WinBI-NG: Solusi desktop aman (secure desktop)

Avinanta Tarigan*, I Made Wiryana†, Tedi Heriyanto‡

16 Nopember 2002

Ringkasan

Pemerintah Indonesia, melalui badan risetnya, BPPT (Badan Pengkajian dan Penerapan Teknologi) dan yang didukung oleh komunitas Linux Indonesia, universitas, dan organisasi bisnis, meluncurkan sebuah proyek yang bernama WinBI (Window Bahasa Indonesia)[1], sebagai bagian Proyek Software Indonesia. Sistem operasi desktop WinBI ini berbasiskan sistem operasi Trustix Merdeka ini bertujuan mengembangkan sistem operasi desktop yang menggunakan bahasa Indonesia. Saat ini proyek WinBI telah merilis versi 1.0 software dan dokumentasinya. Pengembangan WinBI saat ini relatif dilakukan secara terbuka. Pada tulisan ini WinBI digunakan sebagai istilah sistem operasi desktop berbasiskan GNU/Linux yang menggunakan bahasa Indonesia sebagai bahasa pengantar.

Dalam tulisan ini kami akan mengetengahkan beberapa fitur baru yang akan ditambahkan dalam WinBI versi mendatang, kami menyebut WINBI ini sebagai WINBI Next Generation (WinBI-NG). Fitur-fitur tersebut secara garis besar bertujuan mengembangkan WinBI menjadi suatu solusi *secure desktop*. Pada pengembangan ini dimplementasikan SmartCard dan Public Key Infrastructure (PKI), serta penggunaan GUI berbahasa Indonesia yang dapat menambah keamanan penggunaan sistem.

Sekuriti penting untuk membangun kepercayaan (*trust*) terhadap sebuah sistem informasi. Sekuriti sering dipandang hanyalah merupakan masalah teknis yang melibatkan bisa atau tidak tertembusnya suatu sistem. Pada pandangan makro sekuriti sendiri memiliki konsep yang lebih luas, juga berkaitan dengan ketergantungan suatu institusi terhadap institusi lainnnya, atau suatu negara terhadap negara lainnya.

Kami pun akan mengetengahkan berbagai kendala yang ditemui dalam penambahan fitur-fitur tersebut serta solusi-solusi yang kami gunakan untuk mengatasi permasalahan-permasalahan tersebut. Pengembangan perangkat bantu untuk pekerjaan penerjemahan akan juga dijabarkan dalam tulisan ini. Perangkat bantu ini akan mempercepat proses penerjemahan program menjadi berbahasa Indonesia.

1 Pendahuluan

Pada saat ini teknologi informasi (TI) telah mulai bermetamorfosa menjadi suatu tahapan teknologi yang pervasif yang terasa keberadaannya ketika tidak ada (atau tak bekerja). Dengan kata lain kita sudah mulai tergantung dengan bantuan teknologi informasi. Kemajuan perkembangan Internet dan World Wide Web (WWW) telah menunjukkan suatu langkah ke arah ini. Konsekuensi dari sistem informasi yang menjadi pervasif adalah timbulnya dampak yang besar pada masyarakat secara luas. Akan banyak industri yang berubah atau digantikan sama sekali. Juga akan banyak tumbuh industri baru sesuai dengan kebutuhan perkembangan teknologi informasi itu.

*`avinanta@rvs.uni-bielefeld.de`. Dosen tetap Universitas Gunadarma, anggota RVS Arbeitsgruppe Universitas Bielefeld

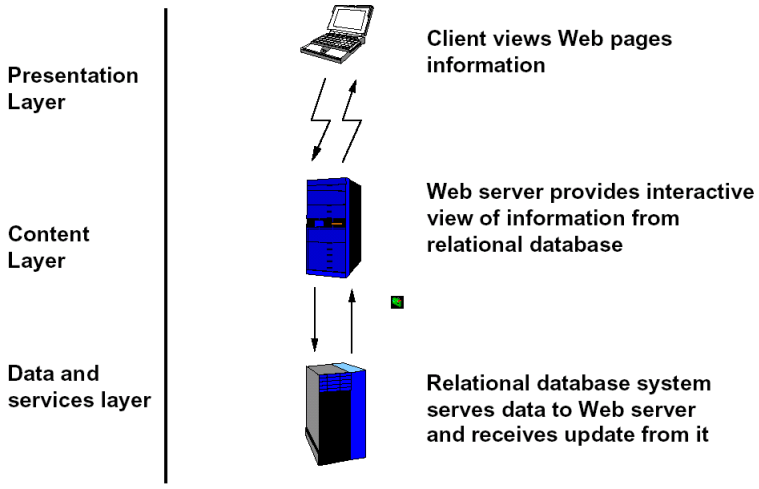
†`mwiryana@rvs.uni-bielefeld.de`. Dosen tetap Universitas Gunadarma, anggota RVS Arbeitsgruppe Universitas Bielefeld

‡`tedi_h@gmx.net`. LCP (Linux is Cool, Pren), LCI (Linux Cool Ih), RHCH (Red Hat Certified Humorist), MCP (Microsoft Certified Pirate), EGP, MBA (Mboten Boten Aja)

[1] `http://www.software-ri.or.id/winbi/`. Penggunaan kata **Window** disebabkan kata tersebut telah begitu dikenal oleh masyarakat dibandingkan istilah yang lebih generik yaitu Graphical User Interface (GUI)

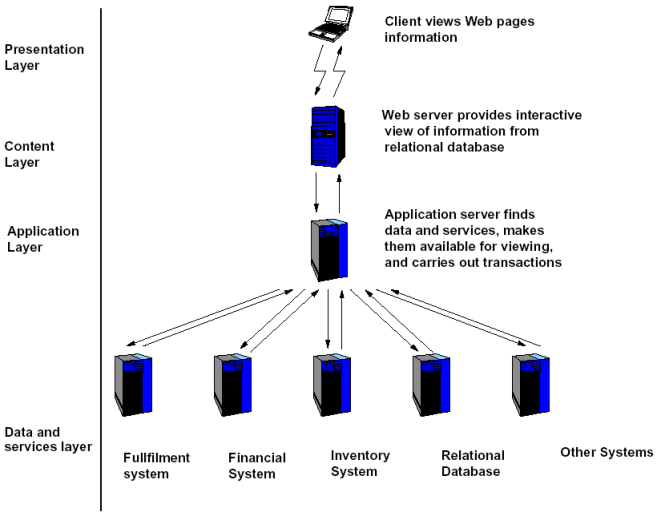
Di samping iming-iming keuntungan dari pemanfaatan teknologi informasi, sangatlah tidak realistik bila mengasumsikan bahwa teknologi informasi tidak menimbulkan permasalahan dalam penerapannya. Informasi jelas dapat disalah-gunakan. Polusi informasi, yaitu propagasi informasi yang salah, dan pemanfaatan informasi (baik benar atau salah) untuk mengendalikan hidup manusia tanpa atau dengan disadari merupakan suatu akibat dari penyalah-gunaan ini. Misinformasi akan terakumulasi dan menyebabkan permasalahan pada masyarakat.

Internet, awalnya dikembangkan untuk menghubungkan antar pihak yang saling dipercaya untuk tujuan saling bertukar menukar informasi dan untuk menyediakan data-data atau publikasi ilmiah. Walau merupakan proyek Departemen Pertahanan Amerika, Internet digunakan dan dikembangkan untuk tujuan kolaborasi dunia akademi yang serba terbuka. Perkembangan Internet yang pesat dan kini telah menjadi suatu jaringan raksasa yang saling menghubungkan berbagai jaringan. Pemanfaatannya di bidang bisnis menjadikan terjadinya pergeseran model. Dari bentukan komunitas pengguna Internet yang cenderung berupa suatu *Gemainschaft* dengan norma internal dan tradisi yang diatur berdasarkan status dan didorong oleh kecintaan, kewajiban serta kesamaan pemahaman dan tujuan, Sekarang telah bergeser dan cenderung menjadi suatu *Gesselschaft* yang terdiri dari individu (organisasi) yang memiliki interest masing-masing yang saling berkompetisi untuk kepentingan material sehingga berbentuk pasar bebas. Pada bentuk pertama bisa dikatakan tak ada batasan antara privat dan publik, sedang pada yang kedua terjadi perbedaan secara jelas.



Gambar 1: Generasi 1 mekanisme Web

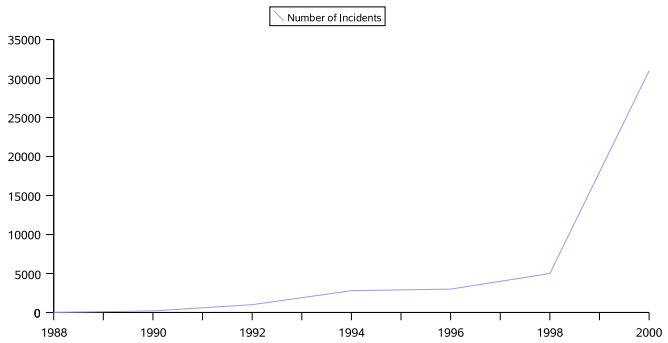
Kini Internet telah berkembang dan menyediakan berbagai layanan. Sehingga dikenalnya konsep "Service" yang lebih kompleks daripada sekedar menyediakan dokumen di suatu situs Internet. Konsep service ini lebih umum dibanding konsep lapisan jaringan yang sangat teknis. Perubahan fungsi, serta komunitas pengguna Internet tampaknya belum diikuti dengan perubahan drastis teknologi jaringan yang mendasarinya. Teknologi yang digunakan relatif masih memanfaatkan TCP/IP yang serba terbuka. Terbuka di sini bukan berarti source code atau standardnya diketahui banyak orang, tetapi dalam mekanismenya yang masih membuka alamat tujuan dan pengirimnya. Ketertutupan informasi yang berkaitan dengan suatu protokol bukan merupakan suatu jaminan bahwa protokol itu akan lebih aman. Seperti diketahui algoritma atau mekanisme kriptografi yang menjadi sandaran usaha penyusunan jalur komunikasi aman pun menggunakan algoritma yang mekanismenya diketahui oleh orang banyak. Beberapa protocol telah dikembangkan untuk mengatasi kekurangan protocol TCP/IP, seperti QoS, IPSEC (IP Secure), IP-NG (IP New Generation).



Gambar 2: Generasi ke 2 mekanisme Web

1.1 Ancaman sekuriti era Internet

Security incidents merupakan proses dan hasil pelanggaran sekuriti suatu sistem baik oleh penggunanya maupun entitas di luar sistem tersebut. Menurut **Computer Security Institute** (CSI) `<http://www.gocsi.com>`, trend terjadinya security incidents menjadi semakin tinggi pada tahun 2000 dan kerugian finansial yang diakibatkan mencapai US$ 377.828.700 pada quarter pertama tahun 2001.



Gambar 3: Security Incident Trend (CSI, 2001)

Security incident merupakan hasil dari suatu ancaman digital (*digital threat*) terhadap suatu sistem oleh entitas yang sering disebut sebagai "**Cracker**". Berbeda dengan Cracker, adalah suatu entitas yang disebut dengan **hacker**. Hacker adalah entitas yang menemukan kelemahan (*vulnerability*) sistem dalam konteks security incidents. Seringkali cracker memanfaatkan hasil penemuan tersebut untuk melakukan eksploitasi dan mengambil manfaat dari hasilnya. Seorang hacker bisa menjadi seorang cracker, tetapi seorang cracker belum tentu menguasai kemampuan yang dipunyai seorang hacker.

Saat ini banyak tersedia perangkat lunak untuk melakukan eksploitasi kelemahan sistem. Software tersebut dapat didownload secara bebas dari Internet dan disebut dengan "*automate exploit tools*". Awalnya perangkat lunak ini digunakan untuk pengujian sistem (*penetration test*). Tapi dengan berbekal software ini, seorang cracker dapat melakukan exploitasi di mana saja dan ka-

pan saja, tanpa harus mempunyai pengetahuan khusus. Cracker jenis ini dikenal sebagai "**script kiddies**".

Motivasi para hacker untuk menemukan vunerability adalah untuk membuktikan kemampuannya atau sebagai bagian dari kontrol sosial terhadap sistem. Sedangkan motivasi para cracker sangat beragam, diantaranya adalah untuk propaganda (deface web site/email), kriminal murni, penyerangan destruktif (akibat dendam atau ketidak-sukaan terhadap suatu insitusi), dan lain-lain. Apapun motif dari cracker selalu ada pihak yang dirugikan akibat tindakannya.

Pada prakteknya suatu pembentukan sistem yang aman akan mencoba melindungi adanya beberapa kemungkinan serangan yang dapat dilakukan pihak lain terhadap kita antara lain (Tarigan dan Wiryana, 2000):

- **Intrusion**. Pada penyerangan ini seorang penyerang akan dapat menggunakan sistem komputer yang kita miliki.
- **Denial of services**. Penyerangan jenis ini mengakibatkan pengguna yang sah tak dapat mengakses sistem.
- **Joyrider**. Pada serangan ini disebabkan oleh orang yang merasa iseng dan ingin memperoleh kesenangan dengan cara menyerang suatu sistem.
- **Vandal**. Jenis serangan ini bertujuan untuk merusak sistem. Seringkali ditujukan untuk site-site besar.
- **Scorekeeper**. jenis serangan in hanyalah bertujuan untuk mendapatkan reputasi dengan cara mengcrack sistem sebanyak mungkin. Saat ini jenis ini lebih dikenal dengan istilah **script kiddies**
- **Mata-mata**. Jenis serangan ini bertujuan untuk memperoleh data atau informasi rahasia dari pihak kompetitor.

Serangan pada suatu sistem jaringan komputer sendiri pada dasarnya memiliki 3 gelombang trend utama yaitu (Wiryana, 2001b)

- Gelombang pertama adalah serangan **fisik** . Serangan ini ditujukan kepada fasilitas jaringan, perangkat elektronis dan komputer.
- Gelombang kedua adalah serangan **sintatik**. Serangan ini ditujukan terhadap keringkihan (*vulnerability*) pada perangkat lunak, celah yang ada pada algoritma kriptografi atau protokol.
- Gelombang ketiga adalah serangan **semantik**. Serangan jenis ini memanfaatkan arti dari isi pesan yang dikirim. Dengan kata lain adalah menyebarkan disinformasi melalui jaringan.

Masih banyak orang yang menyepelekan serangan gelombang ke tiga ini. Serangan bentuk ini dapat dilakukan, misal dengan memposting informasi yang salah ke suatu forum diskusi, mengirimkan email berantai dan sebagainya. Akan lebih rawan lagi bila seseorang dengan melakukan serangan gelombang kedua (sintatik) dapat masuk ke database suatu media online. Lalu melakukan serangan semantik dengan mengubah berita yang ditayangkan pada media online tersebut. Karena relatif masyarakat dan pembaca mempercayai isi berita yang ditayangkan, maka serangan semantik seperti ini akan menimbulkan dampak yang lebih parah lagi. Serangan semantik ini sebetulnya sudah merupakan salah satu senjata *lumrah* dalam kegiatan dinas intelijen. Model serangan ini lazim digolongkan dalam kegiatan *active measure* (Womack, 1998).

1.2 Kebutuhan akan secure desktop

Sebagian besar usaha untuk meningkatkan sekuriti sistem banyak difokuskan pada server, padahal banyak insiden disebabkan oleh kesalahan pengguna akibat sistem desktop yang kurang menjamin keamanan aktivitas penggunanya. Insiden "klikbca.com" , virus, trojan, dan penyadapan nomor kartu kredit oleh keystroke-broadcast program menunjukkan bahwa sisi user adalah mata rantai terlemah keamanan suatu sistem. Begitu juga adanya virus komputer yang sering mengakibatkan sistem tak dapat bekerja semestinya, atau bahkan sebuah virus komputer dapat bekerja sebagai penyadap tanda tangan digital ataupun penyadap data lainnya yang penting. Virus komputer atau malware lainnya banyak menyerang desktop pengguna. Sehingga seaman-amannya sistem server, maka menjadi tidak berarti ketika desktop yang digunakan tidak memiliki keamanan dan integritas yang baik.

Secure desktop merupakan suatu lingkungan kerja di mana user dapat bekerja dan melakukan aktivitas dengan dukungan sekuriti yang memadai. Secure desktop diharapkan akan mampu menangani dan mendukung keamanan aktivitas user yang memerlukan tingkat keamanan yang tinggi seperti melakukan entri data melalui Internet. Hal ini dibutuhkan misal untuk desktop di aplikasi perbankan, atau untuk Pemilihan Umum (PEMILU). Sayangnya seringkali sistme operasi dan sistem desktop tidak begitu dipertimbangkan dalam merancang sistem yang aman tersebut.

Untuk memenuhi konsep secure desktop maka beberapa hal perlu dipertimbangkan

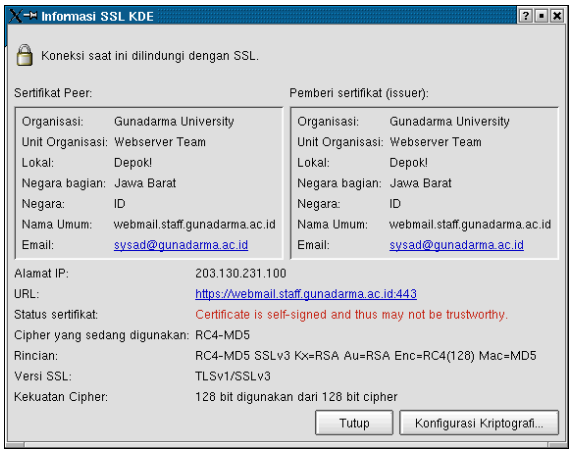
- Perangkat peripheral yang terkontrol (tanpa CDROM, tanpa disket, serial dan USB), sehingga sulit bagi pengguna untuk secara bebas memasukkan program tanpa kontrol yang pasti.
- Ruang alamat yang terproteksi sehingga aplikasi yang tak memiliki hak akses tinggi tak bisa melanggar batasan tersebut.
- API yang aman.
- Struktur berkas (*filesystem*) yang aman, misal dilengkapi dengan sistem berkas terenkripsi.

Dengan menggunakan sistem GNU/Linux seperti WinBI maka pengguna dapat diatur sehingga :

- Pengguna hanya dapat menggunakan aplikasi yang diizinkan. Dengan kata lain pengguna ataupun virus yang memanfaatkan hak akses pengguna tak dapat menginstal aplikasi baru.
- Pengguna hanya dapat menggunakan ruang berkas yang diizinkan.
- Pengguna tak dapat menginstal atau memodifikasi program yang ada.
- Aksi pengguna tak dapat merusak integritas sistem, misal keberadaan virus tak akan mengganggu sistem atau pengguna yang lainnya.

Bahasa yang digunakan oleh program untuk berinteraksi dengan user juga merupakan hal yang sangat penting karena langkah-langkah pengamanan yang seharusnya dimengerti secara jelas dan gamblang oleh user, seperti pada kotak dialog SSL yang dimunculkan oleh browser web, seringkali diabaikan oleh user akibat kendala bahasa. Hal-hal di atas sudah cukup menjadikan dasar bahwa perlu dikembangkan sebuah sistem desktop yang mendukung keamanan aktivitas usernya. Pertimbangan keamanan desktop yang aman tidak saja dari sisi teknis tetapi juga dari sisi pengguna dan organisasi.

Bahasa pengantar pada GUI juga dapat menjadikan suatu celah kemanan. Sebagai contoh, berikut ini disajikan faktor GUI berbahasa Indonesia yang dapat menambah fitur keamanan pada sistem. Karena dengan penjelasan yang dipahami oleh pengguna maka kegamangan akan makna dari istilah yang tak diketahuinya bisa dicegah. Pengguna tidak sekedar menekan "Yes" yang ternyata membawa pengguna ke dalam situasi yang tak aman (misal kasus situs palsu). Dengan cara menyediakan GUI dalam bahasa yang lebih dipahami oleh pengguna, maka sistem dapat dioperasikan secara lebih aman. Dengan memberikan informasi yang lebih mudah dipahami maka pengguna dapat terhindar dari celah sekuriti misal seperti pada kasus KLIKBCA.



Gambar 4: GUI berbahasa Indonesai mencegah typosquat

2 Konsep security

Sekuriti komputer memiliki definisi yang beragam, sebagai contoh berikut ini adalah definisi sekuriti komputer yang sering digunakan (Gollmann, 1999) :

Computer security deals with the prevention and detection of unauthorized actions by users of a computer system.

2.1 Segitiga CIA

Seringkali orang sering mempertimbangkan masalah akses yang tidak sah saja dalam sekuriti. Sebetulnya hal yang perlu dipertimbangkan adalah lebih luas. Dalam perancangan dan pembahasan sistem sekuriti kazimnya kita akan dihadapkan pada pertimbangan yang dikenal dengan istilah **segitiga CIA**

- **Confidentiality,** yang akan berkaitan dengan pencegahan akan pengaksesan terjadap informasi yang dilakukan oleh pihak yang tak berhak.
- **Integrity**, yang akan berkaitan dengan pencegahan akan modifikasi informasi yang dilakukan oleh pihak yang tak berhak.
- **Availability,** pencegahan akan penguasaan informasi atau sumber daya oleh pihak yang tak berhak.

Disain suatu sistem sekuriti akan mencoba menyeimbangkan ke tiga hal di atas. Setiap user harus bertanggung jawab terhadap aksi yang dilakukan pada sistem. Untuk itulah konsep accountability menjadi penting pada sistem komputer.

- **Accountability** : Yang berarti informasi audit harus tersimpan dengan selektif dan terlindungi sehingga akses yang menyebabkan permasalah sekuriti dapat dijejaki untuk mengetahui pihak yang bertanggung jawab.

Dengan kata lain merupakan suatu proses pencatatan yang memadai atas pemakaian resources dalam suatu sistem oleh para penggunanya. Tidak semua sistem operasi memiliki penanganan accountability yang baik. Hal ini terutama kepada sistem operasi yang bukan berkelas multiuser, misal OS/2, MS DOS, atau MS Windows 95.

Dalam membangun sebuah sistem informasi, perlu diperhatikan beberapa objektif dari sekuriti komputer untuk dipertimbangkan dalam desain, implementasi, dan operasional. Di samping hal di atas ada ada beberapa objektif sekuriti yang penting dan diperlukan sebagai pertimbangan dalam membangun sekuriti adalah :

- **Authentication**. Sekuriti menjamin proses dan hasil identifikasi oleh sistem terhadap pengguna dan oleh pengguna terhadap sistem
- **Non Repudiation**. Setiap informasi yang ada dalam sistem tidak dapat disangkal oleh pemiliknya

Pendekatan tradisional pada sekuriti komputer hanya berorientasi pada teknologi dan produk (*hardware* dan *software*). Dalam pendekatan ini, terdapat anggapan bahwa hanya sebagian orang saja yang harus mengerti dan bertanggungjawab dalam masalah sekuriti. Di samping itu, pihak manajemen menempatkan sekuriti komputer pada prioritas yang rendah. Pendekatan tradisional biasanya ditandai dengan ketidak-mengertian pengguna akan pentingnya keikut-sertaan mereka dalam membangun sekuriti. Pengguna menganggap dengan membeli dan menggunakan produk-produk sekuriti seperti firewall dan kriptografi dapat menjamin keamanan suatu sistem.

Pendekatan tradisional harus dihindari dalam membangun sekuriti. Kenyataan menunjukkan bahwa pengguna adalah mata rantai terlemah dalam rantai sekuriti itu sendiri. Oleh karena itu diperlukan pendekatan modern yang komprehensif, yang mengikutsertakan user, policy, manajemen, dan teknologi. Pada hakekatnya seringkali orang melupakan bahwa dalam pelaksanaan sekuriti akan melibatkan **3 M** yaitu (Tarigan dan Wiryana, 2002) :

- Matematika
- Manajemen
- Manusia

Berikut ini akan dibahas lebih dalam mengenai pertimbangan tersebut. Dalam bahasan kali ini manusia akan menempati porsi yang cukup besar, terutama karena sistem desktop adalah sistem yang berinterkasi secara lansung dengan manusia.

2.2 Sistem operasi dan aplikasi

Seringkali security incidents disebabkan oleh kelemahan akibat adanya "bug" dalam sistem operasi, aplikasi server, atau aplikasi desktop. Oleh karena itu, pemilihan sistem operasi atau aplikasi merupakan hal yang penting sebelum sistem tersebut dioperasikan. Bug dalam software tersebut muncul karena kesalahan desain dan proses pengembangan yang kurang tepat. Seringkali vendor software mengutamakan kecepatan waktu dan penghematan biaya pengembangan sehingga mengorbankan sekuriti. Selain menyebabkan kelemahan sekuriti pada sistem, bug menyebabkan sistem tidak bekerja dengan baik. Hal ini menyebabkan tidak tersedianya layanan sistem (*availability*).

- **Sebaiknya hindari penggunaan sistem operasi desktop yang tidak menjaga integritas, atau lengkapi dengan utilitas bantu**. Seringkali masalah keamanan juga timbul akibat pengguna menggunakan sistem operasi yang tidak memiliki proteksi terhadap kernel (bagian sistem operasi). Atau tidak bersifat multi user. (Michener, 1999). Sehingga pengaruh virus, plug-in dapat menyebabkan tingkat sekuriti rendah. Bila pengguna memutuskan menggunakan sistem operasi semacam ini (misal DOS, Windows 95/98/ME), maka sudah sewajarnya tingkat pencegahan harus digunakan oleh pengguna tersebut.
- **Mewaspadai virus, plug-in, Active-X**. Active X dan plug-in dapat pula menyebabkan bahaya. Karena secara otomatis mereka dianggap "*trusted*" (dapat dipercaya). Untuk itu seringkali pengguna menghadapi dilema, karena seringkali suatu Internet Banking mengharapkan pengguna menginstall plug-in tertentu, atau suatu situs mendorong pengguna

menginstal plug-in. Padahal penggunaan plug-in yang diperoleh dari situs di Internet, seringkali sulit diuji keamanannya, dan meletakkan sistem komputer pengguna dalam resiko yang tinggi (Wiryana, 2001c).

- **Sebaiknya pengguna selalu mencatat dan menyimpan log akses ke Internet** . Log ini akan sangat penting sekali ketika timbulnya suatu kasus di masa mendatang. Sayang sekali tidak semua sistem operasi di sisi desktop secara otomatis menyediakan fasilitas log ini.

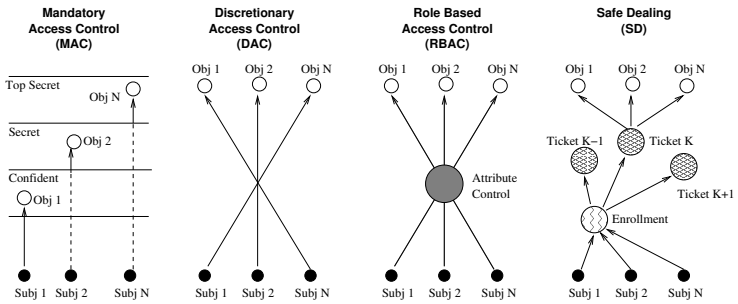
2.3 Otentikasi (*authentication*)

Dalam suatu sistem komputer, semua pengguna mempunyai ID sebagai identifier yang unik. Untuk dapat menggunakan sistem tersebut pengguna melakukan proses autentikasi, sehingga sistem secara dapat mengenal pengguna tersebut dan sebaliknya. Sistem sekuriti tradisional menggunakan *username* sebagai identifier dan *password* sebagai alat validasi. Teknologi berkembang sehingga saat ini terdapat beberapa mekanisme autentikasi :

- **Smartcard dan Secure Token**. Smartcard dan secure token menyimpan digital-ID dari pengguna. Sehingga secara fisik pengguna harus mempunyai smartcard atau securetoken untuk melakukan autentikasi. Untuk mengaktifkan smartcard pengguna diharuskan untuk memasukkan PIN atau *keyphrase*.
- **Biometric authentication**. Mekanisme autentikasi secara biologis memungkinkan sistem dapat mengenali penggunanya lebih tepat. Terdapat beberapa metode diantaranya : fingerprint scanning, retina scanning, dan DNA scanning. Dua metode terakhir masih dalam taraf penelitian, sedangkan fingerprint scanning saat ini telah digunakan secara luas dan digunakan bersama-sama dengan smartcard dalam proses autentikasi.
- **Public Key Infrastructure (PKI) dan Certification Authority**. Public Key Infrastructure memungkinkan mekanisme yang aman untuk melakukan autentikasi, secrecy, integrity, dan non repudiation. Mekanisme PKI berdasar pada private key - public key, dan institusi Certification Authority yang akan melakukan validasi terhadap setiap pengguna dan sistem.

2.4 Akses Kontrol

Sebuah sistem komputer memerlukan akses kontrol untuk melindungi, memberikan izin, dan mengatur pemakaian sumber daya yang ada dalam sistem tersebut, baik sumber daya fisik (memory, disk, processor, jaringan komputer) maupun data/infomasi. Sumber daya yang akan digunakan disebut dengan obyek, dan yang hendak menggunakannya disebut dengan subyek (pengguna sistem, proses, atau obyek lain). Akses kontrol merupakan bagian dari implementasi policy yang diterapkan dalam organisasi. Beberapa metode akses kontrol adalah sebagai berikut :



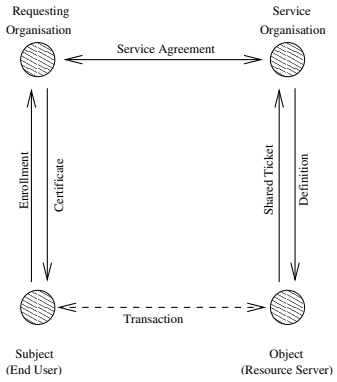
Gambar 5: Akses Kontrol

- **DAC (Discretionary Access Control)**. Dalam mekanisme DAC setiap objek mempunyai attribut akses berupa daftar setiap subjek dan hak aksesnya. Subjek dapat memodifikasi attribut akses (*read*, *write*, *run*) setiap objek yang dibuatnya dengan melakukan proses *granting* (mengijinkan akses) dan *revoking* (menolak akses). DAC merupakan akses kontrol yang fleksibel dan digunakan secara luas, tetapi DAC tidak menjamin tingkat sekuriti yang tinggi dalam implementasinya.

- **MAC (Mandatory Access Control)**. Dalam mekanisme MAC setiap subjek dan objek diklasifikasikan beradasarkan tingkat sensitifitas kerahasiaan informasi atau data yang telah didefinisikan sebelumnya. Setiap subjek hanya boleh mengakses objek yang ada di dalam level yang sama. MAC menjamin integritas dan secrecy dengan mengimplementasikan mekanisme multilevel sekuriti yang dikenal juga dengan nama Bell-LaPadulla. MAC menjamin tingkat sekuriti yang tinggi, tetapi tidak mudah dan sangat kaku dalam implementasinya. MAC diimplementasikan pada proyek-proyek sistem informasi yang membutuhkan tingkat sekuriti yang tinggi seperti pada sistem informasi militer.

- **RBAC (Role Based Access Control)**. RBAC disebut sebagai akses kontrol yang"policy neutral", karena kemudahan dalam mengimplementasikan sekuriti policy. Setiap akses oleh subjek harus mematuhi aturan-aturan (role) yang telah didefinisikan sebelumnya. Desain dan implementasi RBAC yang tepat dan baik menjamin tingkat sekuriti MAC serta mendapatkan fleksibilitas DAC. Keterbatasan RBAC adalah pada implementasinya di tingkat inter-organisasional, karena RBAC didisain untuk diterapkan secara internal dalam suatu institusi atau organisasi.

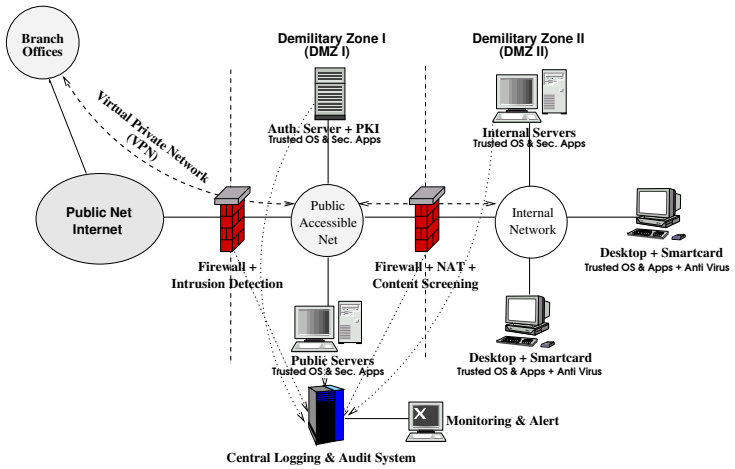
- **SD (Safe Dealing)**. Safe Dealing (SD) merupakan akses kontrol yang didisain untuk diterapkan pada inter-organisasi atau inter-institusi. Dalam arsitektur SD terdapat dua buah institusi perantara (t*rusted intermediate institution*) yang dipercaya oleh semua pihak untuk melakukan proses yang diperlukan untuk membangun sekuriti. Setiap user (subjek) melakukan pendaftaran kepada suatu institusi (misalnya global Certification Authority) yang akan memvalidasi identitas setiap user dan memberikan digital certificate kepada user tersebut. Institusi penyedia layanan (objek) mempercayakan akses kontrol yang telah didefinisikan sebelumnya kepada sebuah institusi yang mengelola akses kontrol setiap layanan (misalnya Chamber of Commerce). Institusi penengah tersebut melakukan perjanjian layanan (*service agreement*) terhadap kelompok user dan setiap layanan. Kedua institusi tersebut dapat melakukan akses bersama terhadap masing-masing data untuk memperoleh kesepakatan dalam konteks autentikasi, validasi, dan akses kontrol. Setiap akses oleh user akan diidentifikasi oleh kedua institusi penengah tersebut dalam bentuk mekanisme pemberian akses berdasarkan karcis (*ticket-granting*).



Gambar 6: Safe Dealing Administration

2.5 Firewall dan Intrusion Detection System

Firewall merupakan alat untuk mengatur akses kontrol pada level network di dalam suatu jaringan komputer. Firewall ditempatkan pada setiap entry-point untuk melakukan pemeriksaan serta otorisasi terhadap setiap paket transaksi yang masuk dan keluar ke dan dari jaringan tersebut berdasarkan rule atau aturan yang sudah didefinisikan sebelumnya. Firewall ini dapat dianalogikan seperti suatu pos penjagaan, yang akan memeriksa ID, memperbolehkan orang masuk, ataupun melarang seseorang masuk.



Gambar 7: Secure Network Architecture

Firewall terkini sudah dilengkapi dengan kapabilitas **Intrusion Detection System** (IDS) dan **Content Screening System**. Hal ini membuat sistem dapat menahan serangan pada level network. Apabila sistem mendeteksi paket transaksi yang mencurigakan (misalnya virus pada file, denial of service, atau intrusion attempt) maka paket tersebut akan ditolak sebelum sampai kepada tujuan berikutnya.

2.6 Koneksi yang aman

SSL (Secure Socket Layer) pada dasarnya merupakan suatu mekanisme yang melindungi koneksi dari usaha penyadapan. Hal ini karena komunikasi yang terjadi antara client-server melalui suatu jalur yang dienkripsi. Tetapi sistem ini tidak melindungi dari salah masuknya pengguna ke *host* yang berbahaya, ataupun tak melindungi apakah suatu kode yang didownload dari suatu situs bisa dipercaya, atau apakah suatu situs itu bisa dipercaya. Abadi (1996) telah menunjukkan kelemahan protokol SSL versi awal secara teoritis. Jadi jelas SSL ini tidak melindungi dari beberapa hal misal (Wiryana dan Heriyanto, 2001):

- Denial of Services
- Buffer overflow
- Man-in-the-middle attack
- Cross scripting attack

Pada model SSL, **user** -lah yang harus bertanggung jawab untuk memastikan apakah server di ujung sana yang ingin diajak berkomunikasi benar-benar merupakan server yang ingin dituju. Pada dunia nyata untuk meyakinkan bahwa orang yang dihubungi adalah orang sesungguhnya, dapat dilakukan dengan mudah karena orang saling mengenal. Dengan melihat muka, suara, bau dan sebagainya kita bisa mendeteksi bahwa dia orang yang sesungguhnya.

Pada dunia Internet hal seperti itu sulit dilakukan, oleh karenanya digunakan sertifikat digital untuk melakukan hal ini. Sertifikat ini mengikat antara suatu *public key* dengan suatu identitas. Sertifikat ini dikeluarkan oleh sebuah pihak yang disebut CA (*Certificate Authority*) misal dalam hal ini Verisign atau Thawte. CA sendiri memperoleh sertifikat dari CA lainnya. CA yang tertinggi disebut root dan tidak memerlukan sertifikat dari CA lainnya. Penanganan sertifikat ini dilakukan secara hierarki dan terdistribusi.

Sayangnya sertifikat digital saja, bukanlah obat mujarab yang bisa mengobati semua jenis permasalahan sekuriti. Agar SSL dapat bekerja dengan semestinya (melakukan koneksi terenkripsi dengan pihak yang semestinya), maka penggunalah yang harus memverifikasi apakah sertifikat yang dimiliki oleh server yang ditujunya adalah benar. Berikut ini adalah beberapa hal minimal harus diperhatikan :

- Apakah sertifikat tersebut dikeluarkan oleh CA yang dipercaya.
- Apakah sertifikat tersebut dikeluarkan untuk pihak yang semestinya (perusahaan yang situsnya dituju).
- Apakah sertifikat itu masih berlaku.

Sebetulnya ketika melakukan koneksi ke sebuah situs yang mendukung SSL, hal tersebut ditanyakan oleh browser, tetapi sebagian besar pengguna selalu menekan **Yes** ketika ditanya untuk verifikasi sertifikat ini. Untuk melihat ketiga hal tersebut, dapat dilakukan dengan *double-click* pada tombol kunci yang ada di bagian kiri bawah browser. Celah ini pada dasarnya dilakukan dengan cara mengalihkan akses user dari situs aslinya ke situs palsu lainnya, sehingga dikenal dengan istilah **page hijacking**. Beberapa kemungkinan teknik yang digunakan telah dijelaskan pada (Wiryana dan Heriyanto, 2001)

2.7 Public Key Infrastructure (PKI)

PKI merupakan teknik enkripsi public-key yang menjamin confidentiality, authentication, data integrity, dan non-repudiation. PKI merupakan pengejawantahan dari algoritma River Samir Adelmann (RSA) yang didalamnya mencakup teknologi dan fungsi legal/hukum. Digital Signature merupakan salah satu kemampuan dari teknologi PKI. Digital Signature dapat menjamin suatu dokumen elektronik otentik terhadap pembuatnya (*authentication*, *integrity*) dan mencegah terjadinya penyangkalan (*non-repudiation*) terhadap dokumen elektronik. Selain itu teknik enkripsi public mencegah terjadinya pembacaan data elektronik oleh yang tidak berhak (*confidentiality/secrecy*). Berikut ini akan dijelaskan secara singkat konsep dan prinsip dari Public Key Infrastructure (PKI) serta peran Certificate Authority (CA).

Perdagangan tradisional berbasiskan kertas dan "trust". Dalam perkembangan perdagangan tradisional telah dikenal sistem EDI yang bersifat : secure, closed, dan menggunakan sistem yang proprietary. Sedangkan saat ini eCommerce yang menggunakan Internet relatif bersifat tak aman, open dan memanfaatkan open system.

Di dunia Internet relatif sulit sekali memastikan apakah seseorang itu benar personal yang dimaksud. Sehingga timbul permasalahan mendasar dalam pemanfaatan eCommerce.

- **Authentication** : untuk mengidentifikasi pihak yang terlibat. Dalam perdagangan tradisional hal tersebut dilakukan dengan surat yang ditanda-tangani.
- **Confidentiality** : untuk menjaga informasi agar tetap privat. Dalam perdagangan tradisional surat ditulis dalam amplop dan lalu ditanda-tangani lalu di-"seal".
- **Integrity**: untuk melindungi manipulasi informasi. Hal ini dilakukan pengiriman dengan surat tercatat, lalu dibuat salinannya dan dikirimkan dua kali.
- **Non repudiation** : untuk menegah pengingkaran informasi oleh pemilik. Hal ini dapat dilakukan dengan adanya saksi yang menguji keabsahan tanda tangan tersebut.

Agar hal tersebut dapat tercapai dalam eCommerce maka perlu diterapkan langkah-langkah :

- **Kriptografi standard** (simetrik dan asimetrik). Kriptografi simetrik cepat, aman tetapi memiliki permasalahan pengelolaan key. Asimetrik kriptografi digunakan dalam public key kriptografi. Ada 2 key, private dan public key. Private key disimpan sendiri, dan publik key didistribusikan. Bila publik key digunakan untuk menenkripsi maka hanya private key yang dapat mendekripsi. Begitu juga sebaliknya.
- **One way hashing**. Menggunakan fungsi satu arah, dan tanpa key. Digunakan untuk menghasilkan suatu sidik data khas terhadap suatu kumpulan data. Digunakan untuk menentukan apakah suatu data telah berubah.
- **Tanda tangan digital.** Beberapa negara telah mensahkan penggunaan tanda tangan digital dalam transaksi elektronis. Tanda tangan digital ini akan menjamin otentikasi suatu dokumen.
- **Certificate Authority.** Suatu sistem yang mengikat kepemilikan public key dan pengguna sesungguhnya.

Langkah di atas dapat digunakan untuk membentuk "trust" dalam transaksi di Internet :

- **Authentication** : publik key digunakan untuk membuat digest dari pesan. Hanya dengan menggunakan private key dari pengirim maka dapat didekrip.
- **Confidentiality** : Pesan dienkripsi dengan menggunakan publik key dari penerima. Hanya dengan menggunakan private key dari pengirim pesan dapat didekripsi.
- **Integrity:** Membandingkan digest dengan tanda tangan digital yang didekripsi.
- **Non repudiation** : tanda tangan digital melakukan hal ini.

Key yang digunakan pada sistem kriptografi memegang peran yang sangat penting. Beberapa hal yang mempengaruhi ketahanan suatu key yang digunakan adalah :

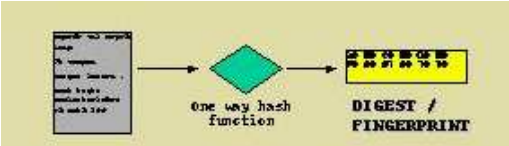
- Pseudo random number. Bilangan random ini digunakan untuk menghasilkan key yang akan digunakan. Semakin random bilangan yang dihasilkan maka kemungkinan tertebaknya key akan makin kecil.
- Panjangnya key, semakin panjang semakin aman. Tetapi perlu diingat bahwa membandingkan dua buah sistem kriptografi yang berbeda dengan berdasarkan panjang keynya saja tidaklah cukup.
- Private key harus disimpan secara aman baik dalam file (dengan PIN atau passphrase) atau dengan smart card.

2.8 Certificate Authority

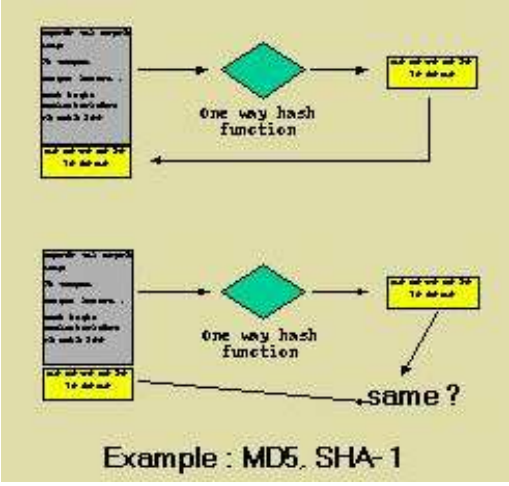
Penggunaan public key memang akan memudah proses manajemen key yang digunakan dalam suatu eCommerce. Tetapi bagaimana mengetahui suatu publik key adalah milik seseorang ? Untuk itu akan dimanfaatkan digital signature dan Certifcate Authority (CA).

Suatu fungsi hash pada dasarnya adalah suatu fungsi sederhana yang tak bersifat reversibel. Sehingga dengan mudah kita dapat menghasilkan suatu "signature" yang khas untuk tiap deretan data. Tetapi dari signature tersebut tak dapat dilakukan pembalikan untuk memperoleh deratan data asli.

Suatu CA akan mengikat (*bind*) suatu publik key dengan pemiliknya. Melakukan penyampulan untuk mendistribusikan publik key. CA yang dipercaya akan melakukan tanda tangan digital untuk menguji kepemilikan kunci tersebut. Suatu Certificate pada dasarnya akan berisi :



Gambar 8: One way hash function



Gambar 9: Pemanfaatan hash

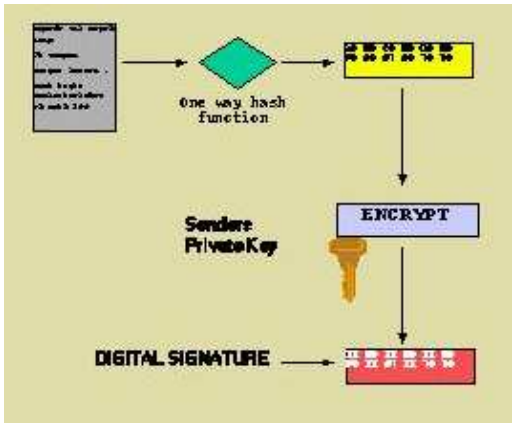
- Keterangan ditail tentang pemilik
- Keterangan tentang pihak yang mengeluarkan sertifikat (Certifier)
- Publik key itu sendiri
- Tanggal valid dan kedaluarsa
- Tanda tangan digital sertifikat tersebut yang dilakukan oleh CA
- Time stamp (penanda waktu)

Suatu CA akan melakukan beberapa hal mendasar :

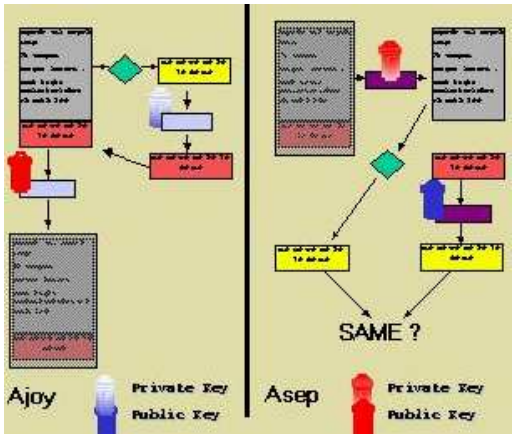
- Membuat sertifikat
- Bertanggung jawab memvalidasi pemilik dari suatu public key.
- Mendistribuskan CA dengan direktori server
- Membuat Certification Revocation List (CRL)

Biasanya CA disediakan oleh suatu institusi yang dipercaya oleh publik, misal suatu institusi pemerintah. Suatu Public Key Infrastructure akan terdiri dari :

- Certification Authority (CA)
- Registratraion Authority (RA)
- Direktori



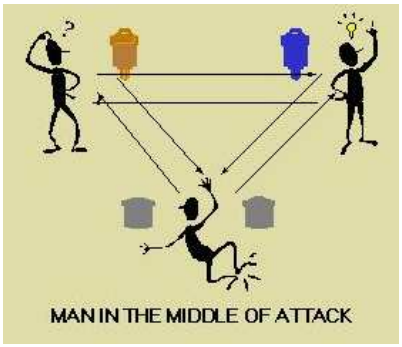
Gambar 10: Tanda tangan digital



Gambar 11: Mekanisme keseluruhan

- Aplikasi yang mendukung PKI
- Prosedur dan policy

Yang perlu dipahami adalah kenyataan bahwa biasanya dalam penyusunan PKI maka akan melibatkan **20% teknologi dan 80% policy**. PKI adalah salah satu infrastructure eCommerce yang penting.



Gambar 12: Pertimbangan serangan man in the middle

2.9 Audit dan monitor

Salah satu dari objektif sekuriti adalah accountability dan untuk mencapainya diperlukan mekanisme log (pencatatan) terhadap setiap akses / transaksi. Dalam melakukan pencatatan ini diperlukan informasi yang cukup untuk melakukan audit. Proses pencatatan ini mencakup informasi user-id (*who*), waktu akses dan durasi (*when*), tempat melakukan akses (*where*), objek yang diakses (*what*), dan aktifitas (*how*). Security indicents dapat dideteksi dengan menganalisa informasi dari hasil pencatatan tersebut (log). Untuk mencegah terjadinya security incidents, dapat dilakukan proses audit secara *real time* terhadap log-log tersebut. Untuk itu diperlukan arsip-log terpusat yang kemudian akan dianalisa dan diaudit secara *real time* sebagai proses security monitoring / surveilance yang terus-menerus (*continues*).

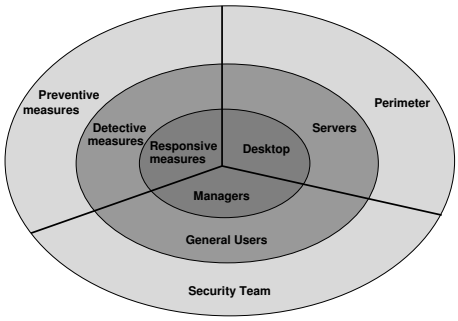
2.10 Pengguna, Security Policy, dan Manajemen

Pengguna seringkali menjadi mata rantai terlemah dalam sekuriti. Kesalahan dalam memilih password, kecerobohan pengguna dengan menuliskan password, dan mendownload file atau email bervirus sering menjadi penyebab utama security incident. Para cracker sering memanfaatkan ketidakpahaman pengguna akan sekuriti untuk mencari jalan masuk ke dalam sistem. Fenomena ini disebut dengan "social engineering" dalam teori sekuriti. Hal ini menjelaskan mengapa pengguna merupakan salah satu komponen penting dalam membangun sekuriti.

Untuk membatasi akses, mengatur pemakaian sumber daya, dan melindungi pengguna dan/atau sistem, perlu dibuat security policy yang berisi aturan-aturan komprehensif yang terdokumentasi dengan baik. Security policy harus diterapkan secara tepat dan menyeluruh kepada sistem dan penggunanya. Oleh karena itu security policy sudah seharusnya menjadi bagian dari strategi organisasi dan jajaran manajemen melakukan enforcement secara top-down dan pengguna melakukan proses feedback bottom-up untuk menyempurnakan security policy.

Pendekatan multidimensi dalam desain dan implementasi sekuriti saat ini sudah tak dapat ditawar lagi. Sebaliknya pendekatan tradisonal mulai ditinggalkan. Pendekatan multidimensi mencakup keseluruhan sumber daya, policy, dan mekanisme sekuriti yang komprehensif. Kunci dalam pelaksanaan sistem sekuriti model ini harus melibatkan keseluruhan staf dari semua jajaran dan area yang ada dalam organisasi tersebut. Tanpa pemahaman yang cukup dan kerjasama dari

semua pihak maka mekanisme sekuriti tersebut tidak dapat dilaksanakan dengan baik. Pendekatan multidimensi ini diketengahkan pada gambar 13.



Gambar 13: Enterprise IT security framework [20]

Untuk mendapatkan pertahanan yang kuat diperlukan sistem pertahanan bertingkat yang melibatkan policy dan teknologi. Secara konseptual pertahanan dapat dibagi menjadi tiga tingkat :

- **Perimeter**. Pertahanan yang terletak paling luar adalah perimeter dimana terdapat mekanisme firewall, mekanisme akses kontrol, proses autentikasi user yang memadai, VPN (virtual private network), enkripsi, antivirus, network screening software, real time audit, intrusion detection system, dan lain-lain. Pada tingkat pertahanan ini terdapat alarm yang akan menyala apabila terjadi serangan terhadap sistem
- **Servers**. Server merupakan entry-point dari setiap layanan. Hampir semua layanan, data, dan pengolahan informasi dilakukan di dalam server. Server memerlukan penanganan sekuriti yang komprehensif dan mekanisme administrasi yang tepat. Diantaranya adalah melakukan pemeriksaan, update patch, dan audit log yang berkala
- **Desktops**. Desktop merupakan tempat akses pengguna ke dalam sistem. Pengalaman telah menunjukkan bahwa kelemahan sekuriti terbesar ada pada tingkat desktop karena pengguna dengan tingkat pemahaman sekuriti yang rendah dapat membuat lobang sekuriti seperti menjalankan email bervirus, mendownload file bervirus, meninggalkan sesi kerja di desktop, dan lain-lain.

3 Ketergantungan sebagai ancaman sekuriti

Makin pentingnya eCommerce dan Internet, maka masalah sekuriti tidak lagi sekedar masalah keamanan data belaka. Berikut ini dikutipkan salah satu pernyataan Erkki Liikanen Commissioner for Enterprise and Information Society European Commission yang disampaikan pada **Information Security Solutions Europe** (ISSE 99), Berlin 14 October 1999. Berikut ini adalah cuplikan utama :

1. *Security is the key to securing users trust and confidence, and thus to ensuring the further take-up of the Internet. This can only be achieved if security features are incorporated in Internet services and if users have sufficient safety guarantees*
2. *Securing the Internal Market is crucial to the further development of the European security market, and thus of the European cryptographic industry.This requires an evolution of mentalities: Regulation in this field transcends national borders. Let's "think European".*

3. *European governments and the Commission now have a converging view on confidentiality. We see this in Council, in Member State policies and in the constructive discussions we have. We must take this debate further and focus of the potential of encryption to protect public security rather than mainly seeing it as a threat to public order.*

4. *Finally,* ***the promotion of open source*** *systems in conjunction with technology development is certainly one important step towards unlocking the potential of the desktop security market for the European cryptographic industry.*

Jadi masalah sekuriti pada infrastruktur eCommerce dan Internet tidak saja terletak pada masalah teknologi dan ekonomi saja, tetapi juga menyangkut dengan keamanan suatu negara atau ketergantungan negara terhadap negara lain. Bukan saja sistem sekuriti dengan teknologi yang aman, tetapi juga pertimbangan bahwa pemanfaatan suatu teknologi tidak dibatasi oleh negera lain. Sebagai contoh USA dengan ITAR-nya membatasi pemanfaatan jenis teknologi kriptografi tertentu.

3.1 Open Source sebagai solusi

Pemerintah Jerman melalui *Bundesminister für Wirtschaft und Technologi* (*BMWi*). menyatakan bahwa selama ini pengembangan infrastruktur Internet sering dilakukan dengan menggunakan pendekatan security through obscurity. Sehingga banyak orang menutup mata terhadap resiko yang mungkin terjadi pada sistem operasi yang dominan. Berdasarkan alasan inilah maka BMWi mendukung pengembangan Open Source, karena menjanjikan keamanan yang lebih baik. Paling tidak memungkinkan para ekspert di luar perusahaan penyedia sistem tersebut untuk memeriksa secara lebih seksama dan menyeluruh.

Dengan tersedianya source code para open source sering pihak merasa ragu akan keamanan sistem tersebut. Sudah barang tentu pendekatan dengan konsep *security through obscurity* ini kurang tepat. Pada saat ini Open Source merupakan salah satu kandidat untuk penyediaan infrastruktur sistem yang aman. Tidak saja aman dari sisi teknologi tapi juga dari sudut pandang ketergantungan suatu negara. Beberapa negara di Eropa telah memutuskan pemanfaatan Open Source dalam pembentukan infrastruktur eCommerce mereka.

BMWi sejak tahun 1999 telah mulai mengembangkan komponen untuk sistem sekuriti dengan perangkat lunak Open Source. Di samping itu, BMWi menganggap Open Source menawarkan solusi yang lebih aman, lebih user friendly dan inovasi yang lebih baik serta interoperabilitas yang baik dengan produk lain. Dengan ketersediaan source code maka diharapkan para developer di Jerman dapat bekerja lebih cepat tanpa bergantung pada vendor negara lain. Saat ini telah banyak developer Open Source yang berasal dari Jerman. Dukungan pemerintah Jerman terhadap Open Source memang sungguh-sungguh tercermin pada studi yang dilakukan *der Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung* (*KBSt*), yang menyarankan penggunaan perangkat lunak Open Source di lingkungan kementerian dalam negeri Jerman. Hal ini juga didukung oleh kajian *Institut für Rechtsfragen der Open Source Software* suatu LSM yang memfokuskan pada aspek hukum dari Open Source.

Negara-negara Eropa telah juga mengembangkan proyek yang dikenal dengan nama Interworking Public Key Infrastructure for Europe. Proyek ini juga berusaha mengembangkan teknologinya dengan pendekatan Open Source. Dengan telah diakuinya secara hukum tanda tangan digital ini, maka sudah saatnya Indonesia mempertimbangkan pembangunan PKI yang mempertimbangkan aspek non teknis dan teknis secara lebih seksama

3.2 WinBI mencoba memulainya

Internet telah memberikan dampak besar di dunia, sayangnya, masih terjadi jurang antara *Digital Have* dan golongan *Digital Have Not* , yang dikenal dengan istilah *Digital Divide* . Bahasa dan kultur juga dikenal sebagai penyebab timbulnya kondisi ini. Hasil survei *Computer literacy* yang dilakukan oleh BPPT pada tahun 2001, menunjukkan hal yang sama terjadi di Indonesia. Pengadopsian teknologi komputer dan informasi yang tersendat-sendat juga disebabkan karena bahasa pengantar yang digunakan dalam perangkat teknologi informasi tersebut. Sejak Internet

telah bergerak menjadi fenomena massal, maka bahasa dan kultur harus dipertimbangkan dalam mengembangkan pemanfaatan Internet.

Faktor lain yang juga diidentifikasi oleh survei BPPT itu adalah harga perangkat keras dan lunak. Sebagian besar negara Asia selalu menghadapi masalah pembajakan perangkat lunak yang disebabkan harga peragnkat lunak tersebut. Perangkat lunak Open Source dapat dipertimbangkan sebagai solusi dalam mengatasi masalah ini. Di samping itu, perangkat lunak Open source dapat disesuaikan untuk memenuhi kebutuhan lokal misal bahasa setempat dalam antarmuka pengguna (*user interface*), tanpa harus terlibat pada proses izin, lisensi yang menambah ketergantungan negara kita terhadap perusahaan asing.

Pengembangan **WINBI** (Window berbahasa Indonesia) merupakan suatu upaya yang difasilitasi oleh BPPT dan UGM yang melibatkan komunitas Open Source untuk mengembangkan suatu sistem operasi desktop berbahasa Indonesia. Sebelum adanya Open Source, usaha memiliki desktop berbahasa Indonesia seperti suatu usaha yang mustahil. Pengembangan sistem operasi desktop berbahasa Indonesia ini dimungkinkan diselesaikan dalam kurun waktu kurang dari 6 bulan karena WINBI menggunakan dasar sistem operasi Open Source (GNU/Linux - Trustix Merdeka). Usaha ini bersumber dari keinginan memberikan hak akses teknologi yang sama bagi masyarakat Indonesia secara luas, termasuk bagi mereka yang tidak bisa berbahasa Inggris.

WINBI menyediakan solusi sistem operasi desktop dengan aplikasi-aplikasi mendasar seperti aplikasi Internet, aplikasi perkantoran, aplikasi multimedia dan juga aplikasi hiburan. Semua antar muka (*user interface*) dan keterangan bantu (*Help*) tersedia dalam bahasa Indonesia. Sehingga orang yang tak memahami bahasa Inggris diharapkan dapat menginstal dan mengadministrasi sistem dengan lebih mudah. Telah dirilis secara bebas kepada publik dengan harapan agar publik dapat menggunakannya. Winbi didisain dan dikembangkan oleh pemerintah Indonesia melalui BPPT dan UGM. Proyek ini mengadopsi semangat Open Source. Diharapkan Winbi dapat memberikan kontribusi secara luas pada pengembangan TI di Indoensia pada umumnya, serta memenuhi kebutuhan masyarakat luas. Beberapa pertimbangan pada pengembangan Software RI adalah hal-hal berikut ini :

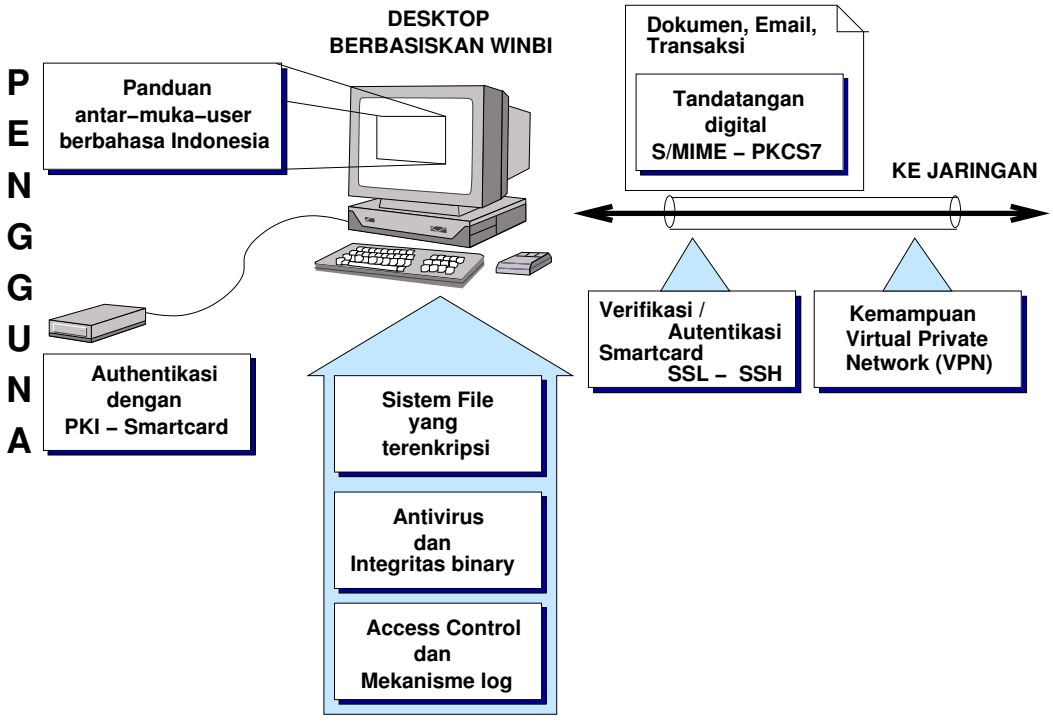
- **Bibit pengembangan di masa depan**. Software RI berupaya untuk menyediakan "bibit" untuk pengembangan solusi di Indonesia secara open source. Oleh karena itu disediakan source code dan dokumentasi.
- **Bisa terus berevolusi memenuhi kebutuhan masyarakat luas**. Diharapkan dengan berbasiskan lisensi Open Source, maka perbaikan dari satu pihak akan dapat dirasakan oleh semua orang (bila dibebaskan kembali). Model ini akan sangat efektif untuk pengembangan solusi yang didanai oleh pemerintah, dan melibatkan kepentingan publik.
- **Bisa dimanfaatkan masyarakat luas.** Berdasarkan survei komputer literasi yang dilakukan BPPT, masalah mahalnya harga perangkat lunak serta terbatasnya kemampuan bahasa Inggris dari masyarakat luas, menyebabkannya masih rendahnya penggunaan komputer secara efektif dan legal. Dengan adanya sistem perangkat lunak desktop berbahasa Indonesia dan berlisensikan bebas, diharapkan masyarakat luas dapat memanfaatkannya.
- **Bisa dimanfaatkan untuk mendorong bisnis lokal**. Kantaya dan WinBi itu adalah bibit yg diharapkan dapat dimanfaatkan oleh penyedia jasa "lokal" untuk menyediakan jasa kustomisasi, pelatihan, penerbitan buku lokal, ataupun kustomasiasi sistem komputer desktop.
- **Mendorong kemandirian di bidang TI**. Dengan berbasiskan perangkat lunak Open Source maka diharapkan WINBI dapat mendorong kemandirian masyarakat di bidang TI. Perusahaan di daerah tidak perlu bergantung kepada perusahaan di pusat yang kebetulan memiliki lisensi produk tertentu.

Di samping tim inti pengembang perangkat lunak ini, pengembangan mendapat dukungan oleh komunitas Open Source di Indonesia. Dalam hal informasi teknis pengerjaan, uji coba, kebutuhan, umpan balik serta beberapa informasi teknis yang telah dikembangkan oleh pengembang Open

Source Indonesia. Dukungan komunitas Open Source Indonesia dan internasional sangat berperan dalam kesuksesan proyek ini.

4 WinBI-NG, langkah selanjutnya

WinBI-NG merupakan suatu kerangka yang didusun untuk mengembangkan agara WinBI dapat memenuhi kebutuhan. WinBI-NG ini memiliki kerangka sebagai berikut :



Gambar 14: Kerangka Secure Desktop WinBI

- **Access Control List (ACL)**
  Kontrol akses file pada tradisional Unix, seperti juga pada Linux, merupakan Discretionary Access Control (DAC) yang menggunakan akses matrix user, group, dan others dengan atribut r(read), w(write), e(execute). Untuk membuat kontrol akses tersebut lebih fleksibel maka sebuah ekstensi mekanisme akses kontrol yang dispesifikasikan dalam POSIX.1e mulai diimplementasikan pada kernel versi terakhir. Ekstensi tersebut memungkinkan administrator mengaplikasikan Mandatory Access Control (MAC), mekanisme information label, bit capability, mekanisme audit yang lebih mapan, dan mekanisme sekuriti lainnya sehingga dapat menguatkan sistem kontrol akses di WinBI-NG.

- **Anti virus dan integritas file binary**
  Meskipun virus bukan merupakan masalah di Linux, kehadirannya tetap harus diwaspadai. Sharing program yang berisi trojan horse dalam suatu jaringan P2P atau sistem file yang terdistribusi, memungkinkan user secara tidak sengaja memasang virus tersebut di desktopnya. Permasalahan tersebut diharapkan dapat diantisipasi dengan beberapa metode : antivirus, pemeriksaan integritas binary, dan menerapkan mekanisme thin client pada suatu jaringan yang melibatkan banyak desktop. Proyek OpenAntiVirus <`http://www.openantivirus.com`> merupakan pilihan tepat untuk diimplementasikan dalam WinBI-NG. Proyek ini mengkombinasikan beberapa usaha untuk memindai dan membersihkan virus melalui file

sistem samba, email server, squid (proxy), dan program loader pada kernel. Pemeriksaan integritas file binary seperti pada Tripwire <http://www.tripwire.org> diperlukan untuk mendeteksi perubahan pada sistem akibat trojan horse maupun eksploitasi sistem. Selain itu, solusi thin client untuk jaringan dekstop mencegah terjadinya perubahan pada sistem desktop tanpa sepengetahuan administrator. Mekanisme ini memungkinkan image sistem disimpan pada sebuah server, sehingga setiap desktop tersebut dinyalakan maka akan terjadi penyegaran sistem desktop. Hal ini memungkinkan jika pada sesi terdahulu terjadi gangguan integritas file binary / konfigurasi sistem pada desktop, maka hal tersebut tidak akan terjadi pada sesi berikutnya. Pemanfaatan Linux BIOS <http://www.acl.lanl.gov> dan LTSP (Linux Terminal Server Project - <http://www.ltsp.org> merupakan solusi-solusi thin client yang akan dikembangkan dalam WinBI-NG selanjutnya.

- **Integrasi PKI dan Smartcard**
  Username dan password merupakan metode autentikasi yang kurang memadai karena hanya mengaplikasikan salah satu prinsip autentikasi (*what you know*). Oleh karena itu, Public Key Infrastructure (PKI) dan integrasinya dengan smartcard merupakan solusi yang bisa menjadi alternatif pengganti metode autentikasi tersebut. Private key dibangkitkan di dalam smartcard dan public key dimuat dalam bentuk sertifikat digital oleh sebuah Certification Authority (CA) yang dipercaya oleh sistem. Mekanisme "sign and challenge" serta verifikasi dengan sertifikat dijital yang dilakukan di dalam sistem dengan smartcard memungkinkan user hanya bisa melakukan akses ke dalam sistem keberadaan dengan smartcard (what you have). Tamper proof smartcard, mekanisme certificate revocation list (CRL), aktivasi smartcard dengan PIN (what you know), dan aktivasi smartcard dengan biometrik (who you really are) diperlukan untuk menjamin agar smartcard dan private key yang ada di dalamnya hanya dapat digunakan oleh pemiliknya. Aplikasi autentikasi, enkripsi, dan tanda tangan dijital melalui PKI dan smartcard ini sudah diaplikasikan pada console login (PAM), SSH (secure shell), dan TLS/SSL (Transport Layer Security/Secure Socket Layer) pada Mozilla / Netscape. Integrasi PKI dan smartcard ini dibahas pada satu bagian khusus dalam tulisan ini.

- **VPN pada lingkungan jaringan nirkabel**
  WEP (Wireless Encryption) pada Wireless LAN telah terbukti kurang memadai sebagai mekanisme pengaman jaringan nirkabel tersebut. Solusi yang cukup memadai adalah dengan menambah kemampuan desktop untuk menjalankan VPN (Virtual Private Network), sehingga akses desktop ke dalam jaringan pada lingkungan nirkabel hanya dapat dilaksanakan dengan VPN. Mekanisme autentikasi dan enkripsi pada VPN pada masa yang akan datang akan diintegrasikan dengan PKI dan smartcard untuk memperoleh sistem pengamanan yang end-to-end.

- **Sistem file yang terenkripsi**
  Pencurian data secara fisik, misalnya mengambil harddisk atau disket, merupakan hal yang sulit untuk dihindari. Oleh karena itu proteksi data pada media penyimpanan perlu dilakukan. Salah satu diantaranya adalah dengan menggunakan sistem file yang terenkripsi (encrypted file system). Sistem file yang terenkripsi di Linux dapat dilakukan dengan memuat partisi melewati device loop yang dimuat dengan pilihan pengunaan sebuah algoritma enkripsi yang didukung oleh kernel seperti twofish. Dalam penggunaannya perlu dipertimbangkan kemampuan CPU dan pilihan algoritma enkripsi karena proses ini akan cukup menyita sumber daya proses dan memori dalam sistem desktop.

- **Panduan dalam bahasa Indonesia**
  Pemahaman sebagian besar user terhadap bahasa ibu lebih tinggi daripada terhadap bahasa asing. Oleh karena itu panduan sekuriti dalam bahasa Indonesia merupakan salah satu cara untuk meningkatkan pemahaman user terhadap sekuriti. Kotak dialog koneksi SSL pada Mozilla adalah salah satu contoh target pengembangan dan penerjemahan dalam WinBIng selanjutnya. Keterangan dalam bahasa Indonesia dan penggunaan simbol yang tepat untuk

orang Indonesia diharapkan dapat mengingatkan dan meningkatkan kewaspadaan user sehingga insiden sekuriti akibat ketidakpahaman user dapat dikurangi. Disamping itu perlu diperhatikan tata letak pada antar muka user (user interface) sehingga user dapat dengan mudah memahami situasi dan tingkat keamanan pada saat melakukan transaksi. Pemilihan simbol yang tepat untuk merepresentasikan gagalnya verifikasi sertifikat dijital seperti gambar tengkorak berpendar atau gembok yang disilang dengan garis merah, mungkin merupakan simbol yang lebih menarik perhatian user dibandingkan gambar gembok atau kunci terpecah yang sering digunakan dalam program-program browser. Newsticker yang memuat berita-berita tentang sekuriti terbaru dapat dimanfaatkan untuk memperbaharui pengetahuan user terhadap perkembangan sekuriti terkini sehingga tingkat kewaspadaan user dapat terus dijaga.

- **Sistem audit dan ketersediaan log yang memadai**
  Log dan audit sudah merupakan keharusan dalam sistem server, tetapi sistem log dan audit pada desktop masih belum dianggap sebagai suatu hal yang penting. Akibatnya user tidak mempunyai alat bukti yang cukup ketika dihadapkan dengan suatu insiden yang melibatkan dirinya. Oleh karena itu pengembangan secure desktop memerlukan kapabilitas log yang baik, misal dengan memanfaatkan digital signature ataupun sistem log yang terdistribusi.

Gambar 15: Panduan sekuriti yang memadai dalam bahasa Indonesia

5 Pengaruh pengguna dalam keamanan

Manusia adalah salah satu faktor yang sangat penting tetap sering kali dilupakan dalam pengembangan Teknologi Informasi. Begitu juga dalam mengembangkan sistem sekuriti. Sebagai contoh karena penggunaan password yang sulit sehingga menyebabkan pengguna malah menuliskannya

pada kertas yang ditempel dekat komputer. Sehingga dalam menyusun kebijakan sekuriti faktor manusia dan budaya setempat haruslah sangat dipertimbangkan. Orang/pengguna merupakan sisi terlemah dari sekuriti. Mereka tak memahami komputer, mereka percaya apa yang disebutkan komputer. Mereka tak memahami resiko. Mereka tak mengetahui ancaman yang ada. Orang menginginkan sistem yang aman tetapi mereka tak mau melihat bagaimana kerja sistem tersebut. Pengguna tak memiliki ide, apakah situs yang dimasukinya situs yang bisa dipercaya atau tidak

Salah satu permasalahan utama dengan user di sisi sekuriti, adalah akibat komunikasi atau penjelasan yang kurang memadai pada user dan disain yang kurang berpusat pada user yang mengakibatkan lemahnya sekuriti (Adams dan Sasse, 1999). User seringkali tak menerima penjelasan yang cukup, sehingga mereka membuat atau mereka-reka sendiri resiko atau model sekuriti yang terjadi. Seringkali ini menimbulkan pengabaian dan mengakibatkan kelemahan sekuriti.

Di samping itu, akibat pengabaian para pendisain sistem terhadap perilaku user dalam berinteraksi terhadap sistem, maka timbul kesalahan misalnya adanya pengetatan yang tak perlu, yang malah mengakibatkan user mengabaikan pengetatan itu. Atau penyesuaian kecil yang seharusnya bisa dilakukan untuk menambah keamanan, tetapi tak dilakukan. Sebagai contoh *layout page* tidak pernah mempertimbangkan sisi sekuriti, ataupun belum ada desain layout yang meningkatkan kewaspadaan pengguna akan keamanan. Disan halaman Web lebih ditekankan pada sisi estetika belaka. Untuk itu sebaiknya dalam disain sistem, user diasumsikan sebagai pihak yang memiliki kewaspadaan terendah, yang mudah melakukan kesalahan. Artinya pihak perancanglah yang mencoba menutupi, atau memaksa si user menjadi waspada.

Beberapa langkah yang perlu dilakukan oleh penyedia layanan dalam merancang sistem yang berkaitan dengan sisi pengguna adalah :

- **Sekuriti perlu menjadi pertimbangan yang penting dari disain sistem**. Memberikan umpan balik pada mekanisme sekuriti akan meningkatkan pemahaman user terhadap mekanisme sekuriti ini.
- **Menginformasikan user tentang ancaman potensial pada sistem**. Kepedulian akan ancaman ini akan mengurangi ketakpedulian pengguna terhadap ditail langkah transaksi yang dilakukan. Memang para pengguna Internet di Indonesia kebanyakan memiliki kendala dalam hal **bahasa** . Sehingga mereka sering melewati dan tak membaca pesan yang tampil di layar. Hal ini menuntut semakin perlunya menu dan keterangan berbahasa Indonesia pada**.**
- **Kepedulian user perlu selalu dipelihara**. Secara rutin penyedia layanan harus memberikan jawaban terhadap pertanyaaan masalah sekuriti, baik yang secara langsung maupun tidak
- **Memberikan user panduan tentang sekuriti sistem , termasuk langkah-langkah yang sensitif.** Sebaiknya ketika user baru memulai menggunakan suatu layanan, mereka telah di-"paksa" untuk membaca petunjuk ini terlebih dahulu.

Seringkali Graphical User Interface (GUI) yang hanya memfokuskan pada kemudahan pengoperasian dan interaksi, sering mengakibatkan sistem menjadi tidak aman. Atau mendorong pengguna kurang memperhatikan masalah keamanan. Contoh yang banyak terjadi adalah kemudahan program pembaca email yang menyebabkan pengguna meng-klik email dan secara tidak sadar menjalankan program virus.

Pada dasarnya seorang pengguna memiliki tanggung jawab penggunaan sumber daya komputasinya. Tanggung jawab ini berdasarkan konvensi yang berupa (Ladkin, 1999) :

- **Legal**, sebagai contoh tak mengancam orang lain, tak menyaru sebagai orang lain, jangan merusak pekerjaan orang lain
- **Kontraktua**l, sebagai contoh tak bermain game, tak menulis email pribadi, menjaga kontrak bisnis tetap bersifat rahasia.

- **Sosial**, tak menunjukkan gambar porno, atau tak membaca email milik orang lain.

Berdasarkan pertimbangan keaman sistem, maka User Interface perlu dikembangkan pada WinBI memiliki pertimbangan berikut ini:

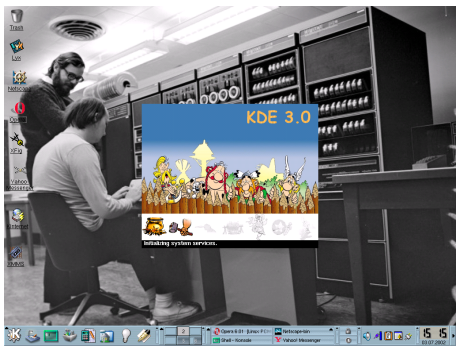
- UI tersebut mampu mendorong awareness pengguna bila perlu setengah memaksa agar pengguna melakukan aksi yang aman. Memang selalu ada pertimbangan antar kemudahan interaksi dan keamanan.
- UI tersebut mendorong pengguna fokus kepada faktor keamanan sebelum melakukan suatu aksi (misal menjalankan program, mendownload sesuatu, koneksi ke URL tertentu). Simbol yang digunakan untuk mengisyaratkan hal tertentu harus dipahami oleh pengguna.
- UI yang memudahkan pengguna memanfaatkan fitur sekuriti. Sebaiknya disediakan fitur yang memudahkan pengguna mengatur fitur seperti enkripsi, VPN dan lain sebagainya. Sehingga karena fitur ini mudah dikonfigurasikan dan digunakan, maka pengguna akan memanfaatkannya.
- UI yang simbol dan kalimatnya dipahami oleh pengguna sehingga tidak menimbulkan interpretasi yang kurang terhadap peringatan yang diberikan oleh sistem.

WinBI telah memiliki beberapa aplikasi dasar yang dapat dimanfaatkan untuk memenuhi hal tersebut.

5.1 Menaikkan kepedulian pengguna

Ketika user mulai menjalankan Graphical User Interface (GUI), dalam hal ini KDE, akan disajikan suatu layar tampilan yang lazim disebut dengan *splash screen* . Pada KDE aplikasi yang bernama **ksplash** yang akan tampil ketika proses ini. Aplikasi ini menampilkan tahapan-tahapan mulai dijalankannya sistem GUI KDE.

Tampilan pada splash screen dapat digunakan untuk menyajikan informasi-informasi tambahan berkaitan dengan sistem operasi ataupun beberapa hal yang dapat mengingatkan pengguna akan faktor sekuriti. Misal informasi terbaru soal exploit ataupun mengingatkan pengguna untuk mengganti password. Dengan penambahan ini maka anda memberikan suatu fungsi kosmetik yang bersifat meninggikan *usabilitas hedonis*. Untuk melihat contoh perubahan splash screen, beberapa *patch* (tambalan) telah tersedia untuk berbagai splash screen ini. Misal seperti yang tampak pada gambar berikut ini, suatu splash screen yang menggunakan tema Asterix dan Obelix:



Gambar 16: Ksplash untuk memberi pesan kepada pengguna

Untuk memberikan informasi baru ataupun tip kemanan bagi pengguna, maka aplikasi **Ktip** dapat dimanfaatkan. Biasanya setiap kali pengguna masuk ke GUI maka ktip ini akan dijalankan. Sehingga dapat diberikan informasi bagi pengguna, misal mengingatkan pengguna untuk mengganti password, atau untuk menggunakan tanda tangan digital.

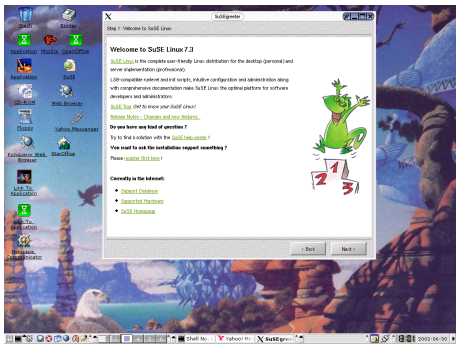
Gambar 17: Ktip untuk meningkatkan kepedulian

Dengan cara memaksa pengguna membaca informasi ini di depan mata, maka diharapkan kesadaran pengguna akan bahaya seperti mail dengan attachment, plug-in berbahaya, virus atau lain halnya dapat diminimalkan. Karena kesadaran pengguna akan keamanan dapat ditingkatkan secara perlahan.

5.2 Wizard untuk menambah aman

Pada sistem KDE di WinBI telah disediakan beberapa utilitas pengaturan sistem, antara lain **ksysv**, **kuser** ,**kpackage** ,**kcron kwuftpd**. Ataupun pada menu KDE telah disediakan semacam Control Panel yang dapat digunakan melakukan konfigurasi sistem.

Sayangnya utilitas konfigurator tersebut bekerja di aras bawah. Sebagian besar belum dilengkapi dengan *wizard* ataupun template model pengaturan yang biasa digunakan. Beberapa distro telah mulai menerapkan utilitas yang membantu pengguna mengatur sistemnya secara mudah. Sebagai contoh pada distro SuSE Linux hal itu dilakukan oleh utilitas program bernama **susewin** yang berfungsi sebagai "*wizard* ". Program ini akan menuntut pengguna langkah demi langkah untuk melakukan konfigurasi.

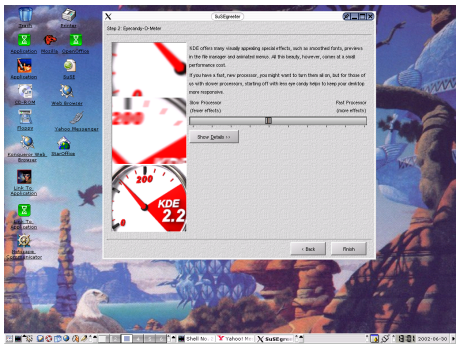


Gambar 18: Layar perkenalan dapat digunakan untuk mengingatkan pengguna

Dengan memberikan wizard-wizard tambahan tersebut, maka pengguna dapat secara lebih mudah mengkonfigurasi sistemnya sehingga dapat lebih meningkatkan keamanan sistem.

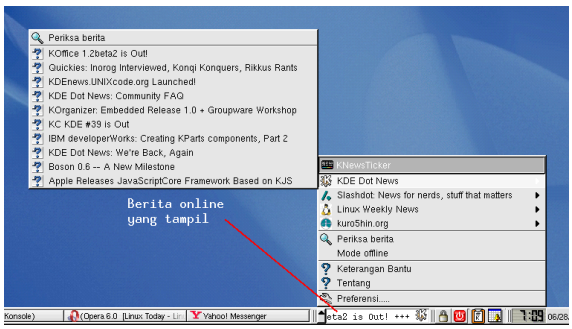
5.3 Sekuriti adalah proses

KDE yang digunakan WINBI telah menyediakan fasilitas untuk menampilkan berita secara langsung (*ticker*). Aplikasi tersebut adalah **knews** . Aplikasi ini merupakan *applet* yang bisa di-*dock*



Gambar 19: Wizard konfigurasi

(ditempelkan) pada desktop). Dan akan menampilkan berita berjalan otomatis. Pada gambar berikut ini berita akan berjalan dari kanan ke kiri. Secara otomatis sesuai waktu yang ditentukan, desktop akan mengontak situs dan menurunkan berita. Pengguna tak perlu mendownload *head line* berita secara manual. Dengan cara ini berita sekuriti terbaru, ataupun masalah sekuriti terbaru dapat diinformasikan kepada pengguna. Dengan cara ini kepedulian pengguna dapat selalu dijaga.



Gambar 20: Aplikasi knews

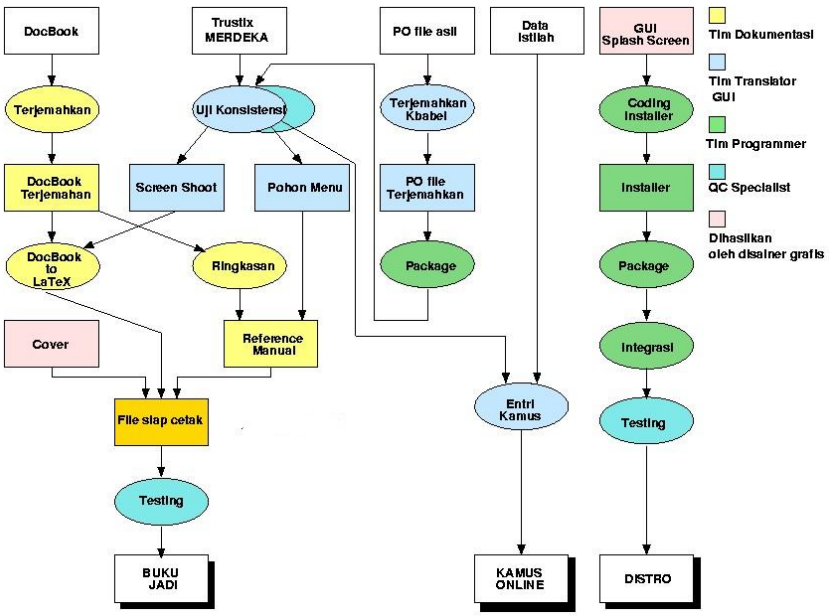
6 Penerjemahan GUI

Berbagai upaya penerjemahan program-program open source ke dalam bahasa Indonesia telah dilakukan oleh para sukarelawan, dan hasilnya sudah mulai terlihat nyata, misalnya dengan munculnya sistem operasi Trustix Merdeka[2] dan sistem operasi WinBI[3]. Penerjemahan GUI ini sepeti yang diungkapkan pada bahasan sebelumnya dapat meningkatkan keamanan sistem desktop. Terutama kepada pengguna yang tidak memahami bahasa Inggris secara baik. Informasi yang lebih ditail yang disampaiakn dalam bahasa Indonesia akan elbih mudah dipahami, sehingga pengguna tidak langsung melakukan aksi sebelum membaca pesan dari GUI.

6.1 Proses penerjemahan

Berikut ini adalah gambar proses penerjemahan yang dilakukan tim pengembang **WinBI** dalam melakukan penerjemahan dan pembuatan distro WinBI. Distro WinBI ini berbasiskan **Trustix**

[2]http://merdeka.trustix.co.id
[3]http://www.software-ri.or.id/winbi/



Gambar 21: Alur pekerjaan

Merdeka. Trustix Merdeka yang digunakan sebagai basis telah memiliki beberapa komponen berbahasa Indonesia seperti instalasi modus teks, man-page, menu KDE. Sehingga sebelum memulai proses penerjemahan, maka dilakukan inventarisasi terlebih dahulu apa yang telah ada pada Trustix Merdeka. Urutan pengerjaan WinBI ini ditampilkan pada Gambar 21 dan dapat dijelaskan sebagai berikut :

1. Tim terdiri dari kelompok penerjemah Berkas PO (untuk menu), penerjemah Help (berkas DocBook), pemrogram, dan disainer grafik. Di samping penguji kualitas, dan koordinator.

2. Tim penerjemah PO dibagi dua, satu tim menerjemahkan berkas PO yang belum diterjemahkan pada Trustix Merdeka. Kelompok lainnya memeriksa hasil penerjemahan Trustix Merdeka dan menyusun suatu daftar terjemahan yang akan digunakan. Daftar ini akan digunakan untuk proses penerjemahan. Berkas PO diedit dengan menggunakan perangkat lunak bantu **KBabel**.

3. Karena berkas Help belum ada yang diterjemahkan maka tim penerjemah berkas Help dapat langsung memulai. Berkas Help ini tertulis dalam format DocBook. Untuk menjaga konsistensi, maka pada tahapan ini istilah asing tidak diterjemahkan terlebih dahulu (sehingga mudah melakukan *Search - Replace*). Untuk mengedit berkas DocBook dapat digunakan sebarang editor teks atau Emacs dengan modul tambahan untuk mengedit DocBook.

4. Tim programmer dan grafik bisa mulai bekerja melakukan tugasnya. Pada program installer terdapat berkas PO untuk menu, dan berkas DocBook untuk online Help-nya. Berkas ini diberikan pada tim penerjemah PO dan tim penerjemah DocBook. Sedangkan pendisain grafik juga memulai mendisain gambar-gambar untuk proses instalasi, gambar yang didisain disesuaikan dengan pesan yang ingin disampaikan pada pengguna ketika proses instalasi. Pendisain grafik juga mulai mendisain sampul buku, dan class LaTeX yang digunakan juga didisain dari awal. Sengaja dipilih menggunakan LaTeX agar konsistensi dokumen yang dihasilkan bisa terjaga.

5. Ketika proses penerjemahan selesai maka dilakukan pemeriksaan konsistensi dari semua berkas PO. Semua frase yang digunakan, istilah dikumpulkan pada berkas lembar data

(*spread sheet*) yang juga dimasukkan ke dalam pangkalan data *database*). Dengan membandingkan istilah terjemahan tim INPRES 02, dan istilah lainnya, maka dipilih istilah yang paling tepat yang digunakan WINBI. Tentu saja tidak hanya masalah tata bahasa yang difikirkan tapi juga masalah lainnya. Kunci yang terpenting adalah usabilitas dari penerjemahan itu. Jangan sampai hanya mengejar benar/tidaknya dari tata bahasa, tetapi menjadi terlalu panjang dan tidak cukup di GUI, atau istilahnya menjadi rancu dengan istilah yang biasa digunakan.

6. Setelah ditentukan istilah yang tepat, konsisten dan sesuai dengan pertimbangan pemanfaaatan, maka dilakukan pemeriksaan konsistensi kembali ke semua berkas PO, dan juga dilakukan proses penggantian istilah di berkas online Help yang tadinya masih berbahasa Inggris. Ini dilakukan untuk semua berkas PO dan semua berkas DocBook. Jadi minimal melalui 2 kali siklus penerjemahan.

7. Setelah proses penerjemahan dokumentasi stabil (tidak berubah lagi), maka dilakukan pengambilan gambar *screen shot* yang akan digunakan untuk dokumentasi atau berkas Help. Untuk proses instalasi, membutuhkan perubahan program instalasi **anaconda.** Sedangkan untuk kebutuhan dokumentasi semua menu dibuat grafik pohonnya dengan menggunakan program **graph-viz** .

8. Setelah proses penerjemahan selesai, maka dilakukan proses pembuatan distribusi (ISO). Bagaimana proses pembuatan distro dan image CD ini tidak dijelaskan dalam tulisan ini. Tentu saja setiap pembuatan image CD, akan dilakukan uji coba instalasi sistem, apakah telah memenuhi formulir pengujian yang telah disiapkan. Misal apakah menu telah berbahasa Indonesia, apakah pilihan bahasa Indonesia telah ada dan lain sebagainya.

9. Sedangkan dari berkas DocBook untuk online Help dilakukan konversi menggunakan db2latex agar menjadi berkas LaTeX yang akan dikonversi ke LyX untuk diedit dengan menggunakan LyX (sebetulnya ada db2lyx, tetapi hasilnya kurang begitu bagus). Setelah mengalami pengeditan (misalnya menyatukan aplikasi yang sejenis, menyingkat informasi yang terlalu ditail) maka dapat dihasilkan buku petunjuk penggunaan dengan hasil akhir adalah berkas PostScript dan PDF yang siap cetak.

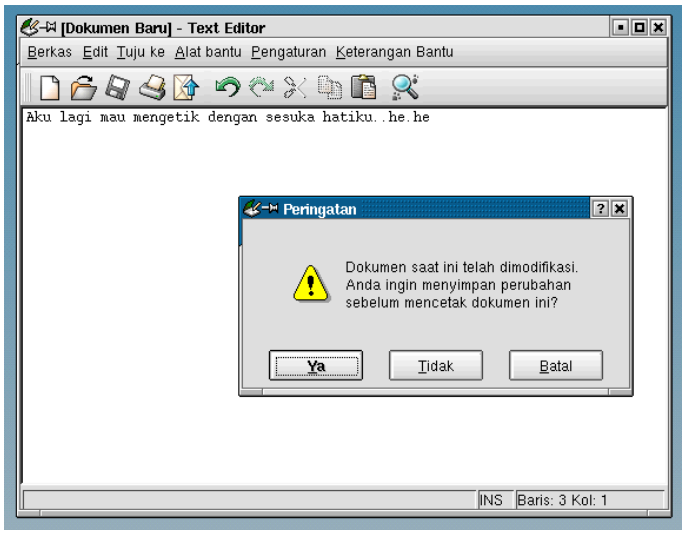
10. Untuk menghasilkan berkas online di Web, dengan format HTML, maka dapat digunakan konversi DocBook ke HTML langsung (seperti yang dimasukkan dalam online Help dari KDE. Berkas DocBook sendiri dalam pembuatan distro akan dikonversi ke HTML. Jadi relatif proses ini telah dilakukan. Salah satu keuntungan dengan menggunakan format seperti DocBook ini adalah proses konversi ke berbagai format dapat dilakukan secara mudah dan konsisten.

6.2 Kompleksitas penerjemahan

Pekerjaan menerjemahkan suatu distro sehingga menjadi distro berbahasa Indonesia adalah pekerjaan mudah-mudah sulit, tetapi jelas jauh dari mudah. Pekerjaan menerjemahkan itu sepertinya tidak membutuhkan pengetahuan teknis, dan sepertinya tidak sulit. Memang kalau menerjemahkan satu program saja kesulitan itu belumlah terasa. Tetapi bila sudah melibatkan komponen-komponen lengkap suatu paket distro lengkap yang tersebar lebih dari **2000 berkas**, maka membutuhkan beberapa hal yang perlu diterjemahkan misal :

- **Komponen User Interface**. Hal ini meliputi, menu, tombol, pilihan pada menu, judul *Window* dan beberapa komponen user interface lainnya. Dalam lingkungan GNU/Linux biasanya hal ini dilakukan dengan menerjemahkan berkas **PO** . Beberapa hal mendasar yang perlu diterjemahkan secara konsisten adalah komponen GUI seperti yang tertera pada **Visual Dictionary** dari tim KDE.

- **Warning message** (peringatan, dan pesan kesalahan). Kalimat terjemahan yang berupa peringatan atau pesan ini jelas harus diterjemahkan dengan konteks yang tepat. Jangan sampai malah pengguna mengabaikan atau takut dan akhirnya tak menggunakan sistem. Kasus KlikBCa yang diakibatkan peringatan tertulis dalam bahasa Inggris (ketika meng-*Accept* sertifikasi digital) tidak dipahami oleh pengguna. Jadi tidak saja kita mempertimbangkan dari sisi bahasa, tapi juga dari sisi manfaat dan "rasa" pesan itu. Belum lagi ketika suatu pesan itu melibatkan keterangan terhadap komponen lain dari program, maka istilah yang digunakan harus tepat. Misal ada harus keluar pesan "Harap anda pilih Menu **Berkas** - **Buka** ". Bayangkan kalau nama menu yang digunakan pada GUI tidak sama dengan yang ditulis di pesan ini. Pengguna bukannya terbantu tetapi malah menjadi bingung.



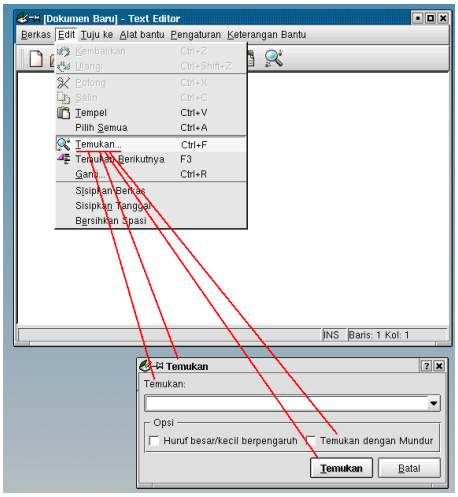
Gambar 22: Konsistensi pesan dan komponen GUI

- **Online Help**. Suatu perangkat lunak biasanya memiliki keterangan bantu yang bersifat *on-line* . Penerjemahan keterangan bantu ini harus mempermudah pengguna, dan jangan hanya memperhatikan faktor benar/tidaknya dari sisi tata bahasa saja. Di samping itu, faktor format berkas yang digunakan harus juga diperhatikan. Pada sistem GNU/Linux seperti WinBI ini biasanya ada beberapa format dan perangkat bantu untuk keterangan bantu On-line misalnya :
  - **man** menggunakan format groff.
  - **info** menggunakan format info.
  - **KDE help** menggunakan format HTML yang berasal dari DocBook (SGML/XML)
  - **Installer anaconda** menggunakan format HTML yang berasal dari DocBook (SGML).
- **Dokumentasi**. Ketika program diberikan ke masyarakat luas, tentunya sebaiknya disertakan dengan petunjuk penggunaan. Buku petunjuk penggunaan ini akan memudahkan pengguna memakai sistem. Pada pekerjaan penerjemahan dokumentasi ini, istilah yang digunakan di buku petunjuk penggunaan harus sama dengan istilah yang ditampilkan oleh program. Agar memudahkan proses ini, maka di dalam pengerjaan WINBI dimanfaatkan konversi dari DocBook ke LaTeX, **db2latex**, yang merupakan kumpulan XLST.
- **Situs Web**. Seperti biasanya pada saat ini, maka dokumentasi program WINBI nantinya akan disediakan sebagai artikel online di situs WINBI. Tentu saja akan menggunakan format HTML, dan agar tidak membingungkan maka konsistensi isi dan istilah dari dokumen

di program dan di situs harus dijaga. Jelas ini menimbulkan masalah teknis dan koordinasi, karena yang mengerjakan adalah orang yang terpisah. Untungya program KDE menggunakan DocBook yang memudahkan pekerjaan ini. Sehingga berkas DocBook dapat dengan mudah dikonversi menjadi HTML.

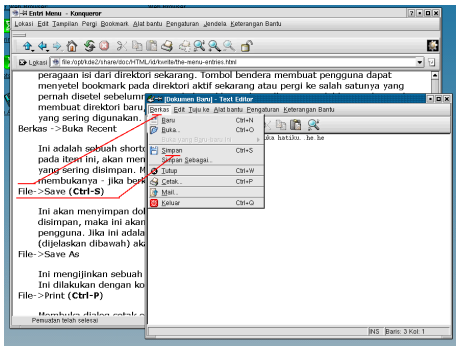
Dari penjelasan pekerjaan di atas, mulai nampak di manakah keruwetan dan kompleksitas proses menerjemahkan suatu distro. Permasalahan ini bukan saja timbul dari sisi bahasa tetapi juga dari sisi teknologi. Beberapa permasalahan yang timbul antara lain :

- **Menjaga penerjemahan yang baik**. Dalam hal ini bukan saja kaidah bahasa Indonesia yang harus dipenuhi, tetapi juga perlu dijaga agar, istilah tersebut tidak menjadi terlalu *aneh* atau jauh dari istilah yang biasa digunakan di pasar (terutama yang menyangkut nama perangkat keras).
- **Menjaga konsistensi penerjemahan pada satu program**. Pada satu program beberapa "terminologi" terjemahan digunakan beberapa kali (dalam satu program ada beberapa frase yang mengacu ke kata *File* atau *Find* . Misal, kata *Temukan* yang merupakan terjemahan dari *Find* muncul pada berbagai komponen menu, yaitu pada nama window, *check-box* , dan pada label. Kata *Temukan* tersebut adalah terjemahan dari *Find* yang digunakan haruslah sama, jangan berbeda-beda, agar tidak membingungkan. Jangan sampai di satu bagian menggunakan istilah *floppy*, tetapi di bagian lain menggunakan istilah *disket*. Jadi bukan saja benar atau tidaknya suatu terminologi tetapi konsisten, dan tepat atau tidaknya suatu terminologi. Seringkali kata yang sama dalam bahasa Inggris dapat memiliki kata padanan yang berbeda dalam bahasa Indonesia, sebagai contoh kata *default* . Hal ini menjadikan pertimbangan konteks yang lain lagi.



Gambar 23: Menjaga konsistensi terjemahan pada program yang sama

- **Menjaga konsistensi penerjemahan antara program dan online Help**. Seringkali suatu teks pada keterangan bantu (*online help*) mengacu ke suatu komponen GUI, misal kalimat "*Push the Cancel button* " tentunya harus diterjemahkan menjadi "*Tekan tombol* ***Batalkan*** . Sehingga kalimat pada keterangan bantu yang digunakan harus mengacu dengan benar kepada istilah yang ada di GUI hasil terjemahan. Misal di menu program telah tertulis **Berkas -> Simpan** , sedangkan pada keterangan help masih tertulis **File -> Save** . Hal ini adalah kesalahan yang sering timbul, ketika kelompok yang terpisah yang mengerjakan penerjemahan.



Gambar 24: Konsistensi help dan menu

- **Menjaga konsistensi penerjemahan antara program, online help dan buku manual**. Di antara ketiga informasi bantu ini, haruslah terjaga konsistensi, istilah dan hal yang diterjemahkan. Jangan sampai di program disebutkan A, tetapi di buku manual disebutkan B. Hal ini akan membingungkan pengguna. Pekerjaan menjaga konsistensi ketiga komponen yang terpisah ini tidaklah mudah, apalagi bahasa Indonesia sangatlah bersifat kontekstual.

- **Menjaga konsistensi penerjemahan antar program-programpada distro tersebut.** Pada satu distro terdapat begitu banyak program. Misal program untuk desktop (KDE), program aplikasi KOffice, program browser Konqueror, dan sebagainya (**lebih dari 100 program**). Tentu saja harus dijaga jangan sampai istilah yang digunakan (terutama untuk komponen GUI, menu) berbeda. Misal jangan sampai di Koffice menggunakan *Berkas* sedangkan di KWrite menggunakan *File* . Seperti pada Gambar 4., program KWrite dan KIcon menampilkan menu yang harus menggunakan terjemahan yang sama untuk hal yang sama. Tentu saja bila 1 atau 2 kalimat itu tak menjadi masalah, tetapi kalau sudah satu paket distro menjadi lain lagi ceritanya. Apalagi terkadang komponen GUI itu ada di bagian komponen program lainnya.

- **Menjaga agar penerjemahan cocok dengan ukuran GUI**. Seringkali kata hasil terjemahan dalam bahasa Indonesia memiliki jumlah huruf yang jauh lebih banyak dari kata aslinya. Sebatai contoh *Help* bisa diterjemahkan menjadi *Bantuan* atau *Keterangan bantu* . Panjang kata ini sering menimbulkan permasalahan, hendak dipotong menjadi ganjil, tidak dipotong tidak sesuai dengan GUI.

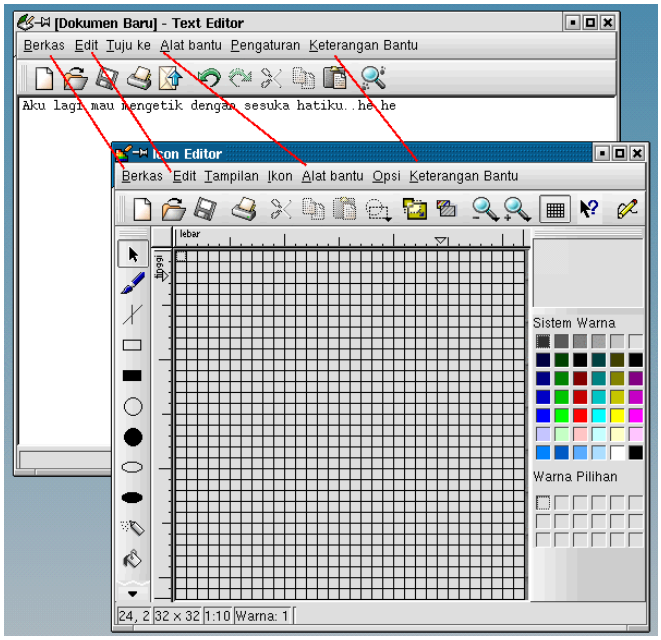
- **Tombol penyingkat** (accelerator button). Biasanya suatu menu memiliki tombol penyingkat, untuk memudahkan memanggil menu, misal New sering menggunakan Ctrl-N. Tentu saja menjadi ganjil bila telah diterjemahkan menjadi Baru tetapi tombol penyingkat tetap Ctrl-N. Jelas ini harus disesuaikan juga (biasanya mencoba menggunakan huruf pertama). Di samping itu, pada satu program juga harus diperhatikan, jangan sampai ada tombol penyingkat yang bentrok (digunakan di dua bagian menu yang berbeda). Pada KDE telah disediakan fasilitas untuk mendeteksi terjadinya tombol penyingkat yang bentrok ini.

6.3 Solusi penerjemahan di WinBI-NG

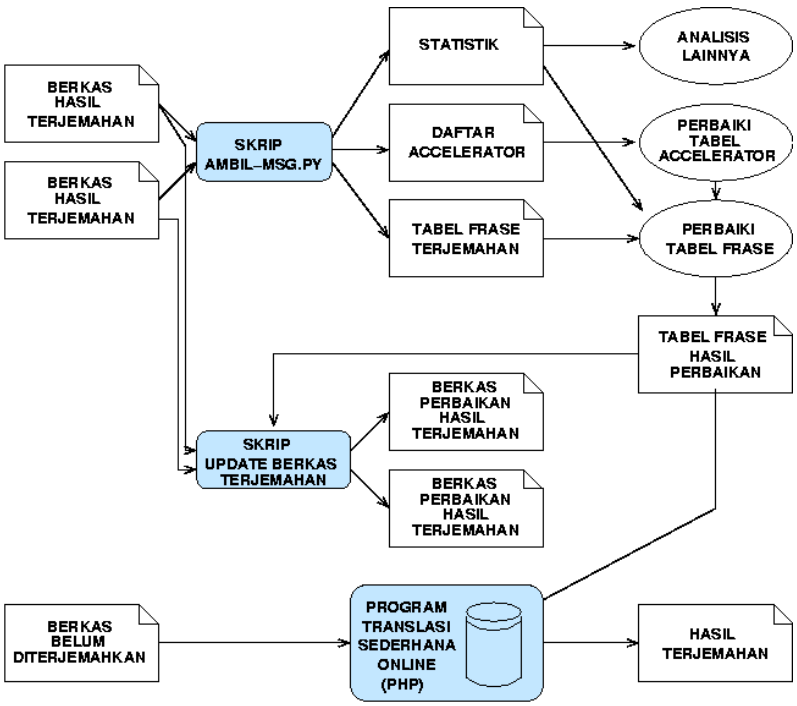
Untuk memudahkan pekerjaan penerjemahan di masa mendatang, maka dikembangkan perangkat bantu ANOA.

Program ini akan memberikan manfaat :

- Mempercepat proses translasi, bagi berkas po yang masih kosong dapat menggunakan fasilitas penerjemahan sederhana online. Sehingga diperoleh frase-frase terjemahan berdasarkan



Gambar 25: Konsistensi antar program



Gambar 26: ANOA untuk memudahkan penerjemahan

contoh-contoh frase yang ada. Untuk beberapa frase yang kontekstual satu kalimat akan diberikan beberapa kemungkinan penerjemahan. Penerjemah hanya perlu memilih sesuai konteknya. Hal ini memudahkan untuk menjaga konsistensi antar program.

- Memudahkan pemeriksaan konsistensi frase terjemahan. Sebab semua frase akan dikelompokkan menjadi satu
- Memudahkan proses pembaharuan penerjemahan (bila tabel frase berubah). Dengan program ini secara otomatis berkas po yang terkait akan diubah.

Alur kerja dari sistem ini adalah sebagai berikut :

- Dari berkas-berkas po yang telah ada (misal dari WinBI), diberikan pada skrip `ambil-msg.py` yang akan mengumpulkannya dan menyimpan semua frase asli dan frase terjemahan menjadi 1 berkas. Dalam berkas ini akan disort sehingga dengan mudah diperiksa apakah suatu frase yang sama diterjemahkan menjadi frase yang sama pada berkas yang berbeda.
- Dari berkas po tersebut juga dijalankan program `ambil-axe.py` untuk mengumpulkan tombol penyingkat (*accelerator key*). Dengan cara ini kita dapat memeriksa apakah tombol yang sama digunakan untuk semua program di lingkungan KDE.
- Hasil dari `ambil-msg.py` akan dimasukkan ke dalam suatu database back-end oleh skrip `taruh-msg.py`.
- Sedangkan program `stat-po.py` digunakan untuk memperoleh statistik dari penerjemahan.
- Ketika tabel frase diperiksa dan diperbaharui, maka dapat dilakukan proses perbaikan pada berkas po yang terkait. Hal ini dapat dilakukan secara otomatis dengan menggunkan program `gsm-libma.py`. Program ini akan memperbaharui berkas po yang mengalami perbaikan penerjemahan frase. Dengan cara ini, maka konsistensi antara tabel frase dan tiap berkas po dapat terjaga.
- Tentu saja masih dibutuhkan orang yang memeriksa penerjemahan frase ini. Dalam memeriksa frase juga perlu dipertimbangkan lebar karakter dari tiap entri menu.
- Di masa mendatang maka penerjemah lain dapat melakukan penerjemahan kasar melalui suatu situs dengan cara memberikan berkas po yang masih kosong. Dengan melihat ke database, maka sistem akan memberikan berkas po yang telah diterjemahkan (berdasarkan tabel frase).

Dengan cara ini maka dapat lebih mudah dilakukan penerjemahan secara konsisten di masa mendatang

ambil-msg.py

Program `ambil-msg` berfungsi untuk mengumpulkan semua kalimat terjemahan ke dalam sebuah file yang mudah diakses. Masih panjang jalan yang harus ditempuh oleh program ini agar dapat lebih berguna dalam mendukung upaya penerjemahan berbagai software open source ke dalam bahasa Indonesia. Program ini dikembangkan dengan menggunakan bahasa Python dan memiliki lisensi GPL. Pada versi terakhir ini (0.1.2), program `ambil-msg` telah memiliki fitur-fitur sebagai berikut :

- Kemampuan memproses file-file PO yang ada di dalam suatu direktori secara rekursif
- Kemampuan memproses PO bentuk plural
- Kemampuan menulis hasil ke dalam suatu file output dengan nama yang diberikan oleh user

- Kemampuan mendeteksi dan menangani bila file output sudah ada
- Berukuran kecil (di bawah 10KB)
- Dukungan untuk command line option
- Kemampuan untuk menyimpan panjang frase tiap message
- Kemampuan untuk mengacuhkan accelerator key

Untuk saat ini, kami beranggapan bahwa fitur yang disediakan oleh `ambil-msg.py` sudah cukup, dan kami tidak lagi akan menambahkan fitur-fiturnya dulu (*feature freeze*), karena kami sedang mengembangkan program-program lainnya. Namun demikian berikut ini adalah beberapa ide fitur untuk versi mendatang yang akan kami tambahkan ke `ambil-msg.py` :

- Dukungan untuk menyimpan output ke berbagai format, misalnya text, MS Excell, KSpread.
- Kemampuan untuk mengkompresi outputnya. Pada saat ini untuk semua berkas WinBI, pesan-pesan file PO yang ada berukuran sekitar 770KB setelah digabungkan oleh `ambil-msg`. Format kompresi yang didukung nantinya adalah BZ2, GZip, dan ZIP.
- Menulis ulang program ini menggunakan teknik OOP.

ambil-axe.py

Program `ambil-axe` berfungsi untuk mengumpulkan semua daftar kunci pemercepat (*accelerator key*) ke dalam sebuah file yang mudah diakses. Selanjutnya semua kunci tersebut akan diperiksa, saat ini masih secara manual, apakah konflik dengan kunci yang digunakan di program yang sama. Intinya, satu simbol kunci pemercepat dalam satu program tidak boleh digunakan untuk yang lain. Misalnya kita mendefinisikan tombol `Ctrl-O` sebagai kunci pemercepat untuk masuk ke menu `Buka File`, maka kunci tersebut tidak boleh digunakan lagi untuk tugas apapun dalam program yang sama.

Program ini dikembangkan dengan menggunakan bahasa Python dan memiliki lisensi GPL (**GNU Public License**). Saat ini, program sedang dalam pengembangan yang ekstensif. Hingga versi terakhir (0.0.1), program `ambil-axe.py` sudah memiliki fitur-fitur sebagai berikut :

- Kemampuan menerima input dari suatu file yang diberikan oleh user
- Kemampuan menulis hasil ke dalam suatu file output dengan nama yang diberikan oleh user
- * Berukuran kecil (di bawah 10KB)
- * Dukungan untuk command line option
- * Kemampuan untuk menyimpan panjang frase tiap message yang berisi accelerator key
- * Kemampuan untuk menyimpan tombol accelerator yang digunakan

stats-po.py

Program ini berfungsi untuk menampilkan semua informasi statistik tentang file-file PO. Informasi tersebut adalah :

- Jumlah pesan (message) yang sudah diterjemahkan dalam satu file PO
- Jumlah pesan yang belum diterjemahkan dalam satu file PO
- Jumlah pesan yang masih bersifat fuzzy dalam satu file PO
- Jumlah pesan total

- Jumlah file PO total

Program ini rencananya akan dikembangkan dengan menggunakan bahasa Python dan memiliki lisensi GPL (**GNU Public License**).

taruh-msg.py

Program ini berfungsi untuk menaruh pesan-pesan yang sudah diterjemahkan ke dalam bahasa Indonesia beserta padanan pesan bahasa Inggrisnya ke dalam database. Nantinya pesan-pesan yang sudah ada di dalam database tersebut dapat digunakan untuk melakukan terjemahan secara "kasar" atas file-file PO yang diserahkan dengan menggunakan browser. Program ini rencananya akan dikembangkan dengan menggunakan bahasa Python dan memiliki lisensi GPL (**GNU Public License**). Database yang akan didukung nantinya adalah PostgreSQL dan MySQL.

gsm-libma.py

Program ini akan berfungsi untuk melakukan update terhadap file-file PO yang telah diperbaiki frasenya dalam daftar frase. Dengan kata lain bila `ambil-msg.py` berfungsi untuk mengumpulkan frase, maka `gsm-libma.py` berfungsi untuk menaruh frase-frase yang telah diperbaiki tadi ke file PO semula.

7 Integrasi Smartcard dan PKI

WinBI-NG memanfaatkan model yang mengintrasikan SmarCard dan PKI. Berikut ini adalah paparan singkat mengenai bagaimana hal tersebut dapat dicapai.

7.1 Public Key Infrastructure dan Smartcard

Kunci privat (*private key*) dalam PKI merupakan komponen penting yang harus dijaga oleh pemiliknya agar hanya pemiliknya sajalah yang dapat menggunakannya untuk melakukan autentikasi maupun tanda tangan dijital. Kunci private ini biasanya disimpan dalam sebuah file dalam format p12 (pkcs12) yang terproteksi dengan pass phrase. Dengan memiliki file tersebut dan mengetahui pass phrase yang melindungi kunci privat tersebut, seseorang dapat menggunakannya. Penyimpanan dalam bentuk file terproteksi semacam ini mempunyai banyak kelemahan, karena orang lain dapat dengan mudah mencuri file tersebut (mungkin dengan bantuan virus) dan mencoba mencari pass phrasenya dengan menggunakan brute force attack.

Gambar 27: Smartcard dan smartcard reader/writer

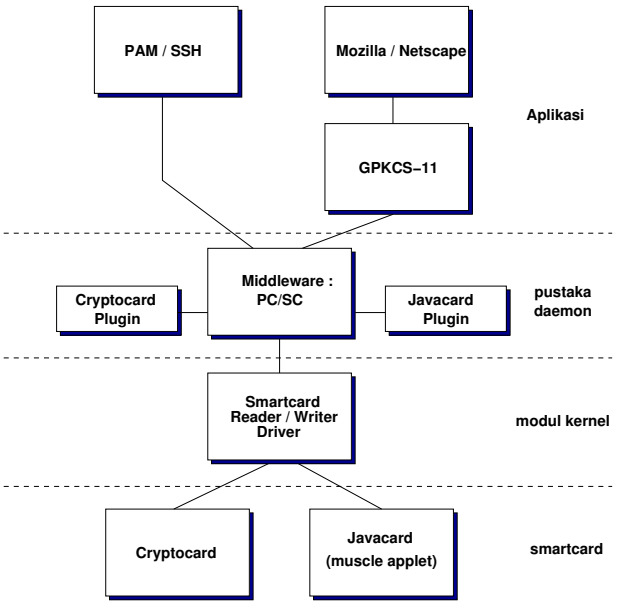
Alternatif yang lebih aman untuk menyimpan kunci privat tersebut adalah dengan menyimpannya di dalam sebuah smartcard. Smartcard merupakan kartu chip komputer mikro yang berisi prosesor dan memori. Sebuah device dapat berkomunikasi dengan smartcard melalui smartcard reader (atau writer). Dengan menggunakan API (Application Programming Interface) yang sesuai dengan smartcard tersebut, sebuah alat (komputer / telefon / alat foto kopi) dapat berkomunikasi

dengannya melalui sebuah alat bernama smartcard reader/writer. Contoh smartcard adalah GSM card yang digunakan sebagai ID di dalam handphone GSM.

Smartcard dapat didisain sebagai sebuah "black box", dengan kata lain isi dari smartcard (program dan memori) tidak dapat diketahui. Sebuah device hanya bisa mengaktifkan smartcard tersebut dengan PIN, dan memanggil fungsi yang tersedia di dalamnya dengan menyertakan data masukan berupa parameter untuk mendapatkan sebuah output. Dengan keistimewaan ini, sebuah smartcard khusus (crypto smartcard) didisain untuk memiliki kemampuan untuk membangkitkan kunci privat dan kunci publik dan kemudian menyimpannya di dalam memori terproteksi. Kunci privat sama sekali tidak dapat dibaca, hanya dapat digunakan melalui pemanggilan fungsi-fungsi enkripsi, dekripsi, sign, dan verifikasi setelah smartcard tersebut diaktifkan dengan PIN. Proteksi PIN memungkinkan smartcard tersebut terkunci apabila terdapat beberapa kali kesalahan memasukkan PIN. Dengan kemampuan ini, resiko pencurian kunci privat dapat dihindari.

7.2 Arsitektur Muscle Project

Integrasi PKI dan smartcard dalam Linux dimotori oleh Muscle Project <`http://www.musclecard.com`> dan Smartsign Project <`http://smartsign.sourceforge.net`>. Muscle Project mengembangkan sebuah arsitektur **Muscle Framework** yang terdiri dari :



Gambar 28: Arsitektur smartcard Muscle Project

- **Muscle Applet**
  Jenis smartcard yang dapat bekerja dalam lingkungan Muscle Framework adalah Schlumberger Cryptoflex dan keluarga Java Card 2.1.1 compliant seperti Schlumberger Cyberflex Access 16k/32k dan Gemplus GemXpresso 211/PK. Cryptoflex dapat langsung digunakan sedangkan keluarga Java Card terlebih dahulu dimuati dengan Muscle Applet yang berisi platform dasar Java Card dan fungsi-fungsi cryptography.

- **Smartcard Reader Driver**
  Smartcard reader/writer dihubungkan ke PC melalui PCMCIA, serial, dan USB. Untuk menjalankan alat ini dibutuhkan driver pada kernel (sebagai module PCMCIA) ataupun sebagai pustaka dinamis (dynamic library) untuk reader serial dan USB.

- **PC/SC Lite (Middleware)**
  PC/SC merupakan spesifikasi standard untuk aplikasi smartcard pada PC yang dibuat oleh PC/SC working group <http://www.pcscworkgroup.com>. PC/SC Lite merupakan hasil implementasi spesifikasi PC/SC oleh Muscle Project yang dikembangkan di atas platform Linux dan Unix. PC/SC menggunakan driver smartcard reader untuk melakukan komunikasi ke smartcard. PC/SC Lite mempunyai fleksibilitas untuk dapat bekerja dengan beberapa jenis smartcard dengan memanfaatkan metode plugin. Lapisan middleware ini menyediakan layanan API yang standard kepada aplikasi-aplikasi yang hendak memanfaatkan smartcard. Selain itu terdapat beberapa administrasi tools yang disertakan untuk mengelola smartcard.

- **GPKCS-11**
  GPKCS-11 merupakan implementasi open source dari PKCS-11 (Cryptographic Token Interface Standard). Standar ini mendefinisikan antar muka komunikasi antara aplikasi dengan suatu sistem kriptografi (token). GPKCS-11 menyediakan pustaka yang dapat dimanfaatkan oleh aplikasi-aplikasi seperti Netscape dan Mozilla. Untuk melakukan akses ke smartcard GPKCS-11 memanfaatkan layanan PC/SC Lite.

- **Aplikasi : autentikasi lokal dan remote**
  Muscle Project mengembangkan sebuah modul PAM agar user dapat melakukan autentikasi lokal dengan menggunakan PKI dan smartcard. Integrasi smartcard dengan OpenSSH dan Netscape/Mozilla (melalui GPKCS-11) dikembangkan agar user dapat melakukan autentikasi secara remote dengan SSH maupun TLS/SSL.

- **Aplikasi : tanda tangan dijital**
  Muscle Framework mendukung fitur tanda tangan digital pada Netscape Email / Mozilla Email (S/MIME) dan secure web form (PKCS7) pada Netscape 4.7. Dengan memanfaatkan kedua fitur tersebut, user dapat mengamankan dokumen dalam bentuk email dan web form untuk melakukan kepentingan bisnis dan pribadi.

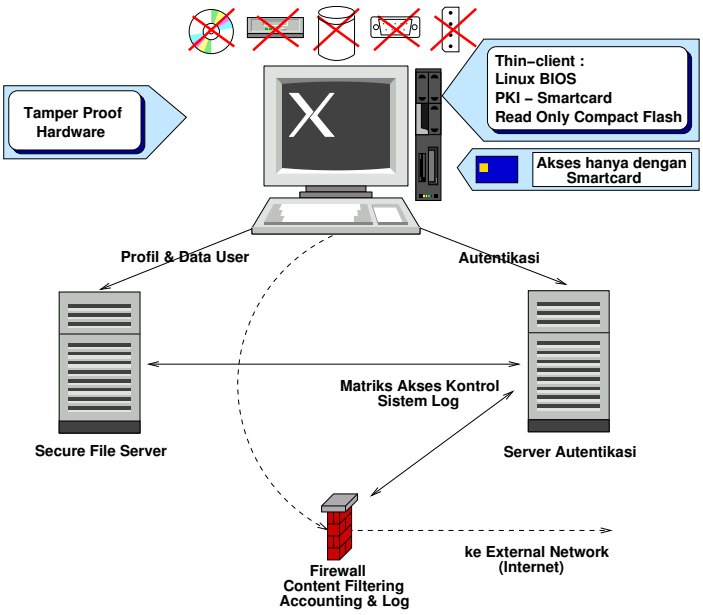
7.3 Pengembangan Selanjutnya : Secure Thin-Client

Banyak insiden bersumber dari kesalahan yang dilakukan oleh user. Problem ini sebenarnya tidak perlu terjadi apabila arsitektur sistem komputer personal didisain dengan konsep-konsep sekuriti. Desain sistem komputer personal yang ada sekarang adalah merupakan pengembangan dari arsitektur terdahulu yang pada waktu pengembangannya tidak memikirkan bahwa komputer personal akan meluas penggunaannya seperti yang terlihat pada saat ini.

Konsep arsitektur sekuriti seperti TCPA (Trusted Computing Platform Alliance) menjanjikan keamanan pada sistem komputer personal. Tetapi banyak pendapat bahwa TCPA merupakan kontrol yang dilakukan oleh vendor terhadap user. Hal ini dipandang sebagai hal yang lebih banyak merugikan user oleh beberapa kalangan terutama komunitas pengembang dan pengguna Open Source.

Sebuah solusi yang disebut dengan secure Thin-Client dengan Linux diharapkan dapat menjamin kebutuhan akan sekuriti serta tidak mengorbankan kepentingan user. Solusi ini berusaha untuk meminimalkan kesalahan yang dibuat oleh user terhadap desktopnya. Setiap desktop didisain agar tidak mempunyai media penyimpanan tetap seperti CDROM, harddisk, floppy dan tidak mempunyai koneksi USB dan serial. Transfer data hanyalah melalui network yang khusus didisain dengan pertimbangan sekuriti. Selain itu perangkat keras dilengkapi dengan tamper-proof-system, seperti menempatkan mekanisme deteksi dengan micro switch, untuk menghindari usaha modifikasi dalam perangkat keras oleh orang yang tidak berhak.

Sistem operasi yang pertama kali dimuat waktu boot adalah Linux BIOS <http://www.acl.lanl.gov/linuxbios> yang tersimpan di dalam BIOS computer tersebut. Tujuan dari menggunakan Linux BIOS ini adalah untuk memuat fungsi sekuriti pada saat mesin pertama kali akan boot, karena banyak pelanggaran sekuriti dilakukan pada saat BIOS akan memuat sistem operasi dan melakukan proses bootstraap.



Gambar 29: Konsep Linux Thin-Client

Kontrol akses dan autentikasi dapat dilakukan oleh Linux BIOS tersebut melalui smartcard dan PIN, atau pada saat sistem operasi telah siap digunakan. Proses autentikasi dilakukan secara offline dengan mekanisme “challenge & sign” atau secara online dengan ticket-based authentication dengan menggunakan Kerberos.

Setelah proses autentikasi dilakukan maka BIOS memuat sistem operasi dari image yang disimpan dalam read only flash card atau langsung dari sebuah file server. Sebelum dimuat, terlebih dahulu BIOS memeriksa hash atau footprint untuk menghindari adanya penyusupan dalam image tersebut. Setelah proses bootstraap, maka didapat sebuah sistem yang bersih dan siap untuk digunakan dengan tingkat sekuriti yang memadai oleh user.

7.4 Layanan keamanan

Di samping memanfaatkan smarcard dan PKI, WinBI-NG memanfaatkan kemungkinan layanan sekurit yang ada pada GNU/linux. Pada dasarnya GNU/Linux merupakan sistem ”multi user”, sehingga tiap user memiliki perizinan yang berbeda (*previledge*). User memiliki ”file atau device” sendiri-sendiri. Sehingga setiap program (atau proses lebih tepatnya) dijalankan pada ruang memori dengan kepemilikan yang tertentu. Berkas (*file*) juga ditentukan kepemilikannya. Dengan cara ini suatu proses yang tak memiliki ”hak” mencukupi maka tak bisa mengubah suatu file, atau mengakses suatu divais.

Dengan model ini, maka misal komputer di rumah digunakan oleh orang tua dan anak, maka orang tua tak perlu khawatir si anak akan menghapus berkas-berkasnya. Atau si anak menghapus berkas sistem yang memaksa harus menginstall ulang sistem. Sehingga masalah virus menjadi sulit timbul di GNU/Linux (walau bukan berarti tak ada), karena suatu program milik pengguna berjalan dengan ”previledge” (perizinan) yang terbatas dan sesuai dengan yang telah didefinisikan. Jadi tidak bisa suatu program yang dijalankan user A, menghapus suatu file, atau menginfeksi file sistem.

Di samping itu keragaman GNU/Linux (karena tiap orang bisa mengkonfigurasi GNU/Linux dan memberikan patch ke bagian manapun dari sistem GNU/Linux maka GNU/Linux menjadi sistem yg tidak monokultur, sehingga bersifat ”multikultur”. Seperti halnya pertanian monokultur yg lebih tahan terhadap virus (hama) maka hal ini terjadi juga di Linux, karena WORM dan

VIRUS menjadi sulit tersebar secara massal di lingkungan Linux.

Pada "kernel Linux" telah diberikan fitur-fitur sekuriti dasar misal :

- multi user dan cabability (untuk membedakan antar user), compartment yang "memenjarakan" suatu aplikasi sehingga berjalan di lingkungan yang aman.
- tcpflooding protection, dan proteksi level kernel untuk TCP/IP
- file system cryptography
- divais `/dev/random` dan `/dev/urandom`. Divais ini ditangani kernel Linux dan menyediakan data random yang sangat dibutuhkan untuk aplikasi seperti kriptografi, "bibit" untuk TCP Sequence number (bila TCP sequence number tidak benar-benar random maka koneksi TCP/IP mudah dihijack (dibajak, sehingga kita dibelokkan ke server palsu).
- infrastruktur "netfilter" yang merupakan sistem pemfilteran paket yang mengimplementasikan ipchain dan ipwfadm, sehingga infrastruktur ini menyediakan kemampuan mengubah paket ketika melewati bagian-bagian kernel. Dengan infrastruktur ini maka dapat diberikan fungsi "masquerading" (ip lokal tidak tampak, dan yang berhubungan dengan Internet hanya 1 IP), juga kemampuan "*statefull inspection*" dapat ditambahkan

Pada Linux pengguna/administrator dapat memilih pada tingkat manakah fitur sekuriti ditambahkan misal :

- Pada tingkat kernel
- Pada tingkat antara kernel - aplikasi
- Pada tingkat aplikasi

Misal pada tingkat kernel dapat diterapkan beberapa patch yang memberikan penambahan fungsi security :

- **Openwall** patch <`http://www.openwall.com`> yang melindungi dari serangan stack, buffer overflow, /tmp dan beberapa serangan.
- **LIDS** (Linux Intrusion Detection) <`http://www.lids.org`> dengan cara ini pemanfaatan capability bit lebih tinggi lagi, sehingga bisa diatur model kepemilikan file, lebih ditail (misal walau root ketika sistem beroperasi tetap tak bisa menghapus log file, atau meload module, dlsb)
- **ACL** patch <`http://acl.bestbits.at/`> digunakan untuk memenuhi prasyarat C2 yg menggunakan MAC. Dengan patch ini maka Linux dapat memberikan bit tambahan untuk kontrol akses. Dikenal dengan penambahan fungsi MAC (Mandatory Access Control)
- **RSBAC** (Rule Set Base Access Control) <`http://www.rsbac.org/`> untuk menambah kontrol akses pada sistem linux. Dengan cara ini seorang pengguna dapat diatur akses-nya sesuai dengan *"role"* pada organisasi (pada model biasa hanya group saja)
- **Medusa** <`http://medusa.fornax.sk/`> yang menerapkan model virtual space untuk mengakses object dengan menggunakan matrix access. Medusa ini memungkinkan siatem memiliki kebijakan akses yang lebih luwes (bisa menjadi RSBAC, MAC, atau lainnya)
- **SELINUX** (Flask model) <`http://www.nsa.gov/selinux/`> merupakan suatu sistem Linux yg dikembangkan oleh NSA sehingga model akses, model akses sistem call menjadi lebih aman dan tercatat.

- **Grsecurity** <http://www.grsecurity.net/>, merupakan sistem ACL yang dapat membatasi akses ke berkas, kapabilitas, sumber daya komputasi dan atau socket ke semua pengguna termasuk **root.** Fitur lainnya adlah melindungi serangan yang mencoba mendapatkan root, sehingga mendapatkan root bukan berarti mendapatkan akses penuh ke sistem. Akses dapat diberikan pada suatu proses yang membutuhkan. Hal ini mempersulit diserangnya suatu sistem

Di level atas Kernel, dapat juga dilakukan security tambahan (misal di level library, atau sebelum aplikasi).

- **PAM** (Pluggable Authentication Module) sebetulnya tidak tepat disebut level library, tetapi bisa dikatakan levelnya antar kernel dan program aplikasi. Beberapa library juga memanfaatkan PAM untuk melakukan proses otentifikasi.
- **compartm** (chroot dan lain sebagainya), ini levelnya antara aplikasi yangg dijalankan dan kernel. Sehingga suatu proses dijalankan dalam *"penjara"* nya.

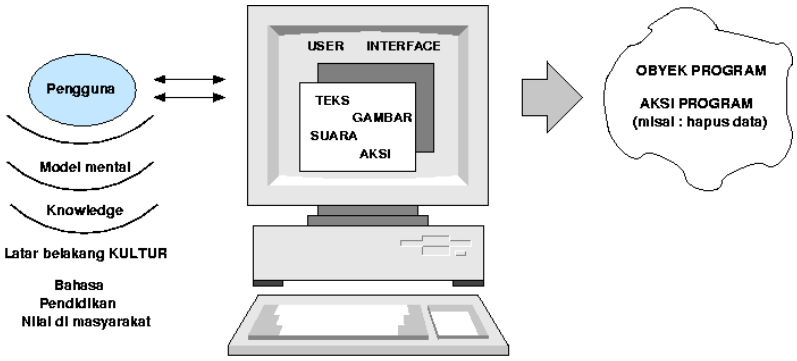
Sedangkan di level aplikasi (distro Linux) biasanya suatu distro telah menyertakan beberapa aplikasi :

- **iptables** (untuk memfungsikan sebagai firewall, masquerading)
- **scanlog** (untuk mencatat ketika ada yang melakukan scan ke port)
- **tcpwrapper** (service TCP/IP tidak langsung berbicara langsung dg client
- **snort** (intrusion detection)
- **aide** (advanced intruder detection)
- **tripwire** (untuk mengecheck apakah binary ada yg diganti, integrity testing)
- **argus** (perangkat monitor jaringan)
- **GPG** (Gnu Privacy Guard)
- **ippl** (IP protocol logger) untuk melakukan log tiap protocol
- **nessus** (penetration test)
- **nmap** (untuk menguji sistem kita, apakah ada port terbuka)
- **amavis**, anti virus
- **honeyd**, yang dapat memberikan ilusi bahwa suatu server tampak sebagai sistem operasi lain.

Dan masih banyak lagi aplikasi security yang dapat diinstal (misal **chkrootkit**, untuk menguji apakah sistem kita dimasuki oleh intruder, yangg telah menginstal rootkit).

8 Gotong royang mengembangkan WinBI-NG

Dalam pelaksanaan pengembangan WINBI ini, disadari akan terbatasnya perangkat bantu dan pengetahuan yang ada di Indonesia mengenai proses penerjemahan. Baik dari sisi teknologi maupun dari sisi studi bahasa Indonesia. Pata tulisan ini juga dijabarkan beberapa perangkat bantu yang digunakan, serta beberapa perangkat lunak bantu yang dikembangkan untuk melakukan proses penerjemahan. Pada penerjemahan program komputer beberapa pertimbangan-pertimbangan baru perlu dilakukan. Sayangnya hingga saat ini studi yang berkaitan dengan penerjemahan menu dan manual masih langka di lakukan, baik di bidang komputer ataupun bidang bahasa Indonesia. Sehingga bisa dikatakan bidang ini seperti bidang yang terlupakan di Indonesia.



Gambar 30: Mapping GUI

Studi berkaitan soal Interaksi Komputer Manusia (*Human Computer Interaction - HCI*) yang berkaitan dengan bahasa masih minim dilakukan di Indonesia. Beberapa studi komputer linguistik terutama kaitannya dengan analisis bahasa Indonesia oleh mesin atau dengan kata lain mencoba menguak permasalahan *bagaimana mesin memahami bahasa Indonesia* . Misal yang dilakukan oleh BPPT (Kamus Elektronik Bahasa Indonesia), atau oleh UI (*information retrieval*). Sedangkan studi bahasa Indonesia mengenai bagaimana manusia memahami elemen berbahasa Indonesia pada komputer (manual, halaman web, tombol navigasi dan sebagainya), masih tergolong minim.

Tentu saja masih merupakan pertanyaan besar di Indonesia, di jurusan atau fakultas manakan studi tersebut diletakkan ? Fakultas Psikologi, Bahasa, Komputer, atau lainnya ? Struktur universitas, dan fakultas di Indonesia yang masih sangatlah rigid sepertinya masih sulit untuk mengakomodasi kebutuhan-kebutuhan studi seperti ini. Mungkin perlu reformasi Universitas ? Atau sekedar kerjasama antara disiplin ilmu ? Dua-duanya adalah hal yang masih langka di Indonesia.

WinBI-NG barulah pada taraf awal, sehingga masih banyak yang perlu dilakukan. Untuk melaksanakan tugas seperti penerjemahan GUI ataupun penyusunan sistem WinBI-NG, tidaklah mungkin hanya dilakukan oleh segelintir orang saja, betapapun hebatnya mereka. Oleh karena itu, segala partisipasi Anda sangat diharapkan. Ada banyak bidang yang membutuhkan bantuan Anda, di antaranya adalah :

- Pengembangan. Jika anda tertarik untuk melakukan pemrograman di sistem operasi GNU/Linux, tidak ada salahnya ada ikut serta terlibat dalam pengembangan WINBI dari sisi pemrograman. Beberapa bahasa pemrograman yang mungkin harus anda pelajari dan kuasai adalah C, Python, Perl, PHP, Java, BASH, Ruby.

- Dokumentasi. Seringkali bidang dokumentasi ini tidak kita anggap sebagai bagian penting dalam mengembangkan suatu sistem. Padahal dalam kenyataannya, bidang ini mungkin merupakan bidang yang paling sukar, karena ada kalanya betapapun hebat sistem kita, namun bila tidak dapat digunakan oleh pengguna karena ketidakpahamannya, maka sistem kita tersebut menjadi sia-sia belaka. Beberapa software bantu yang perlu anda pelajari bila ingin memberikan sumbangsih dalam bidang ini adalah LyX/KLyX, LaTeX, program-program gambar seperti GIMP, XFig, serta kemampuan membuat program skrip terkadang dapat membantu.

- Menggunakan WINBI. Tentu saja bila anda tidak ingin berpartisipasi di kedua bidang di atas, anda masih dapat memberikan sumbangsih kepada proyek WINBI dengan menggunakannya dalam kegiatan sehari-hari anda dan melaporkan kesalahan-kesalahan yang anda temukan selama menggunakannya.

Mungkin Anda bertanya-tanya, “bila saya ikut membantu proyek pengembangan WINBI, lalu apa manfaatnya bagi saya ?”. Kami tentu saja tidak dapat menjanjikan imbalan apapun kepada

Anda. Namun ada beberapa hal yang mungkin berguna bagi Anda dengan ikut serta terlibat dalam proyek WINBI ini yaitu :

- Jika anda saat ini masih menjadi mahasiswa tingkat akhir, baik itu Sarjana Strata-1, Strata-2 atau bahkan Strata-3, dan mengalami kesulitan dalam mencari topik penelitian skripsi, ada beberapa bidang penelitian dalam proyek pengembangan WINBI yang menarik untuk dijadikan topik penelitian baik di bidang sosial (ekonomi, bisnis, sosiologi, psikologi, sastra) maupun di bidang teknik (informatika, MIPA, elektro).

- Pengalaman bekerja dalam tim. Dalam sekolah kita telah terbiasa untuk bekerja sendiri-sendiri (individu) padahal dalam kenyataan hidup, kita tidak dapat bekerja secara individual. Jadi dengan ikut serta dalam proyek pengembangan WINBI Anda akan merasakan suka dukanya bekerja dalam tim, lebih tepatnya tim virtual, karena lokasi masing-masing anggota tidak lagi terikat oleh lokasi fisik. Bahkan mungkin anda akan mendapatkan pengalaman untuk menjadi pemimpin tim virtual tersebut. Pengalaman-pengalaman ini tentu saja akan secara langsung maupun tidak langsung akan menjadi bekal anda dalam mengarungi kehidupan ini.

Dari contoh di atas, tampak bahwa WINBI sangat mungkin untuk dikembangkan lebih lanjut dan batasannya hanyalah imajinasi dan kemampuan anda. Tidak ada batasan legalitas seperti lisensi yang ada pada produk lainnya. Faktor lain yang perlu dipertimbangakn dalam memanfaatkan WinBI untuk memberikan layanan ataupun memberikan nilai tambahan, adalah perlunya pertimbangan dari sisi pengguna (*user centered*). Dengan kata lain fitur apa yang ditambahkan, konfigurasi yang dilakukan haruslah bergantung pada end user yang dituju oleh layanan atau produk yang diturunkan dari WINBI tersebut.

9 Penutup

Semakin kompleks dan saling terhubungnya antar bagian dalam sistem, menjadikan sistem makin sulit untuk dijamin keamanannya. Sekuriti adalah suatu proses, bukan produk. Sebagai proses maka sekuriti itu memiliki banyak komponen. Sekuriti juga seperti rantai yang terdiri dari banyak mata rantai. Seperti halnya rantai, maka kekuatan sistem setara dengan kekuatan dari mata rantai yang terlemah.

Pengguna adalah mata rantai terlemah pada sekuriti suatu sistem. Untuk sistem yang mempersyaratkan keamanan yang baik, maka perilaku pengguna, termasuk bahasa dan simbol yang digunakan pada GUI haruslah dipertimbangkan. Begitu juga dengan perangkat desktop yang sering dianggap bukan bagian yang menentukan pada sekuriti suatu sistem, haruslah menjadi pusat perhatian, karena banyak kasus sekuriti timbul akibat kelemahan sistem operasi ataupun aplikasi di sisi desktop.

WinBI-NG memanfaatkan GUI berbahasa Indonesia serta SmartCard dan PKI untuk menyediakan solusi desktop yang aman (*secure desktop*). Karena WinBI-NG ini bersifat Open Source yang dikembangkan di Indonesia, maka faktor ketergantungan terhadap negara lain dapat dikurangi. Partisipasi rekan-rekan pengembang Indonesia sangat diharapkan, baik sebagai penerjemah, ataupun sebagai pengembang.

Pustaka

[1] Abadi, Martin (1997). Explicit communication revisited: two new attacks on authentication protocols. IEEE Transactions on Software Engineering, vol 23 (3), Maret 1997, 185 - 186.

[2] Adams, Anne dan Martina Angela Sasse (1999). Users are not the enemy. *Communication of the ACM* . Desember 1999, vol 42 (12), 41-45.

[3] Butler, Randy, Von Welch, Douglas Engert, Ian Foster, Steven Tuecke, John Volmer, Carl Kesselman (2000). A national scale authentication infrastructure. *IEEE Computer* , Desember 2000, 60-64.

[4] Cybenko, George, Guofei Jiang (2000). Developing a distributed system for infrastructure protection. *IT Pro* , July/Agustus 299 hlm. 17 - 22.

[5] *Computerzeitung*, Interview Dirk Henz-BSI : Opensource ist positiv. Nr. 22/31 Mei 2001.

[6] Gollmann, Dieter (1999). *Computer Security*. England : John Willey & Sons Inc.

[7] Gutzmann, Kurt (2001). Access Control and Session Management in the HTTP Environment. *IEEE Internet Computing*, January-February 2001, hlm 26-35.

[8] Heintze, Nevin, J. D. Tyger (1996). A model for secure protocols and their compositions. *IEEE Transactions on Software Engineering*, vol 22 (1), Januari 1996. hlm. 16 - 30.

[9] **Information Security Solutions Europe** (ISSE 99), Berlin 14 October 1999 dapat dibaca di `http://europa.eu.int/comm/commissioners/liikanen/speeches/051099_en.htm`

[10] James B.D. Joshi, Walid G. Aref, Arif Ghafoor, Eugene H. Spafford (2001). Security Models for Web-Based Applications. *Communications of the ACM*, February 2001/Vol. 44. No 2, page 38-44.

[11] James B.D. Joshi, Walid G. Aref, Arif Ghafoor, Eugene H. Spafford (2001). Digital Government Security Infrastructure Design Challenges. I*EEE Computer*, February 2001, hlm 66-72.

[12] Joshi, Ghafoor, Aref, Spafford, "Digital Government Security Infrastructure Design Challenges"

[13] Ladkin, Peter B (1999). *Comment on security*. Lecture material.

[14] Madsen, Mark, Andrew Herbert (1997). A guide to secure electronic bussiness using the E2S architecture. *Web Security : A matter of trust* . USA : O Reilly.

[15] Michener, John (1999). System Insecurity in the Internet Age. *IEEE Software* , July/August 1999, 62-68.

[16] Ronald, Edmund M.A, Moshe Sipper (2000). The challenge of tamperproof Internet Computing. *IEEE Computer*. Oktober 2000, hlm 98-99.

[17] Schneier, Bruce (1996). *Applied Cryptography*. Canada : John Willey & Sons Inc.

[18] Schneier, Bruce (2000). Semantic Network Attacks. *Communications of the ACM* vol 43(12), Desember 200.

[19] Schneier, Bruce (2000). *Secrets & Lies*. USA : John Willey and Sons.

[20] Simon Liu, John Sullivan, Jerry Ormaner. A Practical Approach to Enterprise IT Security. *IT Pro*, September-Oktobe 2001, 35-42.

[21] White House (2000). *National Plan for Information System Protection ver 1.0*

[22] Wiryana, I Made, Avinanta Tarigan (2000). Public Key Infrastructure dan Open Source. *Seminar : Secure your Future*. Tersedia di `http://www.pandu.org/Security/artikel-01`

[23] Wiryana, I Made, Tedi Heriyanto (2001). *Resiko Internet Banking telah tampak*. Diterbitkan di DETIK.COM. Tersedia di `http://www.pandu.org/Security/artikel-03`

[24] Wiryana, I Made (2001b). *Jangan angap enteng virus SMS*. Tersedia di `http://www.pandu.org/Security/artikel-02`

[25] Wiryana, I Made (2001c). *Berbahayanya modem booster*. Tersedia di `http://www.pandu.org/Security/artikel-04`

[26] Womack, Helen (1998). *Under Cover lives : Soviet spies in the cities of the world* . London : Weidenfeld Nicholson : London, 1998.

[27] Zwicky, Elizabeth D, Simon Cooper, D. Brent Chapman (2000). *Building Internet Firewall*. OReilly and Associates