BAB II

TINJAUAN PUSTAKA

II.1. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah: program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

Aplkasi (application) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu , misalnya Microsoft Word, Microsoft Ecxel. Dan aplikasi adalah sebuah perangkat lunak yang menjadi front end dalam sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang berguna bagi orang-orang dan sistem yang bersangkutan.

Berdasarkan beberapa pengertian aplikasi, penulis dapat menyimpulkan apliksai adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya. (Desi Permata sari, 2011)

II.2. *Folder*

Sebuah *folder*, yang nama lainnya adalah direktori, merupakan suatu kontainer yang dapat digunakan untuk menyimpan file. Selain file, suatu *folder* juga dapat menampung *folder* lain yang disebut dengan *subfolder*. Akibat adanya *folder* dengan *subfolder*-nya ini, dapat terbentuk suatu tree dengan seluruh parent-nya merupakan *folder*.

Pada sistem operasi Linux dan turunan Unix lainnya, segala sesuatu dalam sistem dianggap sebagai file, termasuk direktori. Direktori menjadi suatu file khusus yang mengandung daftar dari nama-nama file beserta node masing-masing. Direktori memiliki peran penting dalam sistem file yang hierarkis pada sistem operasi komputer modern dengan mengizinkan pengelompokan direktori dan file untuk mengatur sistem file dalam hierarki yang modular. (Rinaldi Munir, 2011)

II.3. File

File adalah entitas dari data yang disimpan didalam sistem file yang dapat diakses dan diatur olrh prngguna. Sebuah file memiliki nama yang unik dalam direktori di mana ia berada. Alamat direktori dimana suatu berkas ditempatkan/ diistilahkan *path*. Sebuah file berisi aliran data (atau data stream) yang berisi sekumpulan data yang saling berkaitan serta atribut berkas yang disebut dengan properties yang berisi informasi mengenai file yang bersangkutan seperti informasi mengenai kapan sebuah berkas dibuat. (Februriyanti H, 2012)

II.4 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.

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.

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. (Andik Susilo, 2012)

II.5. Kriptografi

Kriptografi berasal dari bahasa Yunani, crypto dan graphia. Crypto berarti rahasia dan graphia berarti tulisan. Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Teknik penyandian data (kriptografi) yang diterapkan pada data maupun informasi, dilakukan dengan mengkodekan atau menyembunyikan data aslinya.

Dalam kriptografi, pesan yang akan dirahasiakan disebut plainteks dan pesan yang sudah diacak disebut cipherteks.

Ada empat tujuan mendasar dari ilmu kriptografi yan juga merupakan aspek keamanan informasi yaitu:

1. Kerahasiaan

   Layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

2. Integritas Data

Berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data dari pihak-pihak yang tidak berwenang, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. Autentikasi

Berhubungan dengan identitas/ pengenalan, baik secara kesatuan sistem atau informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri.

4. Non repudiasi

Tidak ada penyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat. (Hengky Mulyono, 2013)

II.5.1 Sistem Kriptografi Klasik

Sistem kriptografi klasik umumnya telah digunakan jauh sebelum era komputer. Kriptografi klasik juga dibagi menjadi dua jenis cipher yaitu cipher transposisi yang mengubah susunan huruf - huruf di dalam pesan dan cipher substitusi yang mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain. Kriptografi klasik, teknik enkripsi yang digunakan adalah enkripsi simetris dimana kunci dekripsi sama dengan kunci enkripsi. Penyandian ini berorientasi pada karakter.

Terdapat 5 bagian dalam sistem kriptografi klasik yaitu:

1. Plainteks

   Pesan atau data dalam bentuk aslinya yang dapat dibaca dan masukan bagi algoritma enkripsi.

2. *Secret Key*

   Masukan bagi algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi.

3. Cipherteks

   Hasil dari proses algoritma enkripsi dan teks asli dianggap telah tersembunyi.

4. Algoritma Enkripsi

   Algoritma enkripsi memiliki 2 masukan yaitu teks asli dan kunci rahasia, kedua masukan tersebut akan diproses sehingga menghasilkan teks sandi.

5. Algoritma Dekripsi

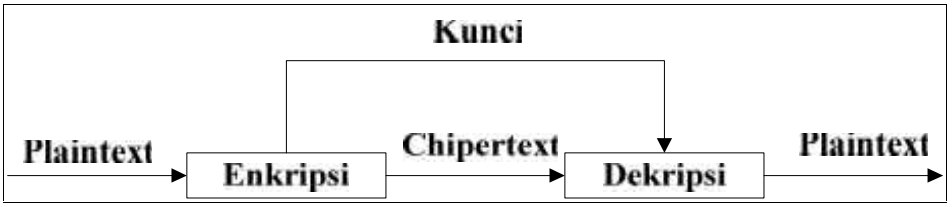
   Algoritma dekripsi memiliki 2 masukan yaitu teks sandi dan kunci rahasia, keduanya akan diproses sehingga menghasilkan teks asli. (Dony Ariyus, 2008)

II.5.2. Sistem Kriptografi Modern

Sistem kriptografi modern umumnya berorientasi pada bit. Untuk public key cryptography, diperlukan teknik enkripsi asimetris dimana kunci dekripsi tidak sama dengan kunci enkripsi. Enkripsi, dekripsi dan pembuatan kunci untuk teknik enkripsi asimetris memerlukan komputasi yang lebih intensif dibandingkan enkripsi simetris, karena enkripsi asimetris menggunakan bilangan - bilangan yang sangat besar. Beberapa mekanisme yang berkembang pada kriptografi modern :

1. Penyandian dengan kunci simetrik (*symmetric key encipherment*).

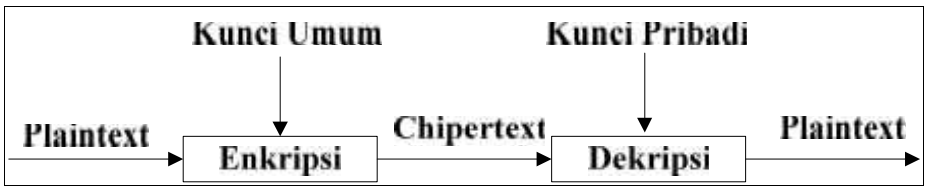
   Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai sama, dan masih digunakan pada kriptografi modern. Skema penyandian ini dapat digambarkan pada Gambar II.1.



Gambar II.1. Sistem Kriptografi Simetrik (*Sumber : Dony Ariyus, 2008*)

2. Penyandian dengan kunci asimetrik (asymmetric key encipherment)

   Penyandian dengan kunci asimetrik yang disebut juga dengan kunci publik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai berbeda. Penyandian ini yang banyak dikembangkan. Skema penyandian ini dapat digambarkan pada Gambar II.2.



Gambar II.2. Sistem Kriptografi Asimetrik (*Sumber : Dony Ariyus, 2008*)

Tidak seperti sistem kriptografi klasik di mana setiap entitas harus saling mengetahui kunci rahasia, sistem kriptografi modern yang juga disebut kriptografi

kunci asimetrik, memiliki dua jenis kunci, yaitu kunci enkripsi dan kunci dekripsi yang berbeda.

Dalam kriptografi kunci asimetris, hampir semua algoritma kriptografinya menggunakan konsep kunci publik, kecuali algoritma Pohlig - Hellman karena kunci enkripsi maupun kunci dekripsinya bersifat privat. (Dony Ariyus, 2008)

II.6. Algoritma One Time Pad

One-time pad (OTP) adalah stream cipher yang melakukan enkripsi dan dekripsi satu karakter setiap kali. Algoritma ini ditemukan pada tahun 1917 oleh Major Joseph Mauborgne sebagai perbaikan dari Vernam cipher untuk menghasilkan keamanan yang sempurna. Mauborgne mengusulkan penggunaan one-time pad (pad = kertas bloknot) yang berisi deretan karakter-karakter kunci yang dibangkitkan secara acak. Satu pad hanya digunakan sekali (*one-time*) saja untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain.

Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci one-time pad :

$ci = (pi + ki) \bmod 26$

Jika karakter yang digunakan adalah anggota himpunan 256 karakter (seperti karakter dengan pengkodean ASCII), maka persamaan enkripsinya menjadi: $ci = (pi + ki) \bmod 256$

Setelah pengirim mengenkripsi pesan dengan kunci, ia menghancurkan kunci tersebut.

Penerima pesan menggunakan pad yang sama untuk mendekripsikan karakter-karakter cipherteks menjadi karakter-karakter plainteks dengan persamaan:

pi = (ci – ki) mod 26 untuk alfabet 26-huruf, atau

pi = (ci – ki) mod 256 untuk alfabet 256-karakter.

Perhatikan bahwa panjang kunci harus sama dengan panjang plainteks, sehingga tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi (seperti halnya pada Vernam cipher).

Algoritma OTP ini tidak dapat dipecahkan (*unbreakable*) karena dua alasan:

1. Barisan kunci acak yang ditambahkan ke pesan plainteks yang tidak acak menghasilkan cipherteks yang seluruhnya acak. Cipherteks ini tidak mempunyai hubungan statistik dengan plainteks.
2. Karena cipherteks tidak mengandung informasi apapun perihal plainteks, maka tidak mungkin ada cara untuk memecahkan cipherteks. Beberapa barisan kunci yang digunakan untuk mendekripsi cipherteks mungkin menghasilkan plainteks yang mempunyai makna, sehingga kriptanalis tidak punya cara untuk menentukan plainteks mana yang benar. (Rinaldi Munir, 2011)

Secara teoritik, algoritma *one time pad* menggunakan kunci yang sama dalam proses enkripsi dan dekripsi serta memanfaatkan operasi xor pada proses enkripsi maupun dekripsi.

Tabel II.1. Operasi XOR

Bit Plaintext	Bit Kunci	Bit Hasil
0	0	0
1	0	1
0	1	1
1	1	0

Operasi xor menghasilkan nilai 0 apabila argumen sama (0 dengan 0 atau 1 dengan 1) dan menghasilkan 1 apabila argumen berbeda (0 dengan 1 atau 1 dengan 0). (Tomoyud S.Warawu, 2016)

II.7. *Microsoft Visual Basic* 2010

Pada akhir tahun 1999, Teknologi .NET diumumkan. Microsoft memposisikan teknologi tersebut sebagai *platform* untuk membangun XML Web *Services*. XML Web *services* memungkinkan aplikasi tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apapun melalui internet. Visual Basic.NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal .NET Framework.

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.2.

Tabel II.2. Sistem *Requirements* Visual Basic 2010

Sistem	Syarat Minimal	Syarat yang direkomendasikan
Arsitektur	X86 dan x64	
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)

II.7. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantic. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).(Sri Dharwiyanti, 2010)