BAB 2

LANDASAN TEORI

2.1 Teori Sistem Basis Data

2.1.1 Definisi Basis Data

Basis data merupakan kumpulan relasi *logical* dari data / deskripsi data yang dapat digunakan bersama dan dibuat untuk memperoleh informasi yang dibutuhkan.

Dengan menyimpan kumpulan informasi yang berkaitan dengan subjek atau tujuan tertentu seperti pendataan siswa dan guru menggunakan komputer dan dalam bentuk basis data, maka akan memudahkan penyimpanan, pengubahan, dan pencarian data dibanding menyimpan semuanya dalam arsip kertas yang dapat tercecer atau hilang.

2.1.2 Keuntungan Penggunaan Sistem Basis Data

Keuntungan menggunakan sistem basis data antara lain:

1. Penggunaan data bersama / *the data can be shared*
2. Mengurangi kerangkapan data / *redundancy can be reduced*
3. Menghindari ketidakkonsistenan data / *inconsistency can be avoided*
4. Integritas data terpelihara / *integrity can be maintained*
5. Keamanan terjamin / *security can be enforced*
6. Kebutuhan *user* yang kompleks dapat teratasi / *balanced conflicting requirements*

7. Pelaksanaan standarisasi / *standards can be enforced*
8. Meningkatkan produktivitas / *increased productivity*
9. Layanan *backup* dan *recovery* semakin baik / *improved backup and recovery services*

2.1.3 Microsoft Access Database

Secara definitif (*diambil dari Microsoft .Net Framework Glossary*), Microsoft Access Database merupakan sekumpulan data dan objek seperti misalnya tabel, *query*, atau *form* yang terkait dengan topik atau tujuan tertentu. Penataan data pada Microsoft Access Database dilaksanakan oleh Microsoft Jet Database Engine.

Microsoft Access Database merupakan salah satu format basis data yang paling banyak digunakan di seluruh dunia oleh individu, bisnis, maupun korporasi. Format basis data ini telah dan terus dikembangkan oleh perusahaan pembuatnya Microsoft dari tahun ke tahun.

2.1.4 Microsoft Jet Database Engine

Microsoft Jet Database Engine adalah bagian dari sistem basis data Microsoft Access yang berfungsi menerima dan menyimpan data dalam basis data perorangan dan sistem.

2.1.5 *Multi User Database*

Multi user database merupakan suatu basis data yang memungkinkan lebih dari satu orang pengguna memperoleh akses dan melakukan modifikasi terhadap kumpulan data yang sama pada saat bersamaan.

2.1.6 *Transaction*

Transaction adalah satu atau sekumpulan tindakan modifikasi terhadap data dalam basis data maupun struktur dari basis data itu sendiri, yang terkumpul dalam serangkaian tindakan yang dieksekusi secara beraturan agar dapat berjalan dengan lancar pada lingkungan *multi user database* oleh pengguna atau aplikasi. Penerapan *transaction* ditujukan untuk menghindari *data loss*, *corruption*, dan *collision* di lingkungan *multi user database*.

Pelaksanaan *transaction* memiliki dua hasil (*return value*). *Return value success* berarti *transaction* berhasil dilakukan dan disimpan. Basis data mencapai *consistent state* yang baru. *Return value failure* berarti *transaction* gagal (*aborted*) dan basis data dikembalikan (*rolled back* / *undone*) ke *consistent state* yang lama sebelum *transaction* dilakukan.

2.1.7 *Data Loss, Corruption, Collision*

Terdapat beberapa permasalahan yang dapat muncul dalam satu basis data terutama yang diimplementasikan dalam lingkungan *multi user*. *Data loss* adalah hilangnya sebagian atau keseluruhan data dari basis data. *Corruption* adalah terjadinya perubahan yang tidak diharapkan pada informasi data dan bersifat merugikan. *Collision* adalah tidak terlaksananya perubahan pada basis data akibat kurang baiknya penerapan *transaction* di lingkungan *multi user*.

2.1.8 *Relational Database*

Relational database adalah suatu format basis data yang berdasarkan pada satu model di mana semua data yang terlihat oleh pengguna disusun dalam bentuk tabel-tabel dan semua operasi pada basis data bekerja pada tabel-tabel tersebut.

Dalam tipe basis data ini, terdapat *primary key* dan *foreign key*. *Primary key* adalah *candidate key* yang dipilih untuk mengidentifikasikan baris secara unik pada suatu table. Setiap tabel hanya memiliki sebuah *primary key* dan karena digunakan sebagai pengenal, maka *primary key* harus memiliki nilai (tidak boleh *null*).

Foreign key adalah suatu atribut atau kumpulan atribut dalam suatu relasi yang menghubungkan suatu relasi ke relasi lain. Dalam menghubungkan suatu relasi ke relasi lain akan melibatkan atribut yang sama dari kedua relasi tersebut. Atribut yang sama tersebut biasanya merupakan *primary key* dari suatu tabel dan *foreign key* dari tabel yang lainnya. *Referential Integrity Rule* menyatakan bahwa *foreign key* dari suatu relasi akan mengacu pada *primary key* dari relasi yang lain. *Foreign key* juga dapat mengacu pada *primary key* dari relasi yang sama.

Ada tiga jenis relasi antar *record* dalam tabel yaitu:

1. Relasi *one to one*

   Yaitu relasi antara satu *record* dengan satu *record* dalam tabel yang saling berhubungan.

2. Relasi *one to many*

   Yaitu relasi antara satu *record* dengan lebih dari satu *record* dalam tabel yang saling berhubungan.

3. Relasi *many to many*

   Yaitu relasi antara beberapa *record* dengan lebih dari satu *record* dalam tabel yang saling berhubungan.

2.1.9 *Structured Query Language* (SQL)

Structured Query Language adalah suatu *non procedural language* yang dirancang secara spesifik untuk operasi pengaksesan data pada struktur *relational database* yang sudah dinormalisasi. Dengan menggunakan SQL dapat dibuat struktur-struktur basis data dan relasi antara tabel-tabelnya. Operasi pengaksesan data meliputi penyisipan data (*insert*), pengubahan data (*update*), pemilihan data (*select*), dan penghapusan data (*delete*).

Idealnya, bahasa pemrograman basis data harus melaksanakan operasi-operasi tersebut dengan usaha minimal dari user dan sintaks / struktur instruksi harus mudah dipahami dan dipelajari.

SQL disebut *non procedural language* karena pengguna cukup menspesifikasikan informasi apa yang dibutuhkan daripada bagaimana mendapatkannya.

2.1.10 ActiveX Database Object (ADO), OLE, dan Keamanan Informasi

ActiveX telah menjadi standar integrasi bagi perangkat lunak yang berjalan di bawah sistem operasi Microsoft Windows seperti Microsoft Office, Corel Suite, dan Adobe. Dengan adanya standar ActiveX ini, dokumen yang dihasilkan oleh perangkat lunak A dapat diakses dan dimodifikasi melalui perangkat lunak B dan sebaliknya. Sebagai contoh, seorang pengguna kini dapat dengan mudah menyisipkan gambar yang disunting di Adobe Photoshop ke dalam dokumen yang sedang ia susun di Microsoft Word. Dengan hadirnya ActiveX muncul pula standar OLE (*Object Linking and Embedding*) yaitu suatu standar yang memungkinkan dokumen yang dihasilkan suatu program untuk dapat dibuka di program lain tanpa perlu memiliki program pembuatnya.

Implementasi dukungan ActiveX ke dalam fungsi basis data khususnya pada Microsoft Access yang dikenal dengan ActiveX Database Object memungkinkan basis data Microsoft Access untuk dapat diakses oleh perangkat lunak lain tanpa harus memiliki Microsoft Access di dalam komputernya. Bila basis data yang dibuat di Microsoft Access diberi *password* maka perangkat lunak lain yang memanggil dokumen tersebut juga akan meminta pengguna memasukkan *password* yang benar ketika mengaksesnya. Hal ini kemudian mendasari dikembangkannya beragam algoritma yang bertujuan membuka basis data yang dilindungi oleh *password*. Penyalahgunaan aplikasi yang ditujukan untuk memulihkan *password* yang hilang dari suatu basis data untuk memperoleh informasi

secara tidak sah sudah sering terjadi karena apliaksi semacam ini sudah dibuat dan dapat diperoleh dengan mudah di toko-toko yang menjual perangkat lunak atau melalui internet.

Umumnya, program-program semacam ini bekerja dengan cara mengakses aplikasi yang digunakan untuk membuat dokumen tersebut melalui kemampuan OLE dari ActiveX. Ketika diminta untuk memasukkan *password* maka program ini akan mencoba terus beraneka ragam kemungkinan hingga menemukan *password* yang benar dan dokumen tersebut dapat dibuka. Metoda seperti ini disebut *exhaustive* / *brute force attack*. Metoda lain yang dikenal adalah *dictionary attack* atau *known word attack* yang pada intinya hampir sama dengan metoda sebelumnya tetapi metoda ini mengandalkan kata-kata yang telah tersimpan sebelumnya untuk dicoba sebagai *password*. Metoda *dictionary attack* ini dapat diumpamakan seorang yang mencari-cari kata dalam kamus untuk menemukan jawaban yang benar dari sebuah pertanyaan kosa kata. Kedua metoda ini juga dikenal dengan metoda *knocking* (mengetuk) yang artinya mencoba terus untuk menemukan *password* yang benar.

2.1.11 *Password* dan *Pass Sentence*

Password atau kata kunci pada awalnya adalah kata sandi agar seseorang dapat masuk ke satu tempat khusus yang memang hanya diperuntukkan untuk sekelompok orang dan hanya terdiri atas satu kata. Dalam kaitannya dengan dunia komputer dan terutama basis data,

password digunakan untuk menentukan apakah seseorang memiliki hak untuk mengakses suatu informasi tertentu.

Pass sentence merupakan kumpulan kata yang membentuk kalimat yang digunakan dengan tujuan yang sama dengan kata kunci. Penggunaan lebih dari satu kata akan menyulitkan serangan dengan teknik *dictionary attack*. Tetapi dalam penggunaan secara umum, istilah *pass sentence* jarang digunakan karena istilah *password* dianggap mempunyai pengertian yang mencakup *pass sentence* itu sendiri.

2.1.12 Microsoft Visual Basic

Microsoft Visual Basic merupakan salah satu dari bahasa pemrograman aras tinggi (*high level programming language*) yang memungkinkan perancangan aplikasi berbasis Microsoft Windows 32 bit dan aplikasi *web*. Pembuatan aplikasi didukung dengan perangkat visual yang memudahkan programmer dalam merancang tatap muka sehingga dapat lebih berkonsentrasi pada esensi aplikasi yang dibuat. Dukungan integrasi dengan berbagai aplikasi lain berbasis Microsoft Windows lainnya merupakan salah satu daya tarik yang menyebabkan Microsoft Visual Basic menjadi bahasa pemrograman yang banyak digunakan di seluruh dunia mengingat sistem operasi yang paling luas dipakai saat ini adalah Microsoft Windows.

Pada awalnya bahasa pemrograman ini dikembangkan oleh Microsoft Corporation untuk menghasilkan aplikasi berbasis MS-DOS (*Microsoft Disk Operating System*) dan hadir dalam beberapa versi seperti

Microsoft Basic, Microsoft QBasic, dan Microsoft Quick Basic. Perusahaan saingan Microsoft seperti Borland Corporation juga sempat mengembangkan bahasa Basic menggunakan nama Turbo Basic. Ketika perkembamgan sistem operasi mengalami revolusi menjadi GUI (*Graphic User Interface*) *oriented* maka lahirlah berbagai macam bahasa pemrograman berbasis visual seperti Visual Basic, Visual C/C++, dan Delphi (yang sering disebut Visual Pascal).

2.2 Teori Kriptografi

2.2.1 Definisi Kriptografi dan Peristilahan Lainnya

Kriptografi sudah mulai ada jauh sebelum adanya komputer. Pada masa lampau, kriptografi erat kaitannya dengan ilmu bahasa. Kini penekanan kriptografi ada pada matematika terutama matematika diskrit. Tetapi dari dulu sampai sekarang, kriptografi merupakan bagian penting dari keamanan data.

Kriptografi (*cryptography*) secara umum merupakan ilmu atau seni penyandian informasi agar tidak dapat dimengerti oleh pihak yang tidak diinginkan. Menurut ensiklopedia *online* www.wikipedia.com, kriptografi berasal dari 2 (dua) kata dalam bahasa Yunani (Grika) yaitu *kryptós* yang berarti tersembunyi dan *gráphein* yang berarti menulis atau tulisan. Kata ini memiliki makna mengubah informasi dari bentuk normal yang dapat dimengerti menjadi bentuk yang tidak dapat dimengerti tanpa memiliki rumus rahasia tertentu.

Kriptoanalisis (*cryptanalysis*) merupakan cabang ilmu yang mempelajari cara membuka hasil penyandian untuk mendapatkan naskah asli.

Kriptologi (*cryptology*) merupakan cabang ilmu yang mewadahi kriptografi dan kriptoanalis.

Kriptoanalis (*cryptanalyst*) merupakan individu yang menekuni bidang kriptoanalisis.

Kriptosistem (*cryptosystem*) adalah sistem yang berisi serangkaian langkah kriptografi yang ditujukan untuk menjaga keamanan informasi.

Enkripsi (*encription*) merupakan proses penyandian informasi menggunakan teknik kriptografi.

Dekripsi (*decryption*) merupakan proses pembukaan sandi hasil enkripsi menggunakan teknik kriptografi.

Plaintext merupakan informasi asli / asal.

Ciphertext merupakan hasil ekripsi dari *plaintext*.

Algoritma kriptografi (*cipher*) adalah algoritma yang berisi fungsi matematika yang dipergunakan untuk melakukan enkripsi dan dekripsi.

Public Key merupakan kode publik yang digunakan untuk melakukan enkripsi atau dekripsi.

2.2.2 Tujuan Penerapan Kriptografi

Tujuan diterapkannya kriptografi adalah untuk mengamankan data. Yang ingin dicapai dengan prosedur pengamanan data ini adalah:

1. *Confidentiality* atau kerahasiaan.

   Hanya orang yang berhak terhadap informasi yang dapat mengakses informasi tersebut. Jika orang yang tidak berhak mendapatkan informasi tersebut, maka yang dapat ia lihat hanyalah kumpulan karakter yang tidak berarti.

2. *Integrity* atau integritas data

   Informasi yang ada harus terlindungi agar tidak dapat diubah oleh pihak lain dan dapat mengetahui apakah informasi yang ada sudah diubah atau belum.

3. *Authentication* atau keaslian data

   Setiap orang yang berhak mengakses informasi bersangkutan harus dapat mengetahui apakah informasi tersebut disediakan oleh orang lain yang juga berhak. Poin ini berkaitan erat dengan integritas data.

4. *Non-repudiation*

   Berkaitan dengan keaslian data, orang yang melakukan *entry* atau mengubah informasi tidak dapat mengelak karena terdapat bukti bahwa orang bersangkutan telah melakukan *entry* atau perubahan pada informasi bersangkutan.

2.2.3 Kriptografi Sederhana: *Caesar Shift Cipher*

Shift Cipher atau *Caesar Cipher* atau *Caesar Shift Cipher* dipercaya merupakan salah satu algoritma kriptografi yang paling awal yang dipergunakan. Teknik penyandian data ini dipercaya dipergunakan untuk meneruskan pesan pada saat perang sehingga kalaupun pengirim pesan tertangkap, pesan yang dia bawa tidak berguna bagi pihak musuh.

Metode penyandian ini dilakukan dengan mengganti huruf yang ada dengan huruf lain yang berjarak a dari huruf tersebut. Secara matematis, kita berikan nilai kepada huruf tersebut menggunakan deret aritmatik tertutup sederhana dan menambahkan a kepada x (nilai huruf yang mau digantikan tersebut). Untuk menghindari a + x yang tidak memiliki koresponden dalam deret tersebut karena terlalu besar, maka setelah nilai huruf terakhir dilewati, a + x berulang ke awal deret. Jika deret dibuat dengan huruf pertama diberi nilai 0 dan hiruf terakhir diberi nilai b maka rumusan mendapatkan *ciphertext* atau f(x) dapat dituliskan sebagai berikut:

$$f(x) = (x + a) \bmod (b+1)$$

Contoh penggunaan metode ini pada huruf alfabet dapat dilihat pada tabel berikut ini.

Huruf	Nilai Huruf x	Nilai Huruf Pengganti f(x) = (x+5) mod 26	Huruf sandi
A	0	5	F
B	1	6	G
C	2	7	H
D	3	8	I
E	4	9	J
F	5	10	K
G	6	11	L
H	7	12	M
I	8	13	N
J	9	14	O
K	10	15	P
L	11	16	Q
M	12	17	R
N	13	18	S
O	14	19	T
P	15	20	U
Q	16	21	V
R	17	22	W
S	18	23	X
T	19	24	Y
U	20	25	Z

V	21	0	A
W	22	1	B
X	23	2	C
Y	24	3	D
Z	25	4	E

Tabel 2.1 Contoh Tabel *Shift Cipher*

Berdasarkan tabel di atas, maka jika kita menyandikan kata ‘kemerdekaan’ maka kita akan mendapatkan kata ‘pjrjwijpffs’ yang tidak memiliki arti apapun.

2.2.4 Kriptografi Polialfabetik: *Vigenere Chiper* dan Variannya

Masalah utama dengan kriptografi sederhana adalah mudahnya pesan dibongkar kembali. Salah satunya adalah dengan cara *Frequency Analysis* atau analisis tingkat kemunculan. Dengan melihat simbol yang paling sering muncul dan kemudian menggantinya dengan huruf yang sering muncul pada bahasa bersangkutan (misalnya huruf ‘a’ pada Bahasa Indonesia) maka dengan mudah kita dapat melihat gambaran pesan aslinya. Karena itu, untuk meningkatkan keamanan dari metoda kriptografi, para ahli berusaha mengembangkan model kriptografi yang sulit dipecahkan dengan metoda *Frequency Analysis* ini.

Salah satu caranya adalah dengan menggunakan metoda *Polyalphabetic Cipher*. Jika pada kriptografi sederhana huruf asal dan simbol akhir memiliki relasi satu-satu (satu huruf disandikan dengan satu

simbol) maka pada kriptografi polialfabetik, satu huruf dapat diwakilkan oleh beberapa simbol.

Salah satu metoda kriptografi polialfabetik yang paling terkenal adalah *Vigenere Cipher* yang diusulkan oleh Blaise de Vigenere pada abad XVI. Kunci penggunaannya ada pada kata kunci yang dipetakan pada tabel huruf yang diberi nama *Tableau de Vigenere* seperti di bawah ini.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Tabel 2.2 *Tableau de Vigenere*

Untuk dapat menggunakannya, kita memerlukan kata kunci. Misalkan kata kuncinya adalah ‘cintaku’ dan kalimat yang mau disandikan adalah ‘besokmaumakanmalamtidak’ maka proses penyandiannya adalah proses mencocokkan huruf dari *plaintext* (kolom) dan huruf kata kunci (baris) untuk mendapatkan kombinasi yang sesuai. Prosesnya sendiri dapat digambarkan serupa dengan ini:

Plaintext	Kata Kunci	*Ciphertext*
B	C	D (Kolom B, Baris C)
E	I	M (Kolom E, Baris I)
S	N	F (Kolom S, Baris N)
O	T	H (Kolom O, Baris T)
K	A	A (Kolom K, Baris A)
M	K	W (Kolom M, Baris K)
A	U	U (Kolom A, Baris U)
U	C	W (Kolom U, Baris C)
M	I	U (Kolom M, Baris I)
A	N	N (Kolom A, Baris N)
K	T	D (Kolom K, Baris T)
A	A	A (Kolom A, Baris A)
N	K	X (Kolom N, Baris K)
M	U	G (Kolom M, Baris U)
A	C	C (Kolom A, Baris C)
L	I	T (Kolom L, Baris I)

A	N	N (Kolom A, Baris N)
M	T	F (Kolom M, Baris T)
T	A	T (Kolom T, Baris A)
I	K	S (Kolom I, Baris K)
D	U	X (Kolom D, Baris U)
A	C	C (Kolom A, Baris C)
K	I	S (Kolom K, Baris I)

Tabel 2.3 Contoh Tabel *Vigenere Cipher*

Proses pengembalian menggunakan cara yang sama dengan proses penyandian dan hanya dapat dilakukan orang yang mengetahui kata kuncinya. Perhatikan bahwa setiap kolom merupakan teknik penyandian dengan *shift cipher* mulai dari x+0, x+1 mod 26, terus hingga x+25 mod 26. Akibatnya jika kita menggunakan kata kunci yang memiliki huruf seragam seperti ‘xxx’ maka sama saja kita menggunakan *shift cipher* dengan rumus x+23 mod 26.

Karena itu, metoda ini juga dapat dinotasikan secara matematis dengan mengubah kata kunci menjadi vektor. Jumlah huruf pada kata kunci menjadi panjang vektor dan setiap huruf menjadi satu nilai. Seperti pada contoh di atas, kata kunci ‘cintaku’ dapat diganti menjadi vektor v=(2, 8, 13, 19, 0, 10, 20). Kemudian setiap huruf dari *plaintext* yang bersesuaian digeser menggunakan teknik *Shift Cipher* sebesar vektor yang berkoresponden dengan huruf tersebut.

Vigenere Cipher dianggap merupakan metoda kriptografi yang tidak mungkin terpecahkan selama 300 tahun hingga akhirnya pada tahun 1863 Kasiski menemukan metoda memecahkannya dan diteruskan oleh Kerckhoff (metodanya bernama Kasiski-Kerckhoff).

Dalam penulisan skripsi ini, *Vigenere Cipher* ini digunakan dengan panjang vektor 16 yang dibagi menjadi 4 bagian dan disusun seolah-olah menjadi suatu matriks. Bentuk matriks dipilih untuk memudahkan proses penyandian tingkat berikutnya. Seperti dibahas di atas, vektor (yang akan diganti cara representasinya dengan menggunakan matriks) *Vigenere Cipher* ini menggunakan angka kunci berjumlah 16 buah yang akan dimodifikasi menjadi 16 buah fungsi.

$$f(x) = (ax + b) \bmod c$$

Dalam menentukan besarnya a dan b kita harus kembali melihat *input* yang dibutuhkan. Untuk *input* dari *keyboard* maka kita dapat menyimpulkan bahwa nilai x berkisar dari 48 hingga 122. Sedangkan kita mengetahui karakter ASCII berkisar dari 1 hingga 255. Karakter ASCII 1 akan kita pergunakan untuk mengisi bagian yang kosong dari matriks penyandian sehingga kita harus memasukkan nilai a dan b sedemikian rupa sehingga $2 < f(x) < 255$.

Fungsi yang dipergunakan dalam penulisan ini antara lain (disusun dalam bentuk matriks):

| x+3 | x+6 | x+9 | x+12 |

| 2x-5 | 2x-10 | 2x-15 | 2x-20 |

| x+8 | x+16 | x+24 | x+32 |

| 2x-9 | 2x-18 | 2x-27 | 2x-36 |

Maka jika kita menyusun ulang proses penyandian menggunakan *Vigenere Cipher* di atas dengan menggunakan matriks dan rumus enkripsi yang kita buat, maka akan tampak seperti contoh di bawah ini:

| b e s o | | a m t i |

| k m a u | | d a k ☺ |

| m a k a | | ☺ ☺ ☺ ☺ |

| n m a l | | ☺ ☺ ☺ ☺ |

Dengan simbol ☺ merupakan karakter ASCII 1 (alt+1) dan digunakan untuk melengkapi bagian matriks yang kosong sehingga membentuk matriks 4x4. Dan proses penyandian menggunakan kode ASCII dari karakter *plaintext* dan rumus yang disediakan akan menjadi seperti:

| 98+3 | 101+6 | 115+9 | 111+18 |

| 2(107)-5 | 2(109)-10 | 2(97)-15 | 2(117)-20 |

| 109+8 | 97+16 | 107+24 | 97+32 |

| 2(110)-9 | 2(109)-18 | 2(97)-27 | 2(108)-36 |

dan

| 97+3 | 109+6 | 116+9 | 105+18 |

| 2(100)-5 | 2(97)-10 | 2(107)-15 | ☺ |

| ☺ | ☺ | ☺ | ☺ |

| ☺ | ☺ | ☺ | ☺ |

Sehingga matriks *ciphertext* akan menjadi seperti berikut:

101	107	124	129		e	k	ü
209	208	179	214	atau	╤	╨	╽	╓
117	113	131	129		u	q	â	ü
211	200	167	180		╙	╚	°	┤

dan

100	115	125	123		d	s	}	{
195	184	199	☺	atau	├	╕	╟	☺
☺	☺	☺	☺		☺	☺	☺	☺
☺	☺	☺	☺		☺	☺	☺	☺

2.2.5 *Transposition Cipher* dan Varian *User Dependant*-nya

Transposition Cipher merupakan metoda kriptografi yang sepintas kelihatan sederhana. Jika banyak metoda kriptografi menekankan pada menyamarkan huruf / karakter dengan simbol lain, maka metoda kriptografi ini mengganti urutan munculnya simbol tersebut. Kelebihannya adalah metoda ini tidak dapat dipecahkan dengan menggunakan teknik *Frequency Analysis*. Kunci penyandian untuk motoda ini juga tidak baku karena kita tidak memiliki fungsi matematis baku untuk memetakan setiap huruf. Yang ada adalah urutan pemetaan. Terdapat banyak varian *Transposition Cipher* menurut urutan pemetaannya. Varian yang akan digunakan pada penulisan makalah ini merupakan varian *user dependant* dimana kata kunci yang digunakan user untuk *login* akan menjadi kunci untuk menentukan transposisi.

Pada penyandian pertama, program akan menyandikan menggunakan *Vigenere Cipher* yang satu fasanya akan melibatkan 16 karakter atau simbol. Setiap 16 simbol ini akan disusun menjadi suatu matriks yang disusun berurutan. Setiap matriks tersebut kemudian akan menjadi satu bagian terpisah yang oleh program akan dievaluasi apakah akan disandikan sekali lagi menggunakan *user dependant transposition cipher* atau tidak.

Yang digunakan untuk mengevaluasi hal ini adalah kata kunci yang dimiliki oleh pihak yang terakhir kali mengubah basis data ini. Sebagai contoh, maka akan dimisalkan jika yang melakukan perubahan terakhir kali pada kalimat contoh ‘besokmaumakanmalamtidak’ adalah pengguna Agus Chiawono yang menggunakan kata ‘boss’ sebagai kata kunci untuk *login*. Kata kunci ini akan diubah ke bilangan ASCII-nya yaitu 98, 111, 115, dan 115. Keempat bilangan ini kemudian diubah ke dalam bilangan biner (basis 2) dan ditempelkan menjadi satu rangkaian bersambung yaitu 1100010110111111100111110011. Setiap angka 0 atau 1 mewakili satu matriks pada urutan data. Angka 0 menandakan matriks tersebut dibiarkan apa adanya dan angka 1 menandakan matriks tersebut dijadikan matriks transpos. Mengikuti aturan ini, maka matriks contoh akan menjadi:

| e ╤ u ╚ | | d ├ ☺ ☺ |

| k ╩ q╚ | dan | s ╕ ☺ ☺ |

| | ▌ â ° | | } ╟ ☺ ☺ |

| ü ╓ ü ┤ | | { ☺ ☺ ☺ |

Dan memiliki tiga kemungkinan lain tergantung kata kunci yang dimasukkan oleh pengguna yaitu:

Kemungkinan 1:

| e ㅠ u ㅚ | | d s } { |

| k ㅛ qㅚ | dan | ㅏ ㅋ ㅐ ☺ |

| | ㅣ â ° | | ☺ ☺ ☺ ☺ |

| ü ㅠ ü ㅓ | | ☺ ☺ ☺ ☺ |

Kemungkinan 2:

| e k | ü | | d ㅏ ☺ ☺ |

| ㅠ ㅛ ㅣ ㅠ | dan | s ㅋ ☺ ☺ |

| u q â ü | | } ㅐ ☺ ☺ |

| ㅚ ㅚ ° ㅓ | | { ☺ ☺ ☺ |

Kemungkinan 3:

| e k | ü | | d s } { |

| ㅠ ㅛ ㅣ ㅠ | dan | ㅏ ㅋ ㅐ ☺ |

| u q â ü | | ☺ ☺ ☺ ☺ |

| ㅚ ㅚ ° ㅓ | | ☺ ☺ ☺ ☺ |

Dengan adanya perbedaan jumlah penerapan lapisan enkripsi pada setiap bagian maka akan sulit sekali memecahkan metoda enkripsi ini.