Modifikasi *Vigenere Cipher* dengan Memodifikasi Kunci

Janice Laksana / 13510035
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510035@std.stei.itb.ac.id

***Abstract*—Vigenere Cipher merupakan algoritma kriptografi klasik yang termasuk pada kelompok cipher subtitusi abjad-majemuk. Algoritma ini telah berhasil dipecahkan dengan menggunakan metode Kasiski yang mengambil keuntungan dari kelemahan Vigenere Cipher yang mengenkripsi huruf-huruf plainteks yang berada pada posisi kelipatan panjang kunci menjadi huruf-huruf cipherteks yang sama. Karena algoritma ini dengan mudah dipecahkan, maka akan dilakukan modifikasi pada algoritma Vigenere Cipher ini sehingga keamanan dari algoritma ini dapat ditingkatkan dan kriptanalis pun menjadi sulit untuk memecahkan cipherteks hasil proses enkripsi. Modifikasi algoritma Vigenere Cipher yang dapat dilakukan salah satunya dengan memperbaiki metode perpanjangan kunci yang digunakan saat panjang kunci lebih pendek dari panjang plainteks yang dienkripsi.**

***Kata Kunci*— Algoritma Klasik, Vigenere Cipher, Kriptanalisis, peningkatan keamanan.**

I. PENDAHULUAN

Algoritma kriptografi klasik dapat dibagi menjadi dua kelompok yaitu Cipher subtitusi dan Cipher transposisi. Pada Cipher subtitusi, setiap unit pada plainteks akan diganti dengan satu unit cipherteks. Satu unit sendiri dapat didefinisikan sebagai satu huruf, satu pasangan huruf, ataupun satu kelompok yang terdiri lebih dari dua huruf. Cipher subtitusi dapat dibagi lagi menjadi empat jenis yaitu cipher abjad-tunggal, cipher subtitusi homofonik, cipher abjad-majemuk, dan cipher subtituis poligram. Pada Cipher transposisi, proses enkripsi dilakukan dengan mengubah posisi huruf plainteks terlebih dahulu. Selain cipher subtitusi dan cipher transposisi, dikenal pula cipher super-enkripsi yang menggabungkan cipher subtitusi dan cipher transposisi.

Algoritma Vigenere Cipher merupakan salah satu contoh algoritma klasik yang merupakan cipher abjad-majemuk. Algoritma ini dipublikasikan oleh seorang diplomat yang merupakan seorang kriptologis yaitu Blaise de Vigenere pada tahun 1586. Proses enkripsi pada algoritma Vigenere Cipher dilakukan dengan menggunakan bujur sangkar Vigenere.

Algoritma Vigenere ini sudah dapat dipecahkan dengan metode yang ditemukan oleh Babbage dan Kasiski dan dikenal sebagai metode Kasiski. Metode ini memanfaatkan kekurangan dari algoritma Vigenere Cipher di mana algoritma ini akan mengenkripsi huruf-huruf plainteks yang terletak pada posisi yang merupakan kelipatan panjang kunci menjadi huruf-huruf cipherteks yang sama. Tujuan utama dari metode ini sendiri adalah dengan untuk mengetahui panjang kunci enkripsi. Setelah panjang kunci enkripsi dapat diketahui, maka plainteks dapat ditemukan dengan menggunakan metode metode *exhaustive key search* dan metode analisis frekuensi kemunculan huruf.

Karena Vigenere Cipher bisa dengan mudah dipecahkan dengan metode Kasiski, penulis mencoba untuk meningkatkan keamanan algoritma Vigenere Cipher ini dengan memperbaiki kekurangan dari algoritma Vigenere Cipher. Modifikasi yang dilakukan berfokus pada perbaikan metode perpanjangan kunci yang digunakan pada saat panjang kunci lebih pendek dari panjang plainteks yang akan dienkripsi.

II. TEORI

A. Algoritma Kriptografi Klasik

Algoritma kriptografi klasik dahulu diciptakan tanpa penggunaan komputer. Algoritma ini diciptakan dengan hanya menggunakan pena dan kertas saja. Algoritma klasik ini dapat dibagi menjadi dua bagian yaitu Cipher subtitusi dan cipher transposisi.

Cipher subtitusi sendiri dapat dibagi lagi menjadi empat jenis yaitu cipher abjad-tunggal, cipher subtitusi homofonik, cipher abjad-majemuk, dan cipher subtitusi poligram. Pada cipher abjad-tunggal, satu huruf pada plainteks akan disubtitusi dengan satu huruf yang bersangkutan. Ppada cipher subtitusi homofonik, satu huruf pada plainteks dapat disubitutiskan dengan satu atau pasangan huruf yang bersangkutan. Tujuan dari cipher subtitusi homofonik ini adalah untuk menyamarkan hubungan statistik antara plainteks dengan cipherteks. Cipher abjad-majemuk, setiap huruf pada plainteks akan dienkripsi dengan kunci yang berbeda. Sedangkan pada cipher subtitusi poligram, enkripsi dilakukan tidak huruf per huruf tetapi per dua huruf (digram), per tiga huruf (trigram).

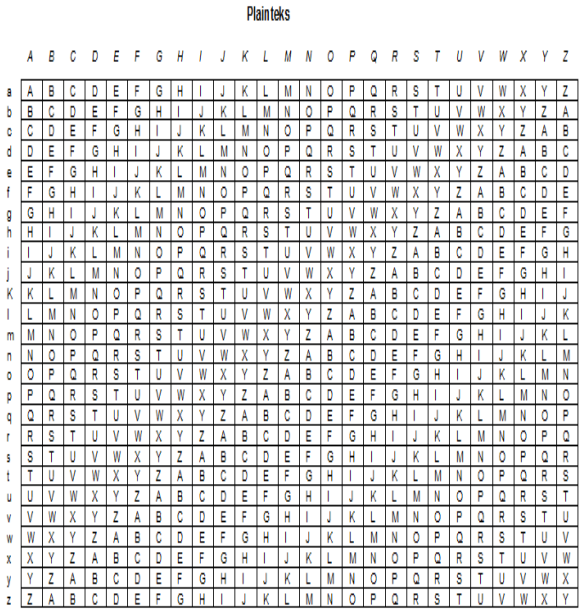
Cipher tranposisi dapat diperoleh dengan cara mengenkripsi plainteks yang sebelumnya sudah diubah posisi hurufnya. Pada algoritma klasik juga dikenal cipher super-enkripsi, yakni adalah cipher yang diperoleh dengan

menggabungkan metode pada cipher subtitusi dan metode pada cipher tranposisi.

B. Algoritma Vigenere Cipher

Algoritma Vigenere Cipher merupakan algoritma kriptografi klasik. Algoritma ini dipublikasikan oleh seorang diplomat yang merupakan seorang kriptologis yaitu Blaise de Vigenere pada tahun 1586. Sebenarnya algoritma ini sudah pernah digambarkan di dalam buku La Cifra del Sig oleh Giovan Batista Belaso.

Untuk melakukan enkripsi, algoritma Vigenere Cipher menggunakan bujur sangkar vigenere. Vigenere Cipher akan mengenkripsi setiap huruf pada plainteks dengan kunci yang berbeda sehingga algoritma Vigenere Cipher ini termasuk ke dalam cipher abjad majemuk.



Gambar 2.1 Bujur Sangkar Vigenere

Dapat dilihat pada bujur sangkar vigenere, indeks kolom merupakan plainteks dari a hingga z, sedangkan indeks baris merupakan kunci dari a hingga z.

Secara umum, algoritma vigenere cipher akan menggeser huruf-huruf plainteks sejauh kunci terkait. Jadi secara umum, enkripsi dilakukan :

$$c_i = (p + k_i) \bmod 26$$

k_i merupakan huruf ke-i dari kunci enkripsi yang juga menyatakan jumlah pergeseran plainteks pada huruf ke-i. Ketika panjang kunci lebih pendek dari panjang plainteks yang akan dienkripsikan, maka kunci akan diulang secara periodik.

Algoritma Vigenere Cipher ini memiliki kelebihan yakni algoritma ini dapat menyembunyikan hubungan statistik antara plainteks dan cipherteks.

C. Kriptanalis

Algoritma Vigenere telah berhasil dipecahkan pada abad ke 19 oleh Babbage dan Kasiski. Metode Kasiski ini membantu untuk menemukan panjang kunci yang digunakan dalam mengenkripsi sebuah plainteks menggunakan Algoritma Vigenere Cipher. Metode ini memanfaatkan kelemahan pada Vigenere Cipher yakni perulangan kelompok huruf pada plainteks memiliki kemungkinan untuk dienkripsi menjadi cipherteks yang sama apabila perulangan kelompok huruf ini memiliki jarak sebesar kelipatan panjang dari kunci yang digunakan. Contoh :

Plainteks :

CRYPTO IS SHORT FOR **CRYPTO**GRAPHY

Kunci :

abcdab cd abcda bcd abcdabcdabcd

Cipherteks :

CSASTP KV SIQUT GQU **CSASTP**IUAQJB

Dengan menemukan kelemahan ini, Kasiski membuat sebuah metode yang memiliki tujuan untuk mencari cipherteks yang berulang sehingga panjang kunci dapat ditemukan. Langkah-langkah dari metode Kasiski adalah :

1. Menemukan cipherteks-cipherteks berulang dari cipiherteks yang dihasilkan dari algoritma Vigenere Cipher.
2. Kemudian hitunglah jarak antar perulangan cipherteks tersebut.
3. Setelah itu, dari semua jarak tersebut, carilah faktor pembagi yang merupakan panjang kunci yang paling mungkin.
4. Terakhir tentukan irisan dari himpunan faktor pembagi tersebut sehingga diperoleh kemungkinan panjang kunci yang digunakan dalam proses enkripsi plainteks.

Setelah panjang kunci yang digunakan pada proses enkripsi diketahui, kata kunci dapat ditemukan dengan penggunaan metode *exhaustive key search* dan dengan teknik analisis frekuensi. Teknik analisis frekuensi merupakan metode yang lebih efektif dalam menentukan kata kunci. Langkah-langkah metode ini adalah:

1. Dengan mengetahui panjang kunci enkripsi, diketahui apabila setiap huruf yang terletak pada kelipatan panjang kunci enkripsi, pasti dienkripsi dengan menggunakan kunci yang sama. Jadi kelompokkan setiap huruf yang terletak pada "panjang kunci" secara bersama-sama sehingga diperoleh cipherteks sepanjang "panjang kunci" yang dienkripsi dengan menggunakan kunci yang sama (seperti pada subtitusi alfabet tunggal).
2. Setelah itu cipherteks yang tadi sudah dikelompokkan dapat dipecahkan dengan menggunakan teknik analisis frekuensi.
3. Setelah semua dipecahkan, huruf-huruf kunci dapat disusun.

III. MODIFIKASI VIGENERE CIPHER

Dapat dilihat cipherteks yang dihasilkan dari proses enkripsi dengan menggunakan algoritma Vigenere Cipher dapat dipecahkan dengan mudah. Sehingga penulis mencoba memperbaiki hal tersebut dengan mencoba

meneliti dahulu algoritma Vigenere Cipher.

Hal yang penulis ketahui adalah bahwa algoritma ini telah dipecahkan dengan metode Kasiski karena kekurangan yang dimiliki oleh algoritma ini yakni perulangan kelompok huruf pada plainteks memiliki kemungkinan untuk dienkripsi menjadi cipherteks yang sama apabila perulangan kelompok huruf ini memiliki jarak sebesar kelipatan panjang dari kunci yang digunakan. Dari pernyataan tersebut, dapat dilihat apabila cipherteks berulang diakibatkan oleh penggunaan kunci yang sama dan hal tersebut disebabkan karena pada algoritma ini telah ditentukan apabila panjang kunci lebih pendek dari panjang plainteks yang akan dienkripsikan, maka kunci akan diulang secara periodik. Sehingga penulis pun berfokus pada hal apa yang penulis harus lakukan pada saat panjang kunci lebih pendek dari panjang plainteks agar plainteks yang belum memiliki kunci akan memiliki kunci akan tetapi perpanjangan kunci ini tidak hanya dilakukan dengan mengulang kunci secara periodik.

Setelah itu penulis terinspirasi oleh algoritma kriptografi modern yang menggunakan cipherteks sebagai perpanjangan dari kunci yang lebih pendek dari plainteks. Jadi hal pertama yang penulis usulkan untuk memodifikasi algoritma Vigenere Cipher ini adalah saat enkripsi suatu plainteks dengan menggunakan algoritma Vigenere Cipher, ketika kunci yang digunakan memiliki panjang yang lebih pendek daripada panjang plainteks yang akan dienkripsi, maka kunci akan diperpanjang dengan cipherteks yang diperoleh dari enkripsi plainteks yang telah memiliki kunci, hingga panjang dari kunci telah sama dengan panjang plainteks. Setelah itu penulis terinsipirasi dengan metode yang digunakan pada enigma cipher yakni setelah sebuah huruf selesai disubstitusi, rotor akan bergerak satu huruf ke atas. Sehingga hal kedua pada modifikasi yang penulis lakukan pada algoritma Vigenere Cipher ini adalah sebelum penulis mengenkripsikan sebuah plainteks yang dienkripsi dengan kunci yang telah diperpanjang oleh cipherteks, semua elemen cipherteks pada bujur sangkar vigenere akan penulis geser sejauh kunci yang digunakan oleh plainteks yang akan dienkripsikan.

Jadi secara menyeluruh, modifikasi yang penulis akan lakukan pada algoritma Vigenere Cipher adalah dengan membuat suatu aturan yakni apabila kunci yang akan digunakan untuk mengenkripsi sebuah plainteks memliki panjang yang lebih pendek, maka kunci akan diperpanjang dengan menggunakan cipherteks yang diperoleh dari proses enkripsi pada plainteks yang telah memiliki kunci.

Contoh :

Plainteks : THIS IS PLAINTEXT
Kunci : SONY
Cipherteks : LVVQ

Panjang kunci yang akan digunakan pada proses enkripsi plainteks lebih pendek dari panjang plainteks yang akan dienkripsi. Dengan aturan yang telah disebutkan di atas, maka untuk mengenkripsikan huruf I pada IS akan digunakan huruf L dari cipherteks LVVQ , untuk mengenkripsikan huruf S pada IS akan digunakan huruf V pada cipherteks LVVQ, dst. Setelah itu proses enkripsi yang dilakukan pada plainteks yang akan dienkripsikan dengan key yang merupakan perpanjangan dari cipherteks berbeda. Sebelum proses enkripsi pada plainteks tersebut dilakukan, akan dilakukan dahulu per geseran bujur sangkar Vigenere Cipher sejauh huruf pada kunci yang digunakan. Contoh : untuk mengenkripsikan huruf I pada IS dengan kunci L, akan dilakukan dahulu penggeseran bujur sangkar Vigenere Cipher sejauh L.

Sebelum digeser :

	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

Setelah digeser sejauh L :

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

Metode yang penulis usulkan untuk memodifikasi Vigenere Cipher ini, penulis implementasikan dengan membuat program berbahasa Java. Dengan menggunakan metode ini, terdapat lima fungsi utama yang digunakan yakni :

1. Fungsi untuk mencari cipherteks dari bujur sangkar Vigenere.
2. Fungsi untuk mencari plainteks dari bujur sangkar Vigenere.
3. Fungsi untuk menggeser elemen-elemen pada bujur sangkar Vigenere.
4. Fungsi untuk mengenkripsi plainteks.
5. Fungsi untuk mendekripsi plainteks.

Untuk mencari cipherteks pada bujur sangkar Vigenere, berikut potongan programnya :

```
public int SearchRow(char char_key){
//Fungsi ini akan menerima input
berupa huruf key
//dan mengembalikan sebuah indeks
baris tempat key tersebut berada
    int num <- 0;
    for (int i =0; i<table26.length;
i++){
        if (table26[i][0]=char_key){
            num <- i;
        }
    }
    return num;
}
```

```
public int SearchColumnEnkripsi(char
char_plain){
//Fungsi ini akan menerima input
berupa huruf plainteks
//dan mengembalikan sebuah indeks
kolom tempat huruf plainteks
tersebut berada
    int num <- 0;
    for (int i =0;
i<table26[0].length; i++){
        if (table26[0][i] =
char_plain){
            num <- i;
        }
    }
    return num;
}
```

Dengan menggunakan kedua fungsi di atas, dapat diketahui letak cipherteks hasil enkripsi pada bujur sangkar Vigenere yaitu cipherteks adalah elemen pada bujur sangkar Vigenere pada baris ke [SearchRow(char char_key)] dan pada kolom [SearchColumnEnkripsi(char char_plain)].

Sedangkan untuk mencari plainteks pada bujur sangkar Vigenere, berikut potongan programnya :

```
public int
SearchColumnDekripsi(char[][] table,
char char_plain, int rownum){
//Fungsi ini akan menerima input
berupa huruf cipherteks dan indeks
baris kunci
//Setelah itu fungsi ini akan
mengembalikan indeks kolom plainteks
    int num <- 0;
    for (int i =1;
i<table[rownum].length ;i++){
        if (table[rownum][i] =
char_plain){
            num <- i;
        }
    }
    return num;
}
```

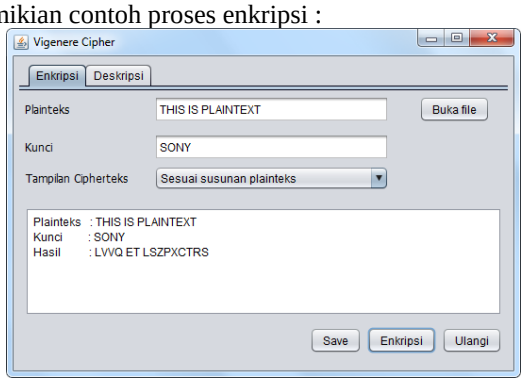
Untuk menggeser elemen-elemen pada bujur sangkar Vigenere, berikut potongan programnya :

```
public char[][] MoveTable(char[][]
source, int move){
//Fungsi ini akan menggeser elemen-
elemen pada table source sebesar
integer move
    char[][] table_copy = new
char[27][27];

    //membuat header plainteks
    int a <- 64;
    for (int j = 1; j<27;j++){
            a <- a+1;
        table_copy[0][j] <- (char)a;
    }

    //membuat header key
    a <- 64;
    for (int i = 1; i<27;i++){
        a <- a+1;
        table_copy[i][0] <- (char)a;
    }

    //menggeser elemen table bujur
sangkar vigenere
    for (int i =1;
i<table_copy.length; i++){
        a <- move;
        for (int j =1;
j<table_copy[i].length; j++){
            if (a<27){
                table_copy[i][j]<-
source[i][a];
                a++;
            }else{
                a <- 1;
                j--;
            }
        }
    }
    return table_copy;
}
```

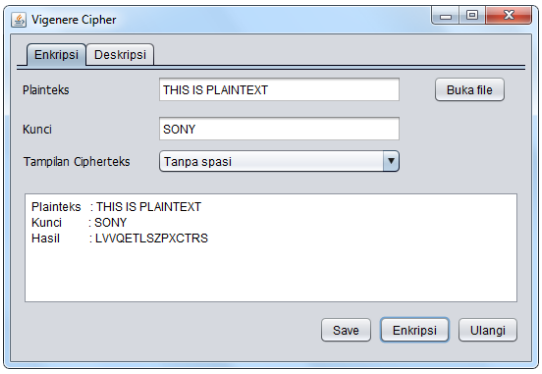
Sedangkan fungsi proses enkripsi dilakukan dengan :

1. Jika kunci lebih pendek dari plainteks, perpanjang kunci dengan cipherteks yang didapatkan dari proses enkripsi plainteks yang telah memiliki kunci. Jika kunci yang digunakan cukup untuk mengenkripsi plainteks, proses enkripsi akan dilakukan sama dengan algoritma Vigenere Cipher.
2. Untuk proses enkripsi plainteks yang tadinya tidak memiliki kunci, dilakukan dengan menggeser elemen bujur sangkar vigenere sejauh huruf kunci yang digunakan terlebih dahulu dan setelah itu lakukan enkripsi Vigenere Cipher dengan menggunakan bujur sangkar yang telah digeser elemen-elemennya

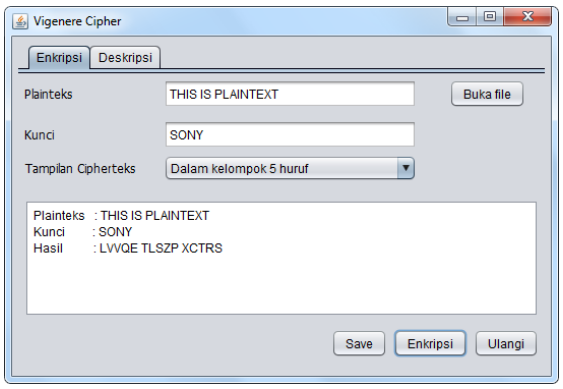
Demikian contoh proses enkripsi :



Gambar 3.1 Contoh proses enkripsi dengan format cipherteks sama dengan format plainteks



Gambar 3.2 Contoh proses enkripsi dengan format cipherteks tanpa spasi

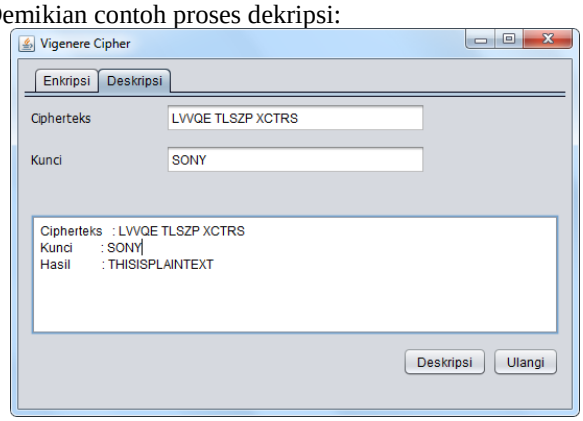


Gambar 3.3 Contoh proses enkripsi dengan format cipherteks dengan pengelompokan 5 huruf

Fungsi dekripsi dilakukan dengan :

1. Jika kunci lebih pendek dari cipherteks, perpanjang kunci dengan cipherteks. Jika kunci yang digunakan cukup untuk mendekripsi cipherteks, proses dekripsi akan dilakukan sama dengan algoritma Vigenere Cipher.
2. Untuk proses dekripsi cipherteks yang tadinya tidak memiliki kunci, dilakukan dengan menggeser elemen bujur sangkar vigenere sejauh huruf kunci yang digunakan terlebih dahulu dan setelah itu lakukan enkripsi Vigenere Cipher dengan menggunakan bujur sangkar yang telah digeser elemen-elemennya

Demikian contoh proses dekripsi:



Gambar 3.4 Contoh proses dekripsi

IV. ANALISIS

Untuk menganalisis, mari kita bandingkan contoh berikut ini :

PLAINTEKS :

It is certain that when the eruption of Vesuvius started on the morning of august AD
it caught the local population utterly unprepared
Although at the same time as we now know in retrospect
all the telltale signs were there to warn them

KUNCI: OCEANOGRAPHY

Dengan menggunakan algoritma Vigenere Cipher biasa, akan diperoleh :

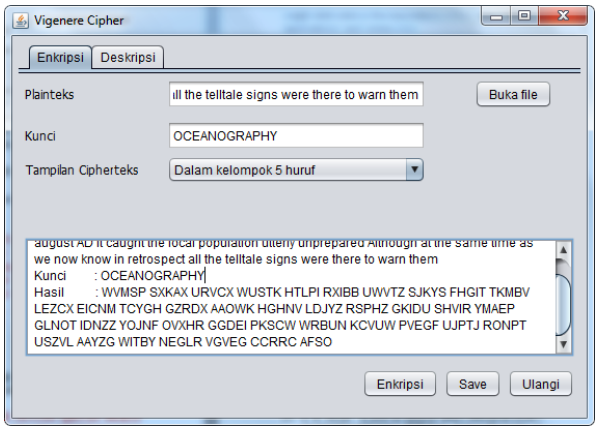
CIPHERTEKS:

WVMSP SXKAX UR**VCX WUSTK HTLPI RXIBB UWVTZ SJKYS FHGIT TKMBV LEZCX EICNM TCYGH GZRDX AAOWK HGHNV LDJYZ RSPHZ GKIDU SHVIR YMAEP GLNOT IDNZZ YOJNF OVXHR GGDEI PKSCW WRBUN KCVUW PVEGF UJPTJ RONPT USZVL AAYZG WITBY NEGLR VGVEG CCRRC AFSO**

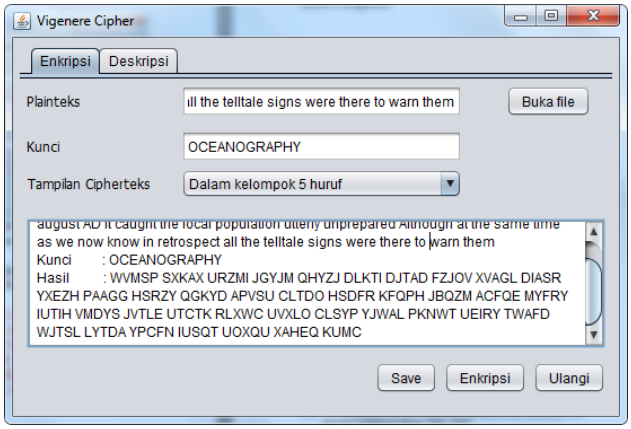
Sedangkan dengan menggunakan algoritma Vigenere Cipher yang sudah dimodifikasi, akan diperoleh :

CIPHERTEKS:

WVMSP SXKAX URZMI JGYJM QHYZJ DLKTI DJTAD FZJOV XVAGL DIASR YXEZH PAAGG HSRZY QGKYD APVSU CLTDO HSDFR KFQPH JBQZM ACFQE MYFRY IUTIH VMDYS JVTLE UTCTK RLXWC UVXLO CLSYP YJWAL PKNWT UEIRY TWAFD WJTSL LYTDA YPCFN IUSQT UOXQU XAHEQ KUMC



Gambar 4.1 Proses enkripsi dengan menggunakan algoritma Vigenere Cipher standar



Gambar 4.2 Proses enkripsi dengan menggunakan algoritma Vigenere Cipher yang telah dimodifikasi

Setelah dimodifikasi, dapat dilihat algoritma Vigenere Cipher ini menjadi lebih aman karena dengan modifikasi kunci dan penggeseran bujur sangkar Vigenere, huruf-huruf pada plainteks yang terletak pada kelipatan panjang kunci tidak akan dienkripsi menjadi cipherteks yang sama.

Keuntungan dari proses modifikasi ini adalah proses ini tidak rumit untuk diimplementasikan akan tetapi keamanan dari algoritma Vigenere Cipher ini dapat meningkat jauh. Dengan menggunakan proses modifikasi ini, metode Kasiski untuk memecahkan pesan yang dienkripsi dengan menggunakan Vigenere Cipher pun menjadi sulit untuk dilakukan karena untuk mengetahui panjang kunci yang menjadi tujuan dari metode Kasiski tidak dapat dilakukan. Selain itu metode analisis frekuensi pun menjadi tidak efektif karena dapat dilihat frekuensi kemunculan huruf cipherteks yang menggunakan algoritma Vigenere Cipher standar dan yang telah dimodifikasi sebagai berikut :

Vigenere Cipher standar	Vigenere Cipher yang telah dimodifikasi
A = 7 = IIIIIII B = 5 = IIIII C = 9 = IIIIIIIII D = 5 = IIIII E = 7 = IIIIIII F = 4 = IIII G = 13 = IIIIIIIIIIIII H = 8 = IIIIIIII I = 9 = IIIIIIIII J = 5 = IIIII K = 8 = IIIIIIII L = 6 = IIIIII M = 4 = IIII N = 8 = IIIIIIII O = 6 = IIIIII P = 8 = IIIIIIII Q = 0 = R = 11 = IIIIIIIIIII S = 10 = IIIIIIIIII T = 10 = IIIIIIIIII U = 8 = IIIIIIII V = 12 = IIIIIIIIIIII W = 8 = IIIIIIII X = 7 = IIIIIII Y = 7 = IIIIIII Z = 9 = IIIIIIIII	A = 12 = IIIIIIIIIIII B = 1 = I C = 7 = IIIIIII D = 10 = IIIIIIIIII E = 5 = IIIII F = 7 = IIIIIII G = 5 = IIIII H = 7 = IIIIIII I = 7 = IIIIIII J = 9 = IIIIIIIII K = 7 = IIIIIII L = 10 = IIIIIIIIII M = 7 = IIIIIII N = 2 = II O = 4 = IIII P = 7 = IIIIIII Q = 8 = IIIIIIII R = 7 = IIIIIII S = 10 = IIIIIIIIII T = 12 = IIIIIIIIIIII U = 10 = IIIIIIIIII V = 7 = IIIIIII W = 6 = IIIIII X = 8 = IIIIIIII Y = 13 = IIIIIIIIIIIII Z = 6 = IIIIII

V. KESIMPULAN

Algoritma Vigenere Cipher masih bisa terus ditingkatkan tingkat keamanannya. Misalnya dengan memperbaiki kekurangan algoritma ini di mana huruf plainteks yang terletak pada kelipatan panjang kunci akan dienkripsi menjadi cipherteks yang sama.

Salah satu cara untuk memperbaiki kekurangan algoritma tersebut adalah dengan tidak mengulangi kunci secara periodik apabila panjang kunci lebih pendek dari panjang plainteks. Salah satu cara memperpanjang yang dapat dilakukan ketika panjang kunci lebih pendek adalah menyambungkan kunci dengan cipherteks yang telah dihasilkan oleh plainteks yang telah memiliki kunci. Selain itu untuk lebih meningkatkan keamanan dari algoritma Vigenere Cipher, dapat dilakukan penggeseran bujur sangkar Vigenere sejauh kunci yang digunakan untuk proses enkripsi huruf-huruf plainteks yang menggunakan perpanjangan kunci.

Dengan cara demikian, pemecahan cipherteks menjadi plainteks dengan menggunakan metode Kasiski akan menjadi sulit atau bahkan sama sekali tidak dapat dilakukan. Hal ini dikarenakan tujuan dari metode Kasiski untuk memperoleh panjang kunci yang digunakan dalam proses enkripsi pun menjadi sulit untuk dilakukan karena dengan menggunakan modifikasi ini, huruf- huruf plainteks yang terletak pada kelipatan panjang kunci tidak akan dienkripsi menjadi cipherteks yang sama. Selain itu metode analisis frekuensi pun menjadi sulit untuk dilakukan karena frekuensi kemunculan huruf pada cipherteks yang dihasilkan dari proses enkripsi algoritma Vigenere Cipher yang standar dengan cipherteks yang dihasilkan dari proses enkripsi dari algoritma Vigenere Cipher yang telah dimodifikasi berbeda jauh. Dengan demikian para kriptanalis pun akan menjadi sulit untuk

memecahkan cipherteks hasil enkripsi dari algoritma Vigenere Cipher yang telah dimodifikasi ini.

V. DAFTAR REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Algoritma%20Kriptografi%20Klasik_bag1%20(2013).ppt. Diakses pada tanggal 24 Maret 2013 pukul 17.00.

[2] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Algoritma%20Kriptografi%20Klasik_bag2%20(2013).ppt. Diakses pada tanggal 24 Maret 2013 pukul 17.10.

[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Kriptanalisis%20(2013).ppt. Diakses pada tanggal 24 Maret 2013 pukul 17.20.

[4] http://maryonojk.blogspot.com/2011/02/sistem-penyandian-pesan-teks-berbasis.html. Diakses pada tanggal 24 Maret 2013 pukul 23.20.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Maret 2013

Janice Laksana
13510035