IMPLEMENTASI ALGORITMA KRIPTOGRAFI VIGENERE CHIPER UNTUK MENGAMANKAN FILE TEXT MENGGUNAKAN JAVA NETBEAN 8.0

Yuli Praptomo PHS., S.Kom., M.Cs[1]
[1]Teknik InformatikaSTMIK El Rahma Yogyakarta

Email: [1]y.praptomo@gmil.com

Abstract

The change in information and communication technology provides a major change with the increasing utilization of computer networks. The positive impact is information can be shared through computer networks in the form of digital information. At the same time this advantage is also used to perform illegal actions eg hacking bank transaction information, usernames and keywords. So it is necessary to apply security procedures to information, especially information in the form of text which is an important form of digital information.

Provision of security procedures to meet the needs of information security in the form of text can be done by applying cryptographic techniques. One of them by applying vigenere cipher as a model used for the process of encryption and decryption. In this research will be made an application that can provide a solution to solve the problem of security needs in the form of text information using vigenere cipher method.

This cryptographic application is used and runs well to encrypt and decrypt a text file or text message. The key usage method in this study is more secure than the existing vigenere cipher method, since the characters used to encrypt more files are 256 characters.

***Kata kunci** : teks, cryptographic, enkripsi, dekripsi, vigenere, cipher.*

1. PENDAHULUAN

Perubahan teknologi informasi dan komunikasi memberikan perubahan besar dengan meningkatnya pemanfaatan jaringan komputer. Dampak positifnya adalah informasi dapat dibagi melalui jaringan komputer dalam bentuk informasi digital. Pada saat yang sama keuntungan ini juga digunakan untuk melakukan tindakan ilegal misal peretasan informasi transaksi bank, username dan kata kunci. Sehingga perlu diterapkan prosedur keamanan pada informasi khususnya informasi berupa teks yang merupakan bentuk penting dari informasi digital.

Keamanan informasi didapatkan salah satunya dengan menerapkan teknik kriptografi pada informasi. Kriptografi adalah ilmu dan seni mengubah pesan atau informasi untuk membuatnya aman dan kebal dari serangan (Forouzan, 2007). Terdapat dua faktor utama dalam teknik kriptografi yaitu enkripsi dan dekripsi. Enkripsi atau penyandian merupakan proses pengubahan informasi agar tidak terbaca. Hasil dari enkripsi berupa informasi yang disandikan atau ciphertext. Ciphertext dapat diambil informasinya dengan cara membalik sandi tersebut menggunakan algoritma kriptografi yang sama. Proses pembalikan sandi sehingga didapatkan informasi yang nyata ini biasa disebut proses dekripsi.

Pada umumnya kriptografi dibedakan menjadi dua jenis yaitu kriptografi simetris (*symmetric key cryptography*) dan kriptografi kunci tidak simetris (*asymmetric key cryptography*). Kriptografi kunci simetris merupakan algoritma kriptografi yang menggunakan kunci yang sama dalam proses enkripsi dan dekripsi. Berbeda dengan kriptografi kunci tidak simetris yaitu algoritma yang menggunakan dua kunci yang berbeda yaitu private key dan public key dalam proses enkripsi dan dekripsi. *Public key* adalah kunci yang digunakan untuk proses enkripsi. Sedangkan *private key* adalah kunci yang digunakan untuk mendekripsi informasi yang

disandikan dengan tujuan mendapatkan informasi. Oleh karena itu *private key* sebaiknya hanya diketahui oleh pendekripsi informasi yang disandikan (Forouzan, 2007).

Caesar cipher dan *vigenere cipher* merupakan contoh metode kriptografi kunci simetris dengan model penggantian karakter atau subtitusi (*substitution*). Metode caesar cipher menggunakan kunci berupa angka sebagai nilai untuk mengganti karakter pesan dengan karakter yang lain. Hal yang berbeda pada vigenere cipher karena menggunakan abjad sebagai kunci penyandian untuk melakukan penggantian atau subtitusi karakter pesan.Contoh metode kriptografi kunci simetris yang lain adalah columnar transposition cipher. Metode tersebut menyandikan pesan dengan cara mengubah susunan karakter pada pesan atau transposisi karakter (*transposition*) (Tanenbaum, 2011).

Pemberian prosedur keamanan untuk memenuhi kebutuhan kemanan informasi berupa teks dapat dilakukan dengan menerapkan teknik kriptografi. Salah satunya dengan cara menerapkan *vigenere cipher* sebagai model yang digunakan untuk proses enkripsi dan dekripsi.

2. KRIPTOGRAFI

Kriptografi (*cryptography*) merupakan ilmu dan seni untuk menjaga pesan agar aman. (*Cryptography is the art and science of keeping messages secure).* "*Crypto*" berarti "*secret*" (rahasia) dan "*graphy*" berarti "*writing*" (tulisan). Para pelaku atau praktisi kriptografi disebut cryptographers. Sebuah algoritma kriptografik (*cryptographic algorithm*), disebut cipher, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat.

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah enkripsi (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah "*encipher*".

Proses sebaliknya, untuk mengubah ciphertext menjadi plaintext, disebut dekripsi (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah "*decipher*".

Cryptanalysis adalah seni dan ilmu untuk memecahkan *ciphertext* tanpa bantuan kunci. *Cryptanalyst* adalah pelaku atau praktisi yang menjalankan *cryptanalysis*. *Cryptology* merupakan gabungan dari *cryptography* dan *cryptanalysis*.

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi data anda disandikan (*encrypted*) dengan menggunakan sebuah kunci (*key*). Untuk membuka (*decrypt*) data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi (untuk kasus *private keycryptography*) atau dengan kunci yang berbeda (untuk kasus *public key cryptography*).

Algoritma dari enkripsi adalah fungsi-fungsi yang digunakan untuk melakukan fungsi enkripsi dan dekripsi. Algoritma yang digunakan menentukan kekuatan dari enkripsi, dan ini biasanya dibuktikan dengan basis matematika.

Berdasarkan cara memproses teks (*plaintext*), cipher dapat dikategorikan menjadi dua jenis: *block cipher* and *stream cipher*. Block cipher bekerja dengan memproses data secara blok, dimana beberapa karakter / data digabungkan menjadi satu blok. Setiap proses satu blok menghasilkan keluaran satu blok juga. Sementara itu stream cipher bekerja memproses masukan (karakter atau data) secara terus menerus dan menghasilkan data pada saat yang bersamaan.

Kekuatan dari penyandian bergantung kepada kunci yang digunakan. Beberapa algoritma enkripsi memiliki kelemahan pada kunci yang digunakan. Untuk itu, kunci yang

lemah tersebut tidak boleh digunakan. Selain itu, panjangnya kunci, yang biasanya dalam ukuran bit, juga menentukan kekuatan dari enkripsi. Kunci yang lebih panjang biasanya lebih aman dari kunci yang pendek. Jadi enkripsi dengan menggunakan kunci 128-bit lebih sukar dipecahkan dengan algoritma enkripsi yang sama tetapi dengan kunci 56-bit. Semakin panjang sebuah kunci, semakin besar *keyspace* yang harus dijalani untuk mencari kunci dengan cara *brute force attack* atau coba-coba karena keyspace yang harus dilihat merupakan pangkat dari bilangan 2. Jadi kunci 128-bit memiliki keyspace 2128, sedangkan kunci 56-bit memiliki keyspace 256. Artinya semakin lama kunci baru bisa ketahuan.

Plaintext adalah pesan atau informasi yang akan dikirimkan dalam format yang mudah dibaca atau dalam bentuk aslinya. Ciphertext adalah informasi yang sudah dienkripsi.

Kembali ke masalah algoritma, keamanan sebuah algoritma yang digunakan dalam enkripsi atau dekripsi bergantung kepada beberapa aspek. Salah satu aspek yang cukup penting adalah sifat algoritma yang digunakan. Apabila kekuatan dari sebuah algoritma sangat tergantung kepada pengetahuan (tahu atau tidaknya) orang terhadap algoritma yang digunakan, maka algoritma tersebut disebut "*restricted algorithm*". Apabila algoritma tersebut bocor atau ketahuan oleh orang banyak, maka pesan-pesan dapat terbaca. Tentunya hal ini masih bergantung kepada adanya kriptografer yang baik. Jika tidak ada yang tahu, maka sistem tersebut dapat dianggap aman (meskipun semu).

Meskipun kurang aman, metoda pengamanan dengan *restricted algorithm* ini cukup banyak digunakan karena mudah implementasinya dan tidak perlu diuji secara mendalam. Contoh penggunaan metoda ini adalah enkripsi yang menggantikan huruf yang digunakan untuk mengirim pesan dengan huruf lain. Ini disebut dengan "*substitution cipher*".

Pembakuan penulisan pada kriptografi dapat ditulis dalam bahasa matematika. Fungsi-fungsi yang mendasar dalam kriptografi adalah enkripsi dan dekripsi. Enkripsi adalah proses mengubah suatu pesan asli (*plaintext*) menjadi suatu pesan dalam bahasa sandi (*ciphertext*).

C = E (M), *(1)*

Dimana : M = pesan asli
E = proses enkripsi
C = pesan dalam bahasa sandi (untuk ringkasnya disebut sandi)

Sedangkan dekripsi adalah proses mengubah pesan dalam suatu bahasa sandi menjadi pesan asli kembali.

M = D (C) *(2)*

Dimana : M = pesan asli
D = proses dekripsi
C = pesan dalam bahasa sandi (untuk ringkasnya disebut sandi)

Umumnya, selain menggunakan fungsi tertentu dalam melakukan enkripsi dan dekripsi, seringkali fungsi itu diberi parameter tambahan yang disebut dengan istilah kunci.

3. TUJUAN KRIPTOGRAFI

Dalam teknologi informasi, telah dan sedang dikembangkan cara untuk menangkal berbagai bentuk serangan semacam penyadapan dan pengubahan data yang dikirimkan. Salah satu cara yang ditempuh mengatasi masalah ini ialah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak yang tidak berhak mengakses.

Transformasi ini memberikan solusi pada dua macam masalah keamanan data, yaitu masalah privasi (*privacy*) dan keotentikan (*authenticatioan*). *Privaci* mengandung arti bahwa data

yang dikirimkan hanya dapat dimengerti informasinya oleh penerima yang sah atau berhak. Sedangkan keotentikan mencegah pihak ketiga untuk mengirimkan data yang salah atau mengubah data yang dikirimkan.

Kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada 4 syarat yang perlu dipenuhi, yaitu.

a. **Kerahasiaan**. Pesan (*plaintext*) hanya dapat dibaca oleh pihak yang memliki kewenangan.
b. ***Autentikasi***. Pengirim pesan harus dapat diidentifikasi dengan pasti, penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.
c. **Integritas**. Penerima pesan harus dapat memastikan bahwa pesan yang dia terima tidak dimodifikasi saat dalam proses transmisi data.
d. ***Non-Repudiation***. Pengirim pesan harus tidak bisa menyangkal pesan yang dia kirimkan.

4. ALGORITMA KRIPTOGRAFI

Cryptographic system atau *cryptosystem* adalah suatu fasilitas untuk mengkonversikan plaintext ke ciphertext dan sebaliknya. Dalam sistem ini, seperangkat parameter yang menentukan transformasi pencipheran tertentu disebut suatu set kunci. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci-kunci yang digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung pada sistem yang digunakan. Kriptosistem terdiri dari sebuah algoritma, plaintext, chipertext dan kunci-kunci. Secara umum, kriptosistem digolongkan menjadi 2 buah, yaitu.

1. Kriptosistem simetri
2. Kriptosistem asimetri

Dalam kriptosystem simetri, algoritma yang digunakan hanya membutuhkan satu kunci untuk proses enkripsi dan dekripsi. Sedang dalam kriptografi asimetri, algoritma yang digunakan membutuhkan dua kunci yang berbeda dalam proses enkripsi dan dekripsi pesan. Kriptosistem dengan menggunakan *symmetric cryptography* kadang disebut sebagai *Secret-key cryptography* merupakan bentuk kryptografi yang lebih tradisional, dimana sebuah kunci tunggal dapat digunakan untuk mengenkrip dan mendekrip pesan. *Secret-key cryptography* tidak hanya berkaitan dengan enkripsi tetapi juga berkaitan dengan otentikasi, disebut juga *message authentication codes*.

Masalah utama yang dihadapi *secret-key cryptography* adalah membuat pengirim dan penerima menyetujui kunci rahasia tanpa ada orang lain yang mengetahuinya. Ini membutuhkan metode dimana dua pihak dapat berkomunikasi tanpa takut akan disadap. Kelebihan *secret-key cryptography* dari *public-key cryptography* adalah lebih cepat. Teknik yang paling umum dalam *secret-key cryptography* adalah *block ciphers, stream ciphers*, dan *message authentication codes*. Berdasarkan jenis kunci yang digunakannya, algoritma kriptografi dikelompokan menjadi dua bagian, yaitu.

a. Symmetric Algorithm

Symmetric algorithm atau disebut juga *secret key algorithm* adalah algoritma yang kunci enkripsinya dapat dihitung dari kunci dekripsi dan begitu pula sebaliknya, kunci dekripsi dapat dihitung dari kunci enkripsi.

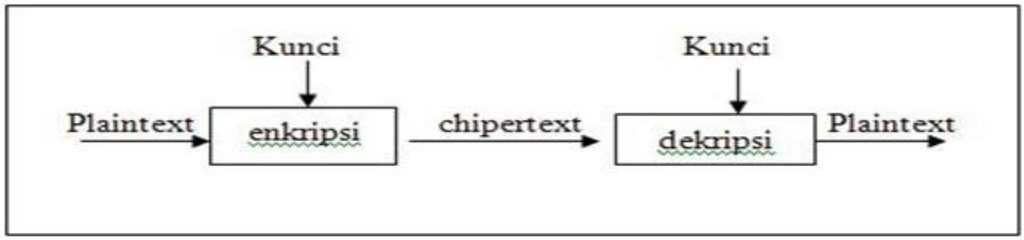
Pada sebagian besar *symmetric algorithm* kunci enkripsi dan kunci dekripsi adalah sama. Symmetric algorithm memerlukan kesepakatan antara pengirim dan penerima pesan pada suatu kunci sebelum dapat berkomunikasi secara aman. Keamanan *symmetric algorithm* tergantung pada rahasia kunci. Pemecahan kunci berarti memungkinkan setiap orang dapat mengenkripsi dan mendekripsi pesan dengan mudah. *Symmetric algorithm* dapat dikelompokkan menjadi dua jenis, yaitu *stream cipher* dan *block cipher*. Stream cipher beroperasi bit per bit (atau

byte per byte) pada satu waktu. Sedangkan block cipher beroperasi per kelompok-kelompok bit yang disebut blok (block) pada satu waktu.

b. Asymmetric Algorithm

Asymmetric algorithm atau disebut juga *public key algorithm* didesain agar memudahkan dalam distribusi kunci yang digunakan untuk enkripsi dan dekripsi. Kunci dekripsi pada *public key algorithm* secara praktis tidak dapat dihitung dari kunci enkripsi. Algoritma ini disebut "*public key*" karena kunci dapat dibuat menjadi publik. Setiap orang dapat menggunakan kunci enkripsi untuk mengenkripsi pesan, tetapi hanya orang yang memiliki kunci dekripsi yang dapat mendekripsi pesan tersebut. Pada sistem ini kunci enkripsi sering disebut kunci publik (*public key*), dan kunci dekripsi disebut kunci rahasia (*private key*).

Metode kriptosystem simetri disebut juga dengan enkripsi simetri atau enkripsi konvensional. Gambar di bawah ini mengilustrasikan kinerja dari proses enkripsi konvensional.



Gambar 1 : Ilutrasi Kinerja Proses Enkripsi Konvensional

Proses enkripsi terdiri dari sebuah algoritma dan sebuah kunci. Kunci adalah sebuah nilai yang terlepas dari pesan asli (plaintext) dan mengontrol algoritma yang dipakai. Penerapan algoritma akan menghasilkan output yang berbeda sesuai dengan kunci yang digunakan. Mengubah kunci berarti mengubah output dari algoritma yang dipakai.

Setelah chipertext dihasilkan, chipertext tersebut dapat diubah kembali menjad pesan asli dengan algoritma dekripsi dan dengan kunci yang sama seperti yang digunakan pada saat enkripsi.

Keamanan dari enkripsi konvensional ini terdiri dari beberapa faktor. Pertama, algoritma enkripsi harus benar-benar teruji, sehingga tidak dimungkinkan untuk mendekripsi sebuah pesan hanya dalam bentuk *ciphertext*. Kedua, keamanan enkripsi konvensional juga ditentukan oleh kerahasiaan kunci yang digunakan, bukan kerahasiaan algoritma yang digunakan. Jadi, kita juga harus yakin bahwa dekripsi tidak dimungkinkan hanya dengan mengetaui ciphertext dan algoritma yang digunakan tanpa mengetahui kunci yang digunakan.

Bentuk umum enkripsi konvensional dapat dilihat dari contoh berikut : user A akan mengenkripsi plaintext X = [x1, x2, .. ,xm] , (m elemen dari X merupakan huruf dari alphabet pesan) dengan kunci K = [K1, k2, .. , kj], dengan pesan X dan kunci K tersebut akan dihasilkan cipertext Y = [Y1, Y2, .. , yn], maka kita dapat menuliskan rumus :

$$Y=Ek(X) \quad (3)$$

Selanjutnya, chipertext tersebut dikirimkan ke user B. User B akan mendekripsi chipertext tersebut agar menjadi pesan asli dengan algoritma dekripsi dan kunci yang sama seperti yang digunakan pada saat enkripsi. Hal ini dapat dirumuskan sebagai berikut:

$$X=Dk(Y) \quad (4)$$

5. ALGORITMA VIGENERE CIPHER

Algoritma enkripsi jenis ini sangat dikenal karena mudah dipahami dan diimplementasikan. Teknik untuk menghasilkan *ciphertext* bisa dilakukan menggunakan substitusi angka maupun bujur sangkar *vigenere*. Teknik susbtitusi *vigenere* dengan menggunakan angka dilakukan dengan menukarkan huruf dengan angka, hampir sama dengan kode geser. Contoh:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 2 Contoh Tabel Substitusi Algoritma Kriptografi *Vigenere Cipher*

Plaintext: **PLAINTEXT**
Kunci: **CIPHER**

Plain	15	11	0	8	13	19	4	23	19
Kunci	2	8	15	7	4	17	2	8	15
Hasil	17	19	15	15	17	10	6	5	8
Ciphertext	R	T	P	P	R	K	G	F	I

Gambar 3 Contoh Tabel Kriptografi dengan Algoritma Vigenere Cipher

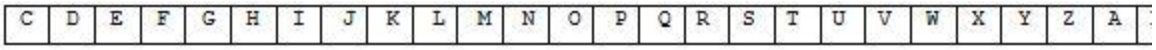
Dengan metode pertukaran angka dengan huruf di atas, diperoleh bahwa teks asli (*PLAINTEXT*) memiliki kode angka (15,11, 0, 8, 13, 19, 4, 23, 19), sedangkan kode angka untuk teks kunci (*CIPHER*) yaitu (2, 8, 15, 7, 4, 17). Setelah dilakukan perhitungan, maka dihasilkan kode angka *ciphertext* (17, 19, 15, 15, 17, 10, 6, 5, 8). Jika diterjemahkan kembali menjadi huruf sesuai urutan awal, maka menjadi huruf **RTPPRKGFI**.

Sedangkan metode lain untuk melakukan proses enkripsi dengan metode vigenere cipher yaitu menggunakan tabularecta (disebut juga bujur sangkar *vigenere*).

	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

Gambar 4 Contoh Tabula Recta Algoritma Kriptografi Vigenere Cipher

Kolom paling kiri dari bujur sangkar menyatakan huruf-huruf kunci, sedangkan baris paling atas menyatakan huruf-huruf plaintext. Setiap baris di dalam bujur sangkar menyatakan huruf-huruf ciphertert yang diperoleh dengan *Vigenere cipher*, yang mana jumlah pergeseran huruf plaintext ditentukan nilai numerik huruf kunci tersebut (yaitu, a=0, b=1, c=2, …, z=25). Sebagai contoh, huruf kunci c (=2) menyatakan huruf-huruf plaintext digeser sejauh 2 huruf ke kanan (dari susunan alfabetnya), sehingga huruf-huruf *ciphertext* pada baris c adalah.



Gambar 5 Potongan Tabula Recta Baris ke-C

Bujur sangkar *vigenere* digunakan untuk memperoleh *ciphertert* dengan menggunakan kunci yang sudah ditentukan. Jika panjang kunci lebih pendek daripada panjang *plaintext*, maka kunci diulang penggunaannya (sistem periodik). Bila panjang kunci adalah m, maka periodenya dikatakan m. Sebagai contoh, jika plaintext adalah *THIS PLAINTEXT* dan kunci adalah sony, maka penggunaan kunci secara periodik sebagai berikut.

Plaintext : THIS PLAINTEXT
Kunci : sony sonysonys

Untuk mendapatkan ciphertext dari teks dan kunci di atas, untuk huruf plaintext pertama T, ditarik garis vertikal dari huruf T dan ditarik garis mendatar dari huruf s, perpotongannya adalah pada kotak yang berisi huruf L.

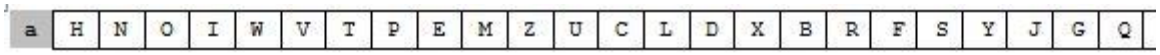
Dengan cara yang sama, ditarik garis vertikal dari huruf Hdan ditarik garis mendatar pada huruf o, perpotongannya adalah pada kotak yang juga berisi berisi huruf V. hasil enkripsi seluruhnya adalah sebagai berikut.

Plaintext : THIS PLAINTEXT
Kunci : sony sonysonys
Ciphertext : LVVQ HZNGFHRVL

Variasi-variasi *vigenere cipher* pada dasarnya perbedaannya terletak pada cara membentuk tabel atau cara menghasilkan kuncinya, sedangkan enkripsi dan dekripsi tidak berbeda dengan *vigenere cipher* standar. Beberapa variasi tersebut sebagai berikut.

a. *Full Vigenere Cipher*

Pada varian ini, setiap baris di dalam tabel tidak menyatakan pergeseran huruf, tetapi merupakan permutasi huruf-huruf alfabet. Misalnya, pada baris a susunan huruf-huruf alfabet adalah acak seperti di bawah ini:



Gambar 6 Contoh Potongan Tabula Recta Full Vigenere Cipher

b. *Auto-Key Vigenere cipher*

Idealnya kunci tidak digunakan secara berulang. Pada auto-keyvigenere cipher, jika panjang kunci lebih kecil dari panjang plaintext, maka kunci disambung dengan plaintext tersebut. Misalnya, untuk mengenkripsi pesan NEGARA PENGHASIL MINYAK dengan kunci INDO, maka kunci tersebut disambung dengan plaintext semula sehingga panjang kunci menjadi sama dengan panjang plaintext:

Plaintext: NEGARA PENGHASIL MINYAK
Kunci: INDONE GARAPENGH ASILMI

c. *Running-Key Vigenere cipher*

Pada varian ini, kunci bukan string pendek yang diulang secara periodik seperti pada vigenere cipher standar, tetapi kunci adalah string yang sangat panjang yang diambil dari teks bermakna (misalnya naskah proklamasi, naskah Pembukaan UUD 1945, terjemahan ayat di dalam kitab suci, dan lain-lain).

Misalnya untuk mengenkripsi plaintext NEGARA PENGHASIL MINYAK dapat menggunakan kunci berupa sila ke-2 Pancasila : KEMANUSIAAN YANG ADIL DAN BERADAB. Selanjutnya enkripsi dan dekripsi dilakukan seperti biasa. (Munir, 2006)

6. Permasalahan

Masalah yang akan dibahas pada makalah ini adalah :

a. Mekanisme kerja enkripsi dan dekripsi data dengan menggunakan algoritma Vigenere Chiper.
b. Mengimplementasikan algoritma Vigenere Chiper dalam enkripsi dan dekripsi data.
c. Menganalisis dan membandingkan algoritma Caesar Chiper dalam enkripsi dan dekripsi data.

7. Analisa dan Pembahasan

Dalam penelitin ini agar permasalahan tidak melebar, maka perlu adanya pembatasan permasalahan yaitu.

1) Pembahasan hanya untuk melakukan proses enkripsi pada file text berupa huruf abjad dari huruf A sampai dengan huruf Z.
2) Algotirma yang digunakan adalah algoritma vigenere chiper standart, yaitu algoritma vigenere chiper dengan menggunakan kunci periodik.
3) Huruf yang digunkan adalah huruf kecil saja.
4) Penulisan pesan tidak menggunakan spasi.

a. MembacaPesan.

Proses pembacaan data awal dilakukan dengan mengkonversi input yang berupa deretan huruf menjadi deretan angka. Proses ini dilakukan dengan menggunakan teknik array dengan membuat indeks sebanyak deretan text yang menjadi inputnya.

Konversi dimulai dari angka 0 untuk huruf a, 1 untuk huruf b, dan seterusnya (yaitu, a=0, b=1, c=2, …, z=25), sepeti terlihat pada tabel 1.

Tabel 1. Konversi data huruf ke data angka

Huruf	**A**	**B**	**C**	**D**	**R**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Angka	0	1	2	3	4	5	6	7	8	9	10	11	12
Huruf	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Angka	13	14	15	16	17	18	19	20	21	22	23	24	25

Proses pembacaan tersebut dilakukan dengan perulangan untuk membaca pesan sesuai dengan panjang pesan, kemudian dibandingkan dengan data huruf dari a sampai dengan z (dideklarasikan data huruf = “abcdefghijklmnopqrstuvwxyz”). Jika data pesan bernilai sama dengan data huruf, maka disimpan dalam data pesan yang bertype integer bersama indeks array data tersebut. Proses pembacaan terlihat pada program gambar 7.

```
for (int i = 0; i < pesan.length(); i++)
{
    for (int j = 0; j < dataHuruf.length(); j++)
    {

        char p = pesan.charAt(i);
        char dh = dataHuruf.charAt(j);

        if (p == dh) {
            dataPesanInt = dataHuruf.indexOf(dh);
            listPesan.add(dataPesanInt);
        }
    }
}
return listPesan;
```

Gambar 7. Proses pembacaan data pesan.

b. Membaca Kunci.

Proses pembacaan kunci dilakukan dengan cara yang sama dengan pembacaan input plaintext, yaitu dengan cara mengkonversi input yang berupa deretan huruf menjadi deretan angka. Proses ini dilakukan dengan menggunakan teknik array dengan membuat indeks sebanyak deretan text yang menjadi inputnya. Seperti terlihat pada tabel 1.

Pada gambar 8, secara garis besar terlihat bahwa proses pembacaan kunci sama dengan proses pembacaan pesan. Perbedaannya adalah untuk panjang kunci harus disamakan dengan panjang pesan, dengan menambah abjad dari pesan yang ada sebanyak selisih antara panjang kunci dengan panjang pesan,

```
int jmlKunci = 0;
while (kunci.length() < pesan.length())
{
    kunci += kunci.charAt(jmlKunci);
    jmlKunci++;
}
for (int i = 0; i < kunci.length(); i++)
{
    for (int j = 0; j < dataHuruf.length(); j++)
    {
        char k = kunci.charAt(i);
        char dh = dataHuruf.charAt(j);

        if (k == dh)
        {
            dataKunciInt = dataHuruf.indexOf(dh);
            listKunci.add(dataKunciInt);
        }
    }
}
return listKunci;
```

Gambar 8. Proses pembacaan kunci.

c. Proses Enkripsi.

Proses enkripsi dilakukan dengan menambah pesan dengan kunci, jika jumlah kedua angka itu belum melebihi panjang data huruf, maka jumlah keduanya merupakan hasil dari enkripsi. Tetapi jika melebihi panjang data huruf, maka jumlah pesan ditambah kunci hasilnya dikurangi dengan panjang data huruf seperti terlihat pada gambar 9.

```
        int c = pesan + kunci;

        if (c < dataHuruf.length())
        {
            chiper = c;
        }
        else
        {
            chiper = c - dataHuruf.length();
        }

        return chiper;
    }
```

Gambar 9. Prosedur enkripsi

d. Proses Dekripsi.

Proses dekripsi dilakukan dengan mengurangi pesan dengan kunci, jika jumlah kedua angka itu belum melebihi panjang data huruf, maka jumlah keduanya merupakan hasil dari dekripsi. Tetapi jika melebihi panjang data huruf, maka jumlah pesan ditambah kunci hasilnya menambah dengan panjang data huruf seperti terlihat pada gambar 9.

```
        int p = chiper - kunci;
        if (p >= 0)
        {
            pesan = p;
        }
        else
        {
            pesan = p + dataHuruf.length();
        }
        return pesan;
    }
```

Gambar 10. Prosedur dekripsi

e. Program Utama

Program utama akan pada saat akan melakukan proses enkripsi akan memanggil prosedur untuk membaca pesan dan porsedur membaca kunci (gambar 7 dan gambar 8). Selanjutnya akan melakukan proses enkripsi dengan prosedur enkripsi (gambar 9). Setelah itu untuk menampilkan hasil akan dipanggil prosedur hasil (gambar 12), kemudian akan ditampilkan hasilnya seperti terlihat pada gambar 13.

```
System.out.println("Enkripsi");
EnkripDekrip ed = new EnkripDekrip();
for (int i = 0; i < pesan.length(); i++)
{
    int iP = Integer.parseInt(sP.get(i).toString());
    int iK = Integer.parseInt(sK.get(i).toString());
    int gE = ed.getEnkrip(iP, iK, dataHuruf);

    en.add(gE);

}
System.out.println("Pesan : " + pesan);
System.out.println("Kunci : " + kunci);
String hEN = ed.getHasil(en, dataHuruf);
System.out.println("Hasli Enkripsi :" + hEN);
chiper = hEN;
```

Gambar 11. Program pemanggilan prosedur enkripsi

Proses dekripsi pada dasarnya tidak terlalu berbeda dengan proses enkripsi, yaitu dengan memanggil prosedur untuk membaca pesan dan porsedur membaca kunci (gambar 7 dan gambar 8). Selanjutnya akan melakukan proses dekripsi dengan prosedur enkripsi (gambar 10). Setelah itu untuk menampilkan hasil akan dipanggil prosedur hasil (gambar 12), kemudian akan ditampilkan hasilnya seperti terlihat pada gambar 14.

```
System.out.println("\nDekripsi");
EnkripDekrip ed2 = new EnkripDekrip();
for (int i = 0; i < pesan.length(); i++)
{
    int iC = Integer.parseInt(sC.get(i).toString());
    int iK2 = Integer.parseInt(sK2.get(i).toString());
    int gDE = ed2.getDekrip(iC, iK2, dataHuruf);

    de.add(gDE);
}
System.out.println("Chiper : " + chiper);
System.out.println("Kunci : " + kunci);
String hDE = ed2.getHasil(de, dataHuruf);
System.out.println("Hasli Dekripsi :" + hDE);
```

Gambar 11. Program pemanggilan prosedur dekripsi

f. Hasil

Prosedure mengampil hasil dari proses endkripsi dan dekripsi, hasil terlihat pada gambar 12.

```
public String getHasil(ArrayList intEnkrip, String dataHuruf)
{
    for (int i = 0; i < intEnkrip.size(); i++)
    {
        int a = Integer.parseInt(intEnkrip.get(i).toString());
        char z = dataHuruf.charAt(a);
        hasil += z;
    }
```

Gambar 12. Prosedure tampil hasil

Hasil dari proses enkripsi dapat dilihat pada gambar 13.

```
Enkripsi
Pesan : programku
Kunci : coba
Hasli Enkripsi :rfpgtonkw
```

Gambar 13. Hasil enkripsi

Hasil dari proses dekripseperti terlihat pada gambar 14.

```
Dekripsi
Chiper : rfpgtonkw
Kunci : coba
Hasli Dekripsi :programku
```

Gambar 14. Hasil dekripsi

8. Kesimpulan dan Saran

a. Kesimpulan

Hasil penelitian menunjukkan bahwa yang pertama, implementasi vigenere chiper dapat dilakukan dengan mudah (khususnya untuk vigenere chiper standar, dengan mengubah pesan dan kunci ke dalam angka sebelum dilakukan proses enkripsi dan dekripsi. Proses enkripsi dan dekripsi dalam implementasi tidak berbeda secara signifikan. Proses endkripsi dilakukan dengan menambah pesan dengan kunci, sebaliknya untuk proses dekripsi dilakukan dengan mengurangi ciperteks dengan kunci.

b. Saran

Implementasi selanjutnya diharapkan dapat mengakomodasi pesan yang menggunakan abjad berupa huruf kecil dan huruf besar (kapital), serta dapat digunakan untuk pesan dengan beberapa kata. Dari segi tampilan hasil alangkah baiknya digunakan fasilitas GUI yang sudah disediakan oleh netbean versi 8.0.

DAFTAR PUSTAKA

[1] Dony Darius, 2006, *Kriptografi Keamanan Data dan Komunikasi*, Graha Ilmu, Yogyakarta.

[2] Dony Darius, 2008, Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi, Andi Offset, Yogyakarta.

[3] Forouzan, B.A. and S.C. Fegan. 2007.*Data Communication and Networking*. 4th ed. McGraw-Hill Companies, Inc. New York.

[4] Katz, J. and Y. Lindell. 2015. *Introduction to Modern Cryptography*. 2nd ed. CRC Press. Boca Raton.

[5] Kester, Q.A. 2012. A *Cryptosystem Based on Vigenere Cipher with Varying Key*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol 1(10): 108-113.
[6] Kromodimoeljo, S. 2009. Teori dan Aplikasi Kriptografi. SPK IT Consulting. Martin, Keith M. 2012. Everyday Cryptography: Fundamental Principles & Apllications. Oxford University Press Inc. New York.
[7] Pramanik, M.B. 2014. Implementation of Cryptography Technique Using Columnar Transposition. International Journal of Computer Applications: 19-23.
[8] Munir, R, 2004, Algoritma Kriptografi Klasik, Departemen Teknik Informatika Institut Teknologi Bandung, Bandung.
[9] Munir, R. 2006, Kriptografi, Informatika, Bandung.
[10] Sadikin, R, 2012, Kriptografi untuk Keamanan Jaringan, Andi Offset, Yogyakarta.
[11] Sinha, N. and K. Bhamidipati. 2014. Improving Security of Vigenere Cipher by Double Columnar Transposition. International Journal of Computer Applications Vol. 100(14): 6-10.
[12] Stallings, W. 2011. Cryptography and Network Security: Principles and Practice. 5th ed. Pearson Education Inc. New York.
[13] Tanenbaum, A.S. and D.J. Wetherall. 2011. Computer Networks. 5th ed. Pearson Education Inc. Boston.

Biodata Penulis

Yuli Praptomo PHS, S.Kom., M.Cs. adalah dosen tetap STMIK El Rahma Yogyakarta, lahir di Kulon Progo, 7 Juli 1972. Memperoleh gelar Sarjana Komputer, jurusan Teknik Informatika STMIK AKAKOM Yogyakarta pada tahun 1999, Memperoleh gelar Magister Komputer, jurusan IlmuKomputerUGM Yogyakarta pada tahun 2015,jabatan akademik terakhir Lektor, Jurusan Teknik Informatika.