Pengamanan Data Informasi menggunakan Kriptografi Klasik

Jati Sasongko
Fakultas Teknologi Informasi, Universitas Stikubank Semarang
email : jati@unisbank.ac.id

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

Kata kunci : kriptografi, enkripsi, dekripsi

PENDAHULUAN

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".

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 sama dengan kunci untuk mengenkripsi (untuk kasus private key cryptography) 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.

Kunci yang digunakan dan panjangnya kunci. 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. Plaintext adalah pesan atau informasi yang akan dikirimkan dalam format yang mudah dibaca atau dalam bentuk aslinya.

Ciphertext. 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".

Algoritma kriptografi klasik :
1. *Substitution Ciphers*
2. *Transposition Ciphers*

Substitution Ciphers

a. Monoalphabetic Subtitution Cipher

Satu huruf di plainteks diganti dengan satu huruf yang bersesuaian. Jumlah kemungkinan susunan huruf-huruf cipherteks yang dapat dibuat adalah sebanyak 26! = 403.291.461.126.605.635.584.000.000

Tidak dapat menyembunyikan hubungan antara plainteks dengan cipherteks. Huruf yang sama dienkripsi menjadi huruf cipherteks yang sama. Huruf yang sering muncul di dalam palinteks, sering muncul pula di dalam cipherteksnya.

Tabel substitusi dapat dibentuk secara acak:

```
Plainteks : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipherteks: DIQMTBZSYKVOFERJAUWPXHLCNG
```

Atau dengan kalimat yang mudah diingat:
Contoh:

```
belajar kriptografi
```

Buang duplikasi huruf:

```
belajrkiptogf
```

Sambung dengan huruf lain yang belum ada:

```
belajrkiptogfcdhmnqsuvwxyz
```

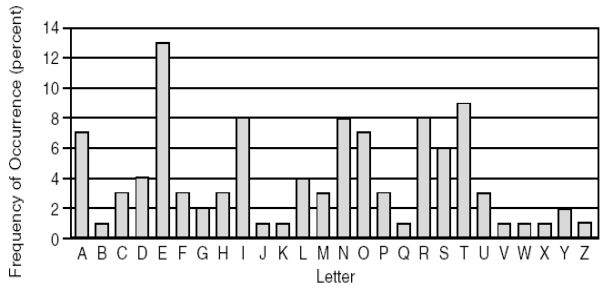
Tabel substitusi:

```
Plainteks  :ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipherteks :BELAJRKIPTOGFCDHMNQSUVWXYZ
```

b. Homophonic Substitution Cipher

Setiap huruf plainteks dipetakan ke dalam salah satu huruf cipherteks yang mungkin. Tujuan : menyembunyikan hubungan statistik antara plainteks dengan cipherteks. Fungsi *ciphering* memetakan satu-ke-banyak (*one-to-many*).

Misal huruf E → `AB, TQ, YT,UX`
huruf B → `EK, MF, KY`



Gambar 1. Frekuensi kemunculan huruf

Contoh :
Sebuah teks dengan frekuensi kemunculan huruf (gambar 1)

Huruf E muncul 13 % → dikodekan dengan 13 huruf homofon (gambar 2).

Unit cipherteks ditentukan secara acak dipilih diantara semua homofon.

Contoh :
Plainteks : KRIPTO
Cipherteks : DI CE AX AZ CC DX

Enkripsi : satu-ke-banyak
Dekripsi : satu-ke-satu
Dekripsi menggunakan tabel homofon yang sama.

c. Chipher abjad-majemuk (polyalphabetic subtitution chipher)

Huruf Plainteks	Pilihan huruf Cipherteks
A	BU CP AV AH BT BS CQ
B	AT
C	DL BK AU
D	BV DY DM AI
E	DK CO AW BL AA CR BM CS AF AG BO BN BE
F	BW CM CN
G	DN BJ
H	AS CL CK
I	DJ BI AX CJ AB BP CU CT
J	BX
K	DI
L	AR BH CI AJ
M	DH BG AY
N	BY DG DF CH AC BR DU DT
O	DZ BF DX AK CG BQ DR
P	BZ DE AZ
Q	DD
R	AQ DC DQ AL CE CF CV DS
S	AP AN AO CD DW DV
T	CB DB DP CC AD CY CW CX AE
U	CA AM BA
V	BB
W	CZ
X	BD
Y	DO DA
Z	BC

Gambar 2. Pilihan huruf homofon

Cipher abjad-majemuk dibuat dari sejumlah *cipher* abjad-tunggal, masing-masing dengan kunci yang berbeda.
Kebanyakan *cipher* abjad-majemuk adalah *cipher* substitusi periodik yang didasarkan pada periode *m*.

Plainteks:
$P = p1p2 \ldots pmpm+1 \ldots p2m \ldots$

Cipherteks:
$Ek(P) = f1(p1)\ f2(p2) \ldots fm(pm)\ fm+1(pm+1) \ldots f2m(p2m) \ldots$

Untuk $m = 1$, *cipher*-nya ekivalen dengan *cipher* abjad-tunggal.

Contoh *cipher* substitusi periodik adalah *cipher Vigenere*
Kunci: $K = k1k2 \ldots km$
ki untuk $1 \le i \le m$ menyatakan jumlah pergeseran pada huruf ke-i.

Karakter cipherteks $ci(p) = (p + ki) \bmod 26$ *(*)*. Misalkan periode $m = 20$, maka 20 karakter pertama dienkripsi dengan persamaan (*), setiap karakter ke-i menggunakan kunci ki. Untuk 20 karakter berikutnya, kembali menggunakan pola enkripsi yang sama.

Contoh: (spasi dibuang)
Plainteks :
KRIPTOGRAFIKLASIKDENGANCIPHER ALFABETMAJEMUK

Kunci :
LAMPIONLAMPIONLAMPIONLAMPIONL AMPIONLAMPIONL

Cipherteks : VR...

Perhitungan:
(K+L) mod 26 = (10 + 11) mod 26 = 21 = V
(R+A) mod 26 = (17 + 00) mod 26 = 17 = A
dst

Contoh 2: (dengan spasi)
Plainteks :
SHE SELLS SEA SHELLS BY THE SEASHORE

Kunci :

```
KEY KEYKE YKE YKEYKE YK EYK
EYKEYKEY
```

Cipherteks :

```
CLC CIJVW QOE QRIJVW ZI XFO
WCKWFYVC
```

Vigènere Cipher

Termasuk ke dalam *cipher* abjad-majemuk (*polyalpabetic substitution cipher*). Ditemukan oleh diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Vigènere pada abad 16. Sudah berhasil dipecahkan pada Abad 19. *Vigènere Cipher* menggunakan Bujursangkar *Vigènere* untuk melakukan enkripsi.

Setiap baris di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh dengan *Caesar Cipher.*

Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci diulang secara periodik. Bila panjang kunci adalah *m*, maka periodenya dikatakan *m*.

Contoh: kunci = fakultas

Plainteks:

```
TEKNOLOGI INFORMASI
```

Kunci:
fakultasf akultasfa

Hasil enkripsi seluruhnya adalah sebagai berikut:

Plainteks : `TEKNOLOGI INFORMASI`
Kunci : fakultasf akultasfa
Cipherteks : `YEKHZEOYN IXZZKMSXI`

Huruf yang sama tidak selalu dienkripsi menjadi huruf cipherteks yang sama pula.

Contoh:
huruf plainteks O dapat dienkripsi menjadi H atau X, dan huruf cipherteks N dapat merepresentasikan huruf plainteks H atau X

Hal di atas merupakan karakteristik dari *cipher* abjad-majemuk: setiap huruf cipherteks dapat memiliki kemungkinan banyak huruf plainteks. Pada *cipher* substitusi sederhana, setiap huruf cipherteks selalu menggantikan huruf plainteks

	plaintext																									
kunci	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 3. Bujur Sangkar Vigenere

tertentu.

Vigènere Cipher dapat mencegah frekuensi huruf-huruf di dalam cipherteks yang mempunyai pola tertentu yang sama seperti pada *cipher* abjad-tunggal. Jika periode kunci diketahui dan tidak terlalu panjang, maka kunci dapat ditentukan dengan menulis program komputer untuk melakukan *exhaustive key search*.

Contoh:
Diberikan cipherteks sbb:
TGCSZ GEUAA EFWGQ AHQMC

diperoleh informasi bahwa panjang kunci adalah *p* huruf dan plainteks ditulis dalam Bahasa Inggris, maka *running* program dengan mencoba semua kemungkinan kunci yang panjangnya tiga huruf, lalu periksa apakah hasil dekripsi dengan kunci tersebut menyatakan kata yang berarti. Cara ini membutuhkan usaha percobaan sebanyak 26*p* kali.

d. Polygram Substitution Cipher

Blok huruf plainteks disubstitusi dengan blok cipherteks.
Misalnya AS diganti dengan RT, BY diganti dengan SL. Jika unit huruf plainteks / cipherteks panjangnya 2 huruf, maka ia disebut digram (*biigram*), jika 3 huruf disebut ternari-gram, dst
Tujuannya : distribusi kemunculan poligram menjadi *flat* (datar), dan hal ini menyulitkan analisis frekuensi

Playfair Cipher
Termasuk ke dalam *polygram cipher*.

Ditemukan oleh Sir Charles Wheatstone dan Baron Lyon Playfair pada tahun 1854. *Cipher* ini mengenkripsi pasangan huruf (digram atau digraf), bukan huruf tunggal seperti pada *cipher* klasik lainnya. Tujuannya adalah untuk membuat analisis frekuensi menjadi sangat sulit sebab frekuensi kemunculan huruf-huruf di dalam cipherteks menjadi datar (*flat*).

Kunci kriptografinya 25 buah huruf yang disusun di dalam bujursangkat 5x5 dengan menghilangkan huruf J dari abjad.
Jumlah kemungkinan kunci: 25!=15.511.210.043.330.985.984.000.000

Contoh Kunci :

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

Gambar 4. kunci polygram cipher

Susunan kunci di dalam bujursangkar diperluas dengan menambahkan kolom keenam dan baris keenam.
Pesan yang akan dienkripsi diatur terlebih dahulu sebagai berikut:

1. Ganti huruf J (bila ada) dengan I
2. Tulis pesan dalam pasangan huruf (*bigram*).
3. Jangan sampai ada pasangan huruf yang sama. Jika ada, sisipkan Z ditengahnya
4. Jika jumlah huruf ganjil, tambahkan huruf Z di akhir

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

Gambar 5. kunci hasil perluasan

Contoh:
Plainteks: TEKNOLOGI INFORMASI
→ Tidak ada huruf J, maka langsung tulis pesan dalam pasangan huruf :
TE KN OL OG II NF OR MA SI

Ciperteks :
SR IS UF PF LL TI WF PS NK

Algoritma enkripsi:

1. Jika dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kanannya.
2. Jika dua huruf terdapat pada kolom kunci yang sama maka tiap huruf diganti dengan huruf di bawahnya.
3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan

kolom huruf kedua. Huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari 3 huruf yang digunakan sampai sejauh ini.

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

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

Gambar 6. algoritma enkripsi

Kunci dapat dipilih dari sebuah kalimat yang mudah diingat, misalnya :

`FAKULTAS TEKNOLOGI INFORMASI`

Buang huruf yang berulang dan huruf J kalau ada : `FAKULTESNOGIRM`

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

Gambar 7. kunci polygram berbeda

Lalu tambahkan huruf-huruf yang belum ada kecuali J : `FAKULTESNOGIRMBCDHNPQVWXZ`

Karena ada 26 huruf abjad, maka terdapat 26 x 26 = 677 bigram, sehingga identifikasi bigram individual lebih sukar. Ukuran poligram di dalam *Playfair cipher* tidak cukup besar, hanya dua huruf sehingga *Playfair cipher* tidak aman.

Meskipun *Playfair cipher* sulit dipecahkan dengan analisis frekuensi relatif huruf-huruf, namun ia dapat dipecahkan dengan analisis frekuensi pasangan huruf.
Dalam Bahasa Inggris kita bisa mempunyai frekuensi kemunculan pasangan huruf, misalnya pasangan huruf TH dan HE paling sering muncul.

Dengan menggunakan tabel frekuensi kemunculan pasangan huruf di dalam Bahasa Inggris dan cipherteks yang cukup banyak, *Playfair cipher* dapat dipecahkan.

Transpotition Cipher

Ciphereteks diperoleh dengan mengubah posisi huruf di dalam plaintekls. Dengan kata lain, algoritma ini melakukan *transpose* terhadap rangkaian huruf di dalam plainteks.

Nama lain untuk metode ini adalah permutasi, karena *transpose* setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut.

Contoh 1:
Misalkan plainteks adalah

```
FAKULTAS TEKNOLOGI INFORMASI
```

Enkripsi :

```
FAKULT
ASTEKN
OLOGII
NFORMA
SIZZZZ
```

Cipherteks : (baca secara vertikal)

```
FAONSASLFIKTOOZUEGRZLKIMZTNIA
FAON SASL FIKT OOZU EGRZ LKIM ZTNI AZ
```

Dekripsi: Bagi panjang cipherteks dengan kunci.
(Pada contoh ini, 30 / 6 = 5)

```
FAONS
ASLFI
KTOOZ
UEGRZ
LKIMZ
TNIAZ
```

Plainteks: (baca secara vertikal)

```
FAKULTAS TEKNOLOGI INFORMASI
```

Contoh 2:
Plainteks: `TEKNOLOGI INFORMASI`

Bagi menjadi blok-blok 8-huruf. Jika < 8, tambahkan huruf palsu.

Cipherteks:

```
GEKONLOTAINOFRMIFIACBDES
```

Contah 3 :
Misalkan plainteks adalah :

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
T	**E**	**K**	**N**	**O**	**L**	**O**	**G**	**I**	**I**	**N**	**F**	**O**	**R**	**M**	**A**	**S**	**I**	**A**	**B**	**C**	**D**	**E**	**F**

G	**E**	**K**	**O**	**N**	**L**	**O**	**T**	**A**	**I**	**N**	**O**	**F**	**R**	**M**	**I**	**F**	**I**	**A**	**C**	**B**	**D**	**E**	**S**
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8

Gambar 8. kunci transposisi

```
KRIPTOGRAFI DAN KEAMANAN DATA
```

Plainteks disusun menjadi 3 baris (k=3) seperti di bawah ini :

```
K   T   A   A   A   A   T
 R P O R F D N E M N N A A
  I   G   I   K   A   D
```

Substitution-Transpotition (Super Enkripsi)

Menggabungkan *cipher* substitusi dengan *cipher* transposisi.

Contoh.
Plainteks : `HELLO WORLD`

dienkripsi dengan *caesar cipher* menjadi `KHOOR ZRUOG`

kemudian hasil enkripsi ini dienkripsi lagi dengan *cipher* transposisi ($k = 4$):

```
KHOO
RZRU
OGZZ
```

Cipherteks akhir adalah:
`KROHZGORZOUZ`

Contoh program enkripsi

```
/* Program enkripsi file dengan Caesar
cipher */
#include <stdio.h>
main(int argc, char *argv[])
{
 FILE *Fin, *Fout;
 char p, c;
 int k;

 Fin = fopen(argv[1], "rb");
 if (Fin == NULL)
   printf("Kesalahan dalam membuka %s
sebagai berkas masukan/n",  argv[1]);
 Fout = fopen(argv[2], "wb");
 printf("\nEnkripsi   %s   menjadi   %s
...\n", argv[1], argv[2]);
 printf("\n");
 printf("k : ");
 scanf("%d", &k);
 while ((p = getc(Fin)) != EOF)
 {
  c = (p + k) % 256;
  putc(c, Fout);
 }

 fclose(Fin);
 fclose(Fout);
}
```

Contoh program dekripsi

```
/* Program dekripsi file dengan Caesar
cipher */

#include <stdio.h>

main(int argc, char *argv[])
{
 FILE *Fin, *Fout;
 char p, c;
 int n, i, k;

 Fin = fopen(argv[1], "rb");
 if (Fin == NULL)
   printf("Kesalahan dalam membuka %s
sebagai berkas masukan/n",  argv[1]);
 Fout = fopen(argv[2], "wb");
 printf("\nDekripsi   %s   menjadi   %s
...\n", argv[1], argv[2]);
 printf("\n");
 printf("k : ");
 scanf("%d", &k);
 while ((c = getc(Fin)) != EOF)
 {
  p = (c - k) % 256;
  putc(p, Fout);
 }
  fclose(Fin);
 fclose(Fout);
}
```

DAFTAR PUSTAKA

1. Handschuh, Helena. 1997. "*Cryptanalysis of the SEAL EncryptionAlgorithm*". Les Moulineaux, France : ENST, Computer Science Department.
2. Gollmann, Dieter. 1999. "*Computer Security*". London, England : John Willey & Sons Inc.
3. Gutmann, Peter. 2001. "*Cryptography and Data Security*". New Zealand : University of Auckland.

4. P. Rogaway and D. Coppersmith. 1994. "*A Software-Optimized Encryption Algorithm*". Cambridge Security Workshop : Springer-Verlag.
5. S. Tanenbaum, Andrew. 1996. "*Computer Networks 3ed edition*". New Jersey : Prentice Hall Inc.
6. Schneier, Bruce. 1996, "*Applied Cryptography 2nd edition*". Minneapolis : Wiley corp.