BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Untuk membuat tampilan yang menarik memang tidak mudah dilakukan. Seorang perancang tampilan selain harus mempunyai jiwa seni yang memadai, ia juga harus mengerti selera pengguna secara umum. Hal lain yang pelu disadari oleh seorang perancang tampilan adalah bahwa ia harus bisa meyakinkan pemrogramnya bahwa apa yang ia bayangkan dapat diwujudkan dengan peranti bantu yang tersedia (Insap Santoso ; 2009 : 185).

Perancangan merupakan proses pengolahan hasil analisis perangkat lunak menjadi rencana pengembangan perangkat lunak dan batasan-batasan perangkat lunak atau masalah yang mungkin dihadapi dalam pengembangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan arsitektur, perancangan modul, dan perancangan antarmuka. (Y. Yohakim Marwanta ; 2010 : 5).

Bagi perancang antarmuka, hal yang sangat penting untuk ia perhatikan adalah mendokumentasikan semua pekerjaan yang dilakukan. Dokumentasi rancangan dapat dikerjakan atau dilakukan dengan beberapa cara :

1. Membuat sketsa pada kertas
2. Menggunakan peranti purwarupa GUI
3. Menuliskan keterangan yang menjelaskan tentang kaitan antara jendela.
4. Menggunakan peranti bantu CASE (*Computer Aided Software Engineering*).

Cara kedua dan keempat tidak selalu dapat diterapkan, karena peranti tersebut biasanya harus dibeli dan seringkali cukup mahal. Cara ini kebanyakan

diterapkan pada pembuatan antarmuka grafis untuk suatu jenis pekerjaan berskala besar.

II.1.1. Cara Pendekatan

Sebuah program aplikasi pastilah ditujukan kepada pengguna, yang utama, bukan perancangan program aplikasi tersebut. Program aplikasi pada dasarnya dapat dikelompokkan dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula dan program aplikasi yang akan digunakan oleh pengguna umum, yang juga sering dikenal dengan sebutan public software. Karena perbedaan pada calon pengguna, maka perancang program antarmuka perlu memperhatikan hal ini (Insap Santoso ; 2009 : 186).

Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi yang serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna maupun ragam antarmuka yang akan digunakan. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan, yakni pendekatan yang disebut dengan pendekatan perancangan berpusat ke pengguna (*user centered design approach*). Cara pendekatan ini berbeda pendekatan perancangan oleh pengguna (*user design approach*).

Pendekatan perancangan berpusat ke pengguna adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna di sini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancangan antarmuka sedang

menggambar wajah antarmuka yang mereka inginkan. Dengan kata lain, perancangan dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya. Sementara perancangan menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna, seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan (Insap Santoso ; 2009 : 187).

Pada perancangan oleh pengguna, pengguna sendirilah yang merancang wajah antarmuka yang diinginkan. Di satu sisi, cara ini akan mempercepat proses pengimplementasian modul antarmuka. Tetapi di sisi yang lain, hal ini justru sangat memberatkan pemrogram karena apa yang diinginkan pengguna belum tentu dapa diimplementasikan dengan mudah, atau bahkan tidak dapat dikerjakan dengan menggunakan peranti bantu yang ada.

Perancang program aplikasi yang dimasukkan dalam kelompok kedua, aau public *software*, perlua menganggap bahwa program aplikasi tersebut akan digunakan oleh pengguna dengan berbagai tingkat kepandaian dan karakteristik yang sangat beragam. Di satu sisi keadaan ini dapat ia gunakan untuk memaksa pengguna menggunakan antarmuka yang ia buat, tetapi pada sisi lain pemaksaan itu akan berakibat bahwa program aplikasinya menjadi tidak banyak penggunanya. Satu kunci penting dalam pembuatan modul antarmuka untuk program-program aplikasi pada kelompok ini adalah dengan melakukan customization. Dengan customization pengguna dapat menggunakan program aplikasi dengan wajah antarmuka yang sesuai dengan selera masing-masing pengguna.

Salah satu contoh dari adanya kemampuan yang dimiliki oleh sebuah program aplikasi atau sistem operasi yang dapat disesuaikan dengan karakteristik pengguna adalah pengaturan desktop pada OS X versi 10.5 favoritnya, sehingga pengguna dapat mengubahnya sesuai keinginan justru akan membuat mata pengguna itu sakit, dikarenakan mata harus melakukan akomomasi maksimum terus menerus untuk menyesuaikan dengan warna tampilan yang ada.

Selain cara pendekatan yang dijelaskan di atas, Anda yang terbiasa menulis program-program aplikasi mungkin mempunyai cara khusus untuk berhadapan dengan pengguna. Tetapi perlu Anda ingat bahwa apapun cara yang Anda gunakan, Anda tetap harus mempunyai pedoman bahwa pada akhirnya program itu bukan untuk Anda sendiri, tetapi akan digunakan oleh orang lain. Dengan kata lain, jangan pernah mengabaikan pendapat (calon) pengguna program aplikasi Anda. (Insap Santoso ; 2009 : 188).

II.1.2. Prinsip Dan Petunjuk Perancangan

Antarmuka pengguna secara alamiah terbagi menjadi empat komponen model pengguna, bahasa perintah, umpan balik, dan penampilan informasi. Model pengguna merupakan dasar dari tiga komponen yang lain (Insap Santoso ; 2009 : 188).

Model mental pengguna merupakan model konseptual yang dimiliki oleh pengguna ketika ia menggunakan sebuah sistem atau program aplikasi. Model ini memungkinkan seorang pengguna untuk mengembangkan pemahaman mendasar tentang bagian yang yang dikerjakan oleh program, bahkan oleh pengguna yang sama sekali tidak mengetahui teknologi komputer. Dengan pertolongan model itu pengguna dapat mengantisipasi pengaruh suatu tindakan yang dilakukan dan dapat

memilih strategi yang cocok untuk mengoperasikan program tersebut. Model pengguna dapat berupa suatu simulasi tentang keadaan yang sebenarnya dalam dunia nyata, sehingga ia tidak perlu mengembangkannya sendiri dari awal.

Setelah pengguna mengetahui dan memahami model yang di inginkan, dia memerlukan peranti untuk memanipulasi model itu. Peranti pemanipulasian model ini sering disebut dengan bahasa perintah (command language), yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasionalkan. (Insap Santoso ; 2009 : 189).

Komponen ketiga adalah umpan balik. Umpan balik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa bentuk umpan balik terutama ditujukan kepada pengguna pengguna yang belum berpengalaman dalam menjalankan program sebuah aplikasi. Umpan balik dapat digunakan untuk member keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut.

Komponen keempa adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini perancang harus menampilkan pesan-pesan tersebut seefektif mungkin sehingga mudah dipahami oleh pengguna. Setelah memahami beberapa prinsip dalam perancangan antarmuka pengguna. Pada

bagian berikut ini akan diberikan petunjuk singkat tentang perancangan anarmuka yang akan Anda lakukan sebagai seorang perancang tampilan.

II.1.3. Urutan Perancangan

Perancangan dialog, seperti halnya perancangan sistem yang lain, harus dikerjakan secara aas ke bawah. Proses perancangannya dapat dikerjakan secara bertahap sampai rancangan yang diinginkan terbentuk, yaitu sebagai berikut (Insap Santoso ; 2009 : 190).

1. Pemilihan ragam dialog

Untuk suatu tugas tertentu, pilihlah ragam dialog yang menurut perkiraan cocok untuk tugas tersebut. Ragam dialog dapat dipilih dari sejumlah ragam dialog yang telah dijelaskan pada bab-bab sebelumnya. Pemilihan ragam dialog dipengaruhi oleh karakterisiktik populasi pengguna, tipa dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut. Ragam dialog yang terpilih dapat berupa sebuah ragam tunggal, atau sekumpulan ragam dialog yang satu sama lain saling mendukung.

2. Perancangan Struktur Dialog

Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna sebaiknya banyak dilibatkan, sehingga pengguna langsung mendapatkan umpan balik dari diskusi yang terjadi. Pada tahap ini suatu purwarupa dialog seringkali dibuat untuk memberik gambarang yang lebih jelas kepada calon pengguna.

3. Perancangan format pesan

Pada tahap ini tata letak tampilan dan keterangan tekstual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengefektifkan pengguna tombol.

II.2. Algoritma Kriptografi

Ditinjau dari asal usulnya, kata algoritma mempunyai sejarah yang menarik. Kata ini muncul di dalam kamus Webster sampai akhir tahun 1957. Kata algorisma mempunyai arti proses perhitungan dalam bahasa Arab. Algoritma berasal dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi (al-Khuwarizmi dibaca oleh orang barat sebagai algorism). Kata algorism lambat laun berubah menjadi algorithm.

Definisi terminologi algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu :

1. Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiannya. Pesan asli disebut plaintext, yang diubah menjadi kode-kode. Sama halnya dengan tidak mengerti akan sebuah kata maka akan melihatnya di dalam kamus atau daftar

istilah. Bedan halnya dengan enkripsi, untuk mengubah teks asli ke bentuk teks kode menggunakan algoritma yang dapat mengkodekan data yang diinginkan.

2. Deskripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya teks asli, disebut dengan enkripsi pesan. Algoritma yang digunakan untuk deskripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi.
3. Kunci yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan deskripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).
4. Keamanan dari algoritma kriptografi tergantung pada bagaimana algoritma itu bekerja. Oleh sebab itu algoritma semacam ini disebut dengan algoritma terbatas. Algoritma terbatas merupakan algoritma yang dipakai sekelompok orang untuk merahasiakan pesan yang mereka kirim. Jika salah satu dari anggota kelompok itu keluar dari kelompoknya amak algoritma yang dipakai diganti dengan yang baru. Jika tidak maka hal itu bisa menjadi masalah di kemudian hari.

II.3. Macam-macam algoritma Kriptografi

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya :

1. Algoritma Simetri.

   Pada sistem kriptografi kunci simetris, kunci untuk enkripsi sama dengan kunci untuk dekripsi, oleh karena itulah dinamakan kriptografi simetris.

Keamanan sistem kriptografi simetris terletak pada kerahasiaan kuncinya. Kriptografi simetris merupakan satu-satuanya jenis kriptografi yang dikenal dalam catatan sejarah hingga tahun 1976.

2. Algoritma Asimetri.

   Algoritma asimetris menggunakan dua jenis kunci, yaitu kunci publik dan kunci rahasia. Nama lainnya adalah kriptografi kunci publik, sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan.

3. *Hash Function*.

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Algoritma ini sudah ada sejak lebih dari 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsi pesan yang dikirim.

II.4. RC5 (*Rivest Code 5*)

Algoritma enkripsi RC5 didesain oleh Profesor Ronald Rivest dan pertama kali dipublikasikan pada Desember 1994. Sejak publikasinya RC5 telah menarik perhatian banyak peneliti dalam bidang kriptografi dalam rangka menguji tingkat keamanan yang ditawarkan oleh algoritma RC5 (RSA Laboratory Technical Report TR-602). (Sayekti Harits Suryawan, dkk ; 2011 : 15).

Parameter-parameter yang digunakan dalam RC-5 adalah sebagai berikut :

1. Jumlah putaran ini disimbolkan dengan *r* yang merupakan parameter untuk rotasi dengan nilai 0, 1, 2, …… 255.

2. Jumlah *word* dalam bit disimbolkan dengan *w*. Nilai bit yang di *support* adalah 16 bit, 32 bit, dan 64 bit.
3. Kata kunci (*key word*) Variable ini disimbolkan dengan *b* dengan range 0, 1, 2, …. 255. *Key word* ini dikembangkan menjadi *array S* yang digunakan sebagai *key* pada proses untuk enkripsi dan dekripsi.

Untuk memahami cara kerja RC-5, dapat dimulai dengan melihat konsep dasar bagaimana RC-5 ini bekerja. RC-5 Menggunakan operasi dasar untuk proses enkripsi sebagai berikut :

1. Data yang akan dienkripsi dikembangkan menjadi 2 bagian bagian kiri dan bagian kanan dan dilakukan penjumlahan dengan *key word* yang yang telah diekspansi sebelumnya. Penjumlahan ditunjukan dengan tanda ``+``, dan disimpan di dua register A dan register B.
2. Kemudian dilakukan operasi EX-OR, yang ditandai dengan tanda ``⊕``.
3. Melakukan rotasi kekiri (*shift left*) sepanjang *y* terhadap *x word* yang ditandai dengan *x*<<< *y*. *y* merupakan interpretasi *modulo w* atau jumlah kata *w* dibagi 2. Dengan *lg[w]* ditentukan jumlah putaran yang dilakukan.
4. Tahap akhir dilakukan penggabungan untuk mendapatkan data yang telah dienkripsi.

Proses dekripsi dilakukan dengan konsep dasar sebagai berikut :

1. Data yang telah dienkripsi dikembangkan kembali menjadi 2 bagian dan disimpan di dua register A dan register B.
2. Kemudian dilakukan rotasi ke kanan sejumlah *r*.
3. Selanjutnya dilakukan operasi EX-OR yang ditandai dengan ``⊕``.

4. Tahap akhir dilakukan pengurangan terhadap masing-masing register dengan key *word* yang ditunjukan dengan tanda ``-``, untuk mendapatkan *plaintext.* (Putranto P Bambang, 2010 ; 19).

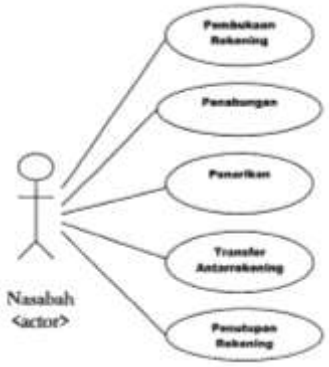
II.5. Pemodelan

Pemodelan UML perangkat lunak bekerja dengan cara yang cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan / gedung berskala besar. Saat seorang arsitek atau insinyur teknik sipil akan membuat sebuah bangunan / gedung berskala besar, ia biasanya membuat denah-denah atau maket-maket yang menggambarkan bentuk jadi dari bangunan / gedung. Sebagai seorang perancang sistem perangkat lunak juga bertindak dengan cara yang serupa, hanya saja yang kita rancang bukan bangunan, melainkan sistem perangkat lunak. Menggambarkan komponen-komponen sistem perangakt lunak dalam bentuk-bentuk geometri tertentu misalmya untuk menggambarkan suatu kelas (class) dalam aplikasi, menggunakan antarkelas (asosiasi), menggunakan garis lurus (Adi Nugroho ; 2009 : 6).

II.5.1. Diagram *Use Case*

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan use case diagram yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, use case diagram tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan (design), untuk mencari (mencoba menemukan) kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (testing) (Adi Nugroho; 2009:7).

Dengan menggunakan use case diagram, kita akan mendapatkan banyak informasi yang sangat penting yang berkaitan dengan aturan-aturan bisnis yang coba kita tangkap. Dalam hal ini, setiap objek yang berinteraksi dengan sistem perangkat lunak misalnya, orang, suatu perangkat keras, sistem lain, dan sebagainya merupakan actor untuk sistem perangkat lunak, sementara use case merupakan deskripsi lengkap tentang bagaimana sistem perangkat lunak berperilaku untuk para actornya. Dengan demikian, use cara diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para actor dengan sistem perangkat lunak yang sedang kita kembangkan. Untuk lebih jelas dapat dilihat pada Gambar II.1.



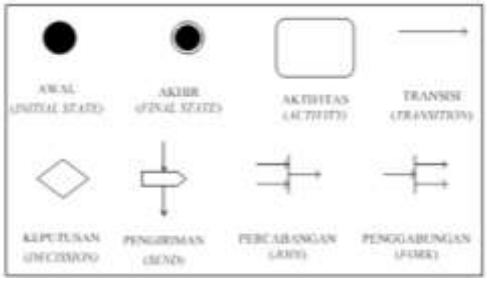
Gambar II.1. Diagram Use Case
(Sumber : Adi Nugroho; 2009 : 8)

Actor pada dasarnya ditentukan berdasarkan perannya (role) pada program /aplikasi yang sedang kita kembangkan, bukan sebagai objek-objek secara mandiri. Sebagai contoh, jika mengambil kasus ATM (Anjungan Tunai Mandiri), seseorang (objek tunggal) mungkin bisa dikelompokkan sebagai actor Karyawan Bank serta Nasabah jika oarng tersebut merupakan karyawan bank yang bersangkutan sekaligus sebagai nasabah karena memiliki tabungan di bank tersebut. Sementara itu, Adi, Ana Geuis, dan beberapa orang lainnya dapat

dikelompokkan menjadi actor nasabah jika mereka semua masing-masing memiliki tabungan di bank tersebut.

III.5.2. Diagram *Activity*

Activity diagram menggambarkan urutan aktifitas yang digunakan untuk menjelaskan aktifitas dari sebuah operasi. Pada activity diagram terdapat keadaan aksi yang berisi spesifikasi dari aktifitas tertentu. Diagram ini berisi, pilihan keputusan dan kondisi serta spesifikasi message yang dikirim atau diterima sebagai gambaran dari aksi.



Gambar II.2 : Diagram Activity
(Sumber : Adi Nugroho ; 2009)

III.5.3. *Sequence*

Diagram interaksi yang menekankan pada waktu pengiriman *message*. *Sequence* diagram menunjukkan sekumpulan objek dan pengiriman serta penerimaan *message* antar objek. Objek yang umumnya memiliki nama atau instansiasi dari *class*, tapi dapat pula merupakan turunan dari *things* lain, seperti *collaboration*, *component* dan *node*. Diagram ini digunakan untuk mengilustrasikan *dynamic view* dari sistem.

III.5.4. *State*

Sebuah *state* diagram menggambarkan keadaan mesin, transisi, *event* dan *activity*. Diagram ini adalah pelengkap khusus untuk mendeskripsikan sebuah *class* yang menggambarkan *state* dari objek dari *class* dan *event* yang menyebabkan *state* berubah. *Event* tersebt dapat berasal dari objek yang mengirimkan suatu *message* atau dari kondisi yang terpenuhi.

II.6. Bahasa Pemrograman Visual Basic 2010

Visual Basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi Windows. Visual Basic 2010 atau Visual Basic 9 adalah versi terbaru yang telah diluncurkan oleh Microsoft bersama C#, visual C++, dan Visual Web Developer dalam satu paket Visual Studio 2010 (Wahana Komputer; 2010: 2).

Visual Basic 2010 merupakan aplikasi pemrograman yang menggunakan teknologi *.NET Framework.* Teknologi *.NET Framework* merupakan komponen Windows yang terintegrasi serta mendukung pembuata, penggunaan aplikasi, dan halaman web. Teknologi *.NET Framework* mempunyai 2 komponen utama, yaitu CLR (*Common Language Runtime*) dan *Class Library*, CLR digunakan untuk menjalan aplikasi yang berbasis .NET, sedangkan *Library* adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi.

Sebelum menginstall komputer harus memenuhi beberapa persyaratan agar Visual Basic 2010 dapat dijalankan dengan baik. Adapun, persyaratan *(System Requirements*) yang harus dipenuhi dapat Anda lihat pada Tabel II.1.

Tabel II.1. Sistem Requirements Visual Basic 2010

Sistem	Syarat Minimal	Syarat yang direkomendasikan
Arsitektur	X86 dan x64 (WOW)	
Sistem Operasi	Microsoft Windows XP Service Pack 2 Microsoft Windows Server 2003 Windows Vista	
Prosesor	CPU 1.6 GHz (Giga Hertz)	Windows XP dan Windows Server 2003:CPU 2,2 GHz atau yang lebih tinggi. Windows Vista : CPU 2,4 GHz
RAM	Windows XP dan Windows Server 2003 384 MB (Mega byte) Windows Vista : 768 MB	RAM 1024 MB / 1 GB atau yang lebih besar.
Harddisk	Tanpa MSDN Ruang Kosong harddisk pada drive penginstalan 2 GB. Sisa ruang harddisk kosong 1 GB Dengan MSDN Ruang kosong harddisk pada drive penginstalan 3,8 GB (MSDN diinstal full) 2,8 GB untuk menginstal MSDN default. Kecepatan Harddisk 5400 RPM.	Kecepatan harddisk 7200 RPM atau yang lebih tinggi.
Display Layar	1024 x 768 display	1280 x 1024 display

(Sumber : Wahana Komputer ; 2010 : 2)

II.7. Keamanan Data

Keamanan data merupakan bagian dari perkembangan teknologi informasi. Ketika berpikir bahwa data yang dimiliki merupakan data yang sangat penting, semua berusaha untuk melindunginya agar jangan sampai jatuh ke tangan orang yang tidak bertanggung jawab. Tetapi buat sebagian orang, mereka justru tidak mengetahui sepenting apakah data yang mereka miliki. Karena ketidaktahuan tersebut, mereka baru menyadari bahwa data yang mereka miliki

sangat penting setelah mengalami kecurian data dan mengalami kerugian. Data di sini bisa bersifat umum tidak terbatas pada data digital saja, tetapi juga seperti data diri (ktp, ijasah, sertifikat, dan lain-lain). Data yang menyangkut informasi pribadi tidak seharusnya diumbar sembarang seperti pada blog, situs jejaring pertemanan, email, selebaran, fotokopi KTP di buang sembarangan dan lain-lain. (Andik Susilo ; 2010 : 59).

Masalah keamanan merupakan salah satu aspek terpenting dari sebuah sistem informasi. Masalah keamanan sering kurang mendapat perhatian dari para perancang dan pengelola sistem informasi. Masalah keamanan sering berada di urutan setelah tampilan, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting. Apabile mengganggu performansi sistem, masalah keamanan sering tidak dipedulikan, bahkan ditiadakan. (Doni Ariyus; 2010:6).

Informasi menentukan hampir setiap elemen dari kehidupan manusia. Informasi sangat penting artinya bagi kehidupan karena tanpa informasi maka hampir semuanya tidak dapat dilakukan dengan baik. Contohnya, jika membeli tiket penerbangan dan membayarnya dengan menggunakan kartu kredit, informasi mengenai diri nantinya disimpan dan dikumpulkan serta digunakan oleh bank dan penerbangan. Demikian juga halnya saat membeli obat di apotik. Harus mendapat resep dari dokter dan memberikan resep tersebut ke pelayan apotik. Resep itu merupakan satu informasi yang disampaikan dokter ke pihak apotik tentang obat yang dibutuhkan.

Kemajuan sistem informasi memberikan banyak keuntungan bagi kehidupan manusia. Meski begitu, aspek negatifnya juga banyak, seperti kejahatan komputer yang mencakup pencurian, penipuan, pemerasan, kompetisi,

dan banyak lainnya. Jatuhnya informasi ke pihak lain, misalnya lawan bisnis, dapat menimbulkan kerugian bagi pemilik informasi. Sebagai contoh, banyak informasi milik perusahaan yang hanya boleh diketahui oleh orang-orang tertentu di perusahaan tersebut, seperti misalnya informasi tentang produk yang sedang dalam pengembangan. Algoritma dan teknik yang digunakan untuk menghasilkan produk tersebut. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas tertentu.

II.7.1. Ancaman Keamanan

Terjadi banyak pertukaran informasi setiap detiknya di internet. Juga banyak terjadi pencurian atas informasi oleh pihak-pihak yang tidak bertanggung jawab. Ancaman keamanan yang terjadi terhadap informasi adalah :

1) *Interruption*

   Merupakan ancaman terhadap *availability* informasi, data yang ada dalam sistem komputer dirusak atau dihapus sehingga jika data atau informasi tersebut dibutuhkan maka pemiliknya akan mengalami kesulitan untuk mengaksesnya, bahkan mungkin informasi itu hilang.

2) *Interception*

   Merupakan ancaman terhadap kerahasiaan (*secrecy*). Informasi disadap sehingga orang yang tidak berhak dapat mengakses komputer di mana informasi tersebut disimpan.

3) *Modification*

   Merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu lintas informasi yang sedang dikirim dan kemudian mengubahnya sesuai keinginan orang tersebut.

4) *Fabrication*

Merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru atau memalsukan informasi sehingga orang yang menerima informasi tersebut menyangka bahwa informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi. (Doni Ariyus; 2010:6).

II.7.2. Aspek-aspek Keamanan Komputer

Keamanan komputer meliputi empat aspek, antara lain :

1) *Authentication*

Agar penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi. Dengan kata lain, informasi itu benar-benar datang dari orang yang dikehendaki.

2) *Integrity*

Keaslian pesan yang dikirim melalui jaringan dan dapat dipastikan bahwa informasi yang dikirm tidak dimodifikasi oleh orang yang tidak berhak.

3) *Non Repudiation*

Merupakan hal yang berhubungan dengan si pengirim. Pengirim tidak dapat mengelak bahwa dialah yang mengirim informasi tersebut.

4) *Authority*

Informasi yang berada pada sistem jaringan tidak dapat dimodifikasi oleh pihak yang tidak berhak untuk mengaksesnya.

5) *Confidentiality*

Merupakan untuk menjaga informasi dari orang yang tidak berhak mengakses.

6) *Privacy*

Lebih ke arah data-data yang bersifat pribadi.

7) *Availability*

Aspek availabilitas berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi

8) *Access Control*

Aspek ini berhubungan dengan cara pengaturan akses ke informasi. Hal ini biasanya berhubungan dengan masalah otentikasi dan privasi. Kontrol akses seringkali dilakukan dengan menggunakan kombinasi *user id* dan *password* dengan mekanisme lain. (Doni Ariyus ; 2010 : 6).

II.8. Data

Setiap pengguna komputer biasanya punya data pribadi yang disimpan di komputer atau di media penyimpanan mobile macam *flashdisk*. Data itu bisa terkait dengan kegiatan pribadi misalnya, imil, riwayat jelajah di internet, atau terkait dengan urusan kerja. Setiap orang tentulah punyak kriteria sendiri tentang data vital pribadinya.

Bagaimana cara menyimpan dan melindungi data pribadi itu pun mungkin berbeda-beda antara satu pengguna dengan lainnya. Namun, pada umumnya para pengguna menyimpan data pribadinya itu dengan cara menempatkannya dalam *folder* tersendiri di partisi tertentu pada harddisk atau di *flashdisk*. Bahkan tak sedikit pengguna yang melatakkan data pentingnya di *My Documents*.

Tentuk saja cara penempatan *file* penting macam itu bisa aman-aman saja sejauh komputer atau flashdisk yang ditempati data itu digunakan sendiri, dan tak mungkin hilang. Dalam keadaan tertentu terpaksa meminjamkan komputer atau *flashdisk* kepada orang lain. Pada kondisi macam ini, tentulah data penting berpotensi atau dilirik orang yang dipinjami. (Bambang P Putranto ; 2010 : 1).