikan “checksum” atau tanda bahwa data tersebut tidak berubah. Cara yang paling mudah dilakukan adalah dengan menjumlahkan karakterkarakter atau data-data yang ada sehingga apabila terjadi perubahan, hasil

penjumlahan menjadi berbeda. Cara ini tentunya mudah dipecahkan dengan menggunakan kombinasi data yang berbeda akan tetapi menghasilkan hasil penjumlahan yang sama. Pada sistem digital biasanya ada beberapa mekanisme pengujian integritas seperti antara lain:

**•** parity checking

**•** checksum

**•** hash function

Hash function merupakan fungsi yang bersifat satu arah dimana jika kita masukkan data, maka dia akan menghasilkan sebuah “checksum” atau “fingerprint” dari data tersebut. Ada beberapa hash function yang umum digunakan, antara lain:

**•** MD5

**•** SHA

Mengamankan Sistem Informasi

Pada umunya, pengamanan dapat dikategorikan menjadi dua jenis: pencegahan (*preventif*) dan pengobatan (*recovery*). Usaha pencegahan dilakukan agar sistem informasi tidak memiliki lubang keamanan, sementara usaha-usaha pengobatan dilakukan apabila lubang keamanan

sudah dieksploitasi. Pengamanan sistem informasi dapat dilakukan melalui beberapa layer yang

berbeda. Misalnya di layer “transport”, dapat digunakan “*Secure Socket Layer*” (SSL). Metoda ini umum digunakan untuk server web. Secara fisik, sistem anda dapat juga diamankan dengan menggunakan “firewall” yang memisahkan sistem anda dengan Internet. Penggunaan teknik enkripsi dapat dilakukan di tingkat aplikasi sehingga data-data anda atau e-mail anda

tidak dapat dibaca oleh orang yang tidak berhak.

**Mengatur akses (Access Control)**

Salah satu cara yang umum digunakan untuk mengamankan informasi adalah dengan mengatur akses ke informasi melalui mekanisme “*authentication*” dan “*access control*”. Implementasi dari mekanisme ini antara lain dengan menggunakan “*password*”. Di sistem UNIX dan Windows NT, untuk menggunakan sebuah sistem atau komputer, pemakai diharuskan melalui proses *authentication* dengan menuliskan “*userid*” dan “*password*”. Informasi yang diberikan ini dibandingkan dengan userid dan password yang berada di sistem. Apabila keduanya valid, pemakai yang bersangkutan diperbolehkan menggunakan sistem. Apabila ada yang salah, pemakai tidak dapat menggunakan sistem. Informasi tentang kesalahan ini biasanya dicatat dalam berkas *log*. Besarnya informasi yang dicatat bergantung kepada konfigurasi dari sistem

setempat. Misalnya, ada yang menuliskan informasi apabila pemakai memasukkan *userid* dan *password* yang salah sebanyak tiga kali. Ada juga yang langsung menuliskan informasi ke dalam berkas *log* meskipun baru satu kali salah. Informasi tentang waktu kejadian juga dicatat. Selain itu asal hubungan (*connection*) juga dicatat sehingga administrator dapat memeriksa keabsahan hubungan. Setelah proses *authentication*, pemakai diberikan akses sesuai dengan level yang dimilikinya melalui sebuah *access control*. *Access control* ini biasanya dilakukan dengan mengelompokkan pemakai dalam “group”. Ada group yang berstatus pemakai biasa, ada tamu, dan ada juga *administrator* atau *super user* yang memiliki kemampuan lebih dari group lainnya.

Pengelompokan ini disesuaikan dengan kebutuhan dari penggunaan sistemanda. Di lingkungan kampus mungkin ada kelompok mahasiswa, staf, karyawan, dan administrator. Sementara itu di lingkungan bisnis mungkin ada kelompok *finance*, *engineer*, *marketing*, dan seterusnya.

**Menutup servis yang tidak digunakan**

Seringkali sistem (perangkat keras dan/atau perangkat lunak) diberikan dengan beberapa servis dijalankan sebagai *default*. Sebagai contoh, pada sistem UNIX servis-servis berikut sering dipasang dari vendornya: *finger*, *telnet*, *ftp*, *smtp*, *pop*, *echo*, dan seterusnya. Servis tersebut tidak semuanya dibutuhkan. Untuk mengamankan sistem, servis yang tidak diperlukan di server (komputer) tersebut sebaiknya dimatikan. Sudah banyak kasus yang menunjukkan *abuse* dari servis tersebut, atau ada lubang keamanan dalam servis tersebut akan tetapi sang administrator tidak menyadari bahwa servis tersebut dijalankan di komputernya.

**Memasang Proteksi**

Untuk lebih meningkatkan keamanan sistem informasi, proteksi dapat ditambahkan. Proteksi ini dapat berupa filter (secara umum) dan yang lebih spesifik adalah firewall. Filter dapat digunakan untuk memfilter e-mail, informasi, akses, atau bahkan dalam level packet. Sebagai contoh, di system UNIX ada paket program “*tcpwrapper*” yang dapat digunakan untuk membatasi akses kepada servis atau aplikasi tertentu. Misalnya, servis untuk “*telnet*” dapat dibatasi untuk untuk sistem yang memiliki nomor IP tertentu, atau memiliki domain tertentu. Sementara firewall dapat

digunakan untuk melakukan filter secara umum. Untuk mengetahui apakah server anda menggunakan *tcpwrapper* atau tidak, periksa isi berkas /etc/inetd.conf. Biasanya *tcpwrapper* dirakit menjadi “tcpd”. Apabila servis di server anda (misalnya *telnet* atau *ftp*) dijalankan melalui tcpd, maka server anda menggunakan *tcpwrapper*. Biasanya, konfigurasi *tcpwrapper* (tcpd) diletakkan di berkas /etc/ hosts.allow dan /etc/hosts.deny.

**Firewall**

Firewall merupakan sebuah perangkat yang diletakkan antara Internet dengan jaringan internal (Lihat Figure 4.1 on page 65). Informasi yang keluar atau masuk harus melalui firewall ini. Tujuan utama dari firewall adalah untuk menjaga (*prevent*) agar akses (ke dalam maupun ke luar) dari orang yang tidak berwenang (*unauthorized access*) tidak dapat dilakukan. Konfigurasi dari firewall bergantung kepada kebijaksanaan (*policy*) dari organisasi yang bersangkutan, yang dapat dibagi menjadi dua jenis:

**•** apa-apa yang tidak diperbolehkan secara eksplisit dianggap tidak

diperbolehkan (*prohibitted*)

**•** apa-apa yang tidak dilarang secara eksplisit dianggap diperbolehkan (*permitted*) Firewall bekerja dengan mengamati paket IP (Internet Protocol) yang melewatinya. Berdasarkan konfigurasi dari firewall maka akses dapat diatur berdasarkan IP address, port, dan arah informasi. Detail dari konfigurasi bergantung kepada masing-masing firewall. Firewall dapat berupa sebuah perangkat keras yang sudah dilengkapi dengan perangkat lunak tertentu, sehingga pemakai (administrator) tinggal melakukan konfigurasi dari firewall tersebut. Firewall juga dapat berupa perangkat lunak yang ditambahkan kepada sebuah server (baik UNIX maupun Windows NT), yang dikonfigurasi menjadi firewall. Dalam hal ini, sebetulnya perangkat komputer dengan prosesor Intel 80486 sudah cukup untuk menjadi firewall yang sederhana. Firewall biasanya melakukan dua fungsi; fungsi (IP) filtering dan fungsi proxy. Keduanya dapat dilakukan pada sebuah perangkat komputer (device) atau dilakukan secara terpisah. Beberapa perangkat lunak berbasis UNIX yang dapat digunakan untuk melakukan IP filtering antara lain:

**•** *ipfwadm*: merupakan standar dari sistem Linux yang dapat diaktifkan pada level kernel

**•** *ipchains*: versi baru dari Linux kernel packet filtering yang diharapkan dapat menggantikan fungsi ipfwadm Fungsi proxy dapat dilakukan oleh berbagai software tergantung kepada jenis proxy yang dibutuhkan, misalnya web proxy, rlogin proxy, ftp proxy dan seterusnya. Di sisi client sering kalai dibutuhkan software tertentu agar dapat menggunakan proxy server ini, seperti misalnya dengan menggunakan SOCKS. Beberapa perangkat lunak berbasis UNIX untuk proxy antara lain:

**•** *Socks:* proxy server oleh NEC Network Systems Labs

**•** *Squid*: web proxy server

Informasi mengenai firewall secara lebih lengkap dapat dibaca pada referensi [26, 34] atau untuk sistem Linux dapat dilakukan dengan mengunjungi web site berikut: <<http://www.gnatbox.com>>.

Satu hal yang perlu diingat bahwa adanya firewall bukan menjadi jaminan bahwa jaringan dapat diamankan seratus persen. Firewall tersebut sendiri dapat memiliki masalah. Sebagai contoh, Firewall Gauntlet yang dibuat oleh Network Associates Inc. (NAI) mengalami masalah1 sehingga dapat melewatkan koneksi dari luar yang seharusnya tidak boleh lewat. Padahal Gauntlet didengung-dengungkan oleh NAI sebagai “*The World’s Most Secure Firewall*”. Inti yang ingin kami sampaikan adalah bahwa meskipun sudah menggunakan firewall, keamanan harus tetap dipantau secara berkala.

**Pemantau adanya serangan**

Sistem pemantau (*monitoring system*) digunakan untuk mengetahui adanya tamu tak diundang (*intruder*) atau adanya serangan (*attack*). Nama lain dari sistem ini adalah “*intruder detection system*” (IDS). Sistem ini dapat memberitahu administrator melalui e-mail maupun melalui mekanisme lain seperti melalui pager. Ada berbagai cara untuk memantau adanya intruder. Ada yang sifatnya aktif dan pasif. IDS cara yang pasif misalnya dengan memonitor logfile. Contoh software IDS antara lain:

**•** *Autobuse*, mendeteksi probing dengan memonitor logfile.

**•** *Courtney* dan *portsentry,* mendeteksi *probing* (*port scanning*) dengan memonitor packet yang lalu lalang. *Portsentry* bahkan dapat memasukkan IP penyerang dalam filter *tcpwrapper* (langsung dimasukkan kedalam berkas /etc/hosts.deny)

**•** *Shadow* dari SANS

**•** *Snort*, mendeteksi pola (*pattern*) pada paket yang lewat dan mengirimkan alert jika pola tersebut terdeteksi. Pola-pola atau *rules* disimpan dalam berkas yang disebut library yang dapat dikonfigurasi sesuai dengan kebutuhan.

**Pemantau integritas sistem**

Pemantau integritas sistem dijalankan secara berkala untuk menguji integratitas sistem. Salah satu contoh program yang umum digunakan di sistem UNIX adalah program *Tripwire*. Program paket *Tripwire* dapat digunakan untuk memantau adanya perubahan pada berkas. Pada mulanya,

tripwire dijalankan dan membuat database mengenai berkas-berkas atau direktori yang ingin kita amati beserta “*signature*” dari berkas tersebut. Signature berisi informasi mengenai besarnya berkas, kapan dibuatnya, pemiliknya, hasil *checksum* atau *hash* (misalnya dengan menggunakan program MD5), dan sebagainya. Apabila ada perubahan pada berkas tersebut, maka keluaran dari *hash function* akan berbeda dengan yang ada di database sehingga ketahuan adanya perubahan.

**Backup secara rutin**

Seringkali tamu tak diundang (*intruder*) masuk ke dalam sistem dan merusak sistem dengan menghapus berkas-berkas yang dapat ditemui. Jika intruder ini berhasil menjebol sistem dan masuk sebagai super user (administrator), maka ada kemungkinan dia dapat menghapus seluruh

berkas. Untuk itu, adanya backup yang dilakukan secara rutin merupakan sebuah hal yang esensial. Bayangkan apabila yang dihapus oleh tamu ini adalah berkas penelitian, tugas akhir, skripsi, yang telah dikerjakan bertahun-tahun. Untuk sistem yang sangat esensial, secara berkala perlu dibuat backup yang letaknya berjauhan secara fisik. Hal ini dilakukan untuk menghindari

hilangnya data akibat bencana seperti kebakaran, banjir, dan lain sebagainya. Apabila data-data dibackup akan tetapi diletakkan pada lokasi yang sama, kemungkinan data akan hilang jika tempat yang bersangkutan mengalami bencana seperti kebakaran.

**Penggunaan Enkripsi untuk meningkatkan keamanan**

Salah satau mekanisme untuk meningkatkan keamanan adalah dengan menggunakan teknologi enkripsi. Data-data yang anda kirimkan diubah sedemikian rupa sehingga tidak mudah disadap. Banyak servis di Internet yang masih menggunakan “*plain text*” untuk *authentication*, seperti

penggunaan pasangan userid dan password. Informasi ini dapat dilihat dengan mudah oleh program penyadap atau pengendus (*sniffer*). Contoh servis yang menggunakan plain text antara lain:

**•** akses jarak jauh dengan menggunakan telnet dan rlogin

**•** transfer file dengan menggunakan FTP

**•** akses email melalui POP3 dan IMAP4

**•** pengiriman email melalui SMTP

**•** akses web melalui HTTP

Penggunaan enkripsi untuk remote akses (misalnya melalui ssh sebagai penggani telnet atau rlogin) akan dibahas di bagian tersendiri.

**Telnet atau shell aman**

*Telnet* atau *remote login* digunakan untuk mengakses sebuah “*remote site*” atau komputer melalui sebuah jaringan komputer. Akses ini dilakukan dengan menggunakan hubungan TCP/IP dengan menggunakan userid dan password. Informasi tentang userid dan password ini dikirimkan melalui

jaringan komputer secara terbuka. Akibatnya ada kemungkinan seorang yang nakal melakukan “sniffing” dan mengumpulkan informasi tentang pasangan userid dan password ini1. Untuk menghindari hal ini, enkripsi dapat digunakan untuk melindungi adanya sniffing. Paket yang dikirimkan dienkripsi dengan algoritma DES atau Blowish (dengan menggunakan kunci session yang dipertukarkan via RSA atau Diffie-Hellman) sehingga tidak dapat dibaca oleh orang yang

tidak berhak. Salah satu implementasi mekanisme ini adalah SSH (Secure Shell). Ada beberapa implementasi SSH ini, antara lain:

**•** ssh untuk UNIX (dalam bentuk source code, gratis, mengimplementasikan protokol SSH versi 1 dan versi 2)

**•** SSH untuk Windows95 dari Data Fellows (komersial, ssh versi 1 dan versi 2) <http://www.datafellows.com/>

**•** TTSSH, yaitu skrip yang dibuat untuk *Tera Term Pro* (gratis, untuk Windows 95, ssh versi 1)

<http://www.paume.itb.ac.id/rahard/koleksi>

**•** SecureCRT untuk Windows95 (shareware / komersial)

**•** putty (SSH untuk Windows yang gratis, ssh versi 1). Selain menyediakan ssh, paket putty juga dilengkapi dengan pscp yang mengimplementasikan secure copy sebagai pengganti FTP.