

# PENERAPAN KOMBINASI PLAYFAIR CIPHER DAN DIGRAPH CIPHER

Eka Yusrianto Toisutta - NIM : 13504116

Program Studi Teknik Informatika, Institut Teknologi Bandung

Jalan Ganesha 10, Bandung

email: if14116@students.if.itb.ac.id

**Abstract** – Makalah ini membahas mengenai pengkombinasian algoritma Playfair Cipher dan Digraph Cipher. Kedua algoritma tersebut merupakan algoritma kriptografi klasik yang melakukan substitusi terhadap kombinasi dua buah huruf. Acuan substitusi terhadap pasangan huruf tersebut berupa tabel. Hanya saja ukuran tabel yang digunakan oleh kedua algoritma tersebut berbeda, Digraph Cipher memakai tabel 26 x 26 sedangkan Playfair Cipher menggunakan tabel 5 x 5.

Pada dasarnya, kedua algoritma di atas sangatlah mudah untuk dipecahkan apabila berdiri sendiri. Terlebih lagi jika tabel acuan untuk melakukan substitusi pasangan huruf telah diketahui. Oleh karena itu, saya menawarkan solusi berupa penggabungan penggunaan kedua algoritma tersebut guna meningkatkan kompleksitas dari Cipher Teks yang akan dihasilkan oleh kombinasi keduanya.

**Kata Kunci:** Playfair Cipher, Digraph Cipher, Cipher Klasik, Kombinasi

## 1. PENDAHULUAN

Perkembangan dunia digital saat ini membuat lalu lintas pengiriman data semakin ramai. Hampir setiap orang melakukan transaksi data setiap harinya. Data yang dipertukarkan pun bervariasi mulai dari segi jenis data hingga ke tingkat kerahasiaan data tersebut. Tingginya tingkat pertukaran data ini mengakibatkan perlunya pengamanan terhadap data yang dipertukarkan agar tidak dapat disadap oleh pihak ketiga. Hingga saat ini telah banyak teknik-teknik yang dipergunakan untuk mengamankan data, baik teknik klasik maupun modern.

Playfair Cipher merupakan salah satu teknik klasik dalam pengamanan data. Teknik ini melakukan substitusi terhadap pasangan huruf sesuai dengan tabel acuan yang dibuat berdasarkan kata kunci yang telah disepakati oleh baik pengirim maupun penerima pesan tersebut.

Karena tabel acuannya berukuran 5 x 5, maka dari keseluruhan alphabet yang tersedia harus ada yang dikorbankan. Umumnya huruf yang dikorbankan adalah huruf “j” yang dianggap menjadi satu dengan huruf “i”. Plain text yang akan dienkripsi lalu dipilah-pilah menjadi pasangan huruf lalu kemudian dienkripsi berdasarkan tabel tersebut.

Digraph Cipher merupakan adaptasi terhadap teknik klasik lainnya yakni Caesar Cipher. Apabila dalam Caesar Cipher substitusi dilakukan terhadap setiap huruf, maka Digraph Cipher melakukan substitusi terhadap setiap pasangan huruf. Untuk mempermudah proses enkripsi dan dekripsi, maka dibentuklah tabel berukuran 26 x 26 yang menunjukkan daftar kombinasi pasangan huruf beserta substitusinya.

## 2. PLAYFAIR CIPHER DAN DIGRAPH CIPHER

### 2.1. Playfair Cipher

Playfair Cipher merupakan salah satu algoritma kriptografi klasik yang ditemukan oleh Sir Charles Wheatstone akan tetapi dipromosikan oleh Baron Lyon Playfair pada tahun 1854. Algoritma kriptografi ini mengenkripsi pasangan huruf, bukan huruf tunggal seperti pada algoritma kriptografi klasik lainnya. Tujuan utamanya adalah untuk mempersulit analisis frekuensi dengan menyetarakan jumlah frekuensi kemunculan huruf-huruf di dalam cipher text.

Kunci di dalam algoritma ini selalu memiliki panjang sebanyak 25 karakter yang disusun menjadi tabel acuan berukuran 5 x 5. Kunci ini mengandung seluruh huruf di dalam alphabet kecuali huruf “j” yang dileburkan dengan huruf “i”. Untuk mempermudah pembuatan kunci, pertama-tama tentukan kata yang akan menjadi kunci. Kata tersebut lalu dihilangkan seluruh huruf “j” yg terkandung di dalamnya. Kemudian kata tersebut di konkatenasi dengan teks “abcdefghijklmnopqrstuvwxyz” dan pada akhirnya dari hasil yang diperoleh akan dihilangkan seluruh kemunculan huruf yang berulang.

Kunci : GANESHA

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

Tabel 1 Contoh tabel acuan playfair cipher

Setelah tabel acuan terbentuk, berikutnya dilakukan operasi terhadap pesan yang akan dienkripsi, yaitu:

1. Ganti seluruh huruf “j” dengan huruf “i”
2. Tulis kembali pesan dalam bentuk pasangan huruf

- Untuk setiap pasangan huruf yang sama harus dipisah dan disisipkan huruf "z" di tengahnya
- Jika jumlah huruf ganjil, maka ditambahkan huruf "z" di akhir

Contoh pengolahan pesan sebelum dienkripsi

Pesan :

JALAN DAAN MOGOT MACET

Pesan setelah diolah:

IA LA ND AZ AN MO GO TM AC ET

Enkripsi yang dilakukan bergantung terhadap tipe pasangan huruf terhadap tabel acuan. Secara umum pasangan huruf tersebut terbagi atas tiga tipe yaitu keduanya di baris yang sama, keduanya di kolom yang sama, atau keduanya tidak berada di baris dan kolom yang sama.

- Jika pasangan huruf terdapat di baris yang sama, maka mereka akan disubsitusi dengan huruf yang pertama di sebelah kanan dari masing-masing huruf. Jika salah satu huruf terdapat di ujung kanan maka akan diganti dengan huruf paling kanan dari baris tersebut  
MO menjadi OI
- Jika pasangan huruf terdapat di kolom yang sama, maka mereka akan disubsitusi dengan huruf yang pertama di sebelah bawah dari masing-masing huruf. Jika salah satu huruf terdapat di ujung bawah maka akan diganti dengan huruf paling atas dari kolom tersebut  
ET menjadi DY
- Jika pasangan huruf tersebut tidak berada dalam kolom dan baris yang sama, maka digunakan pendekatan yang berbeda. Untuk mengganti huruf pertama, telusuri baris tempat huruf pertama hingga tiba pada kolom yang mengandung huruf kedua, huruf yang terdapat pada perpotongan antara kolom dan baris tersebut akan mengganti huruf pertama. Lakukan hal serupa untuk huruf kedua.  
IA menjadi KG

Pesan :

JALAN DAAN MOGOT MACET

Pesan setelah diolah:

IA LA ND AZ AN MO GO TM AC ET

Pesan setelah dienkripsi :

KG KN EC SW NT OI SI YT NB DY

Untuk proses dekripsinya, perlu diketahui tabel acuan terlebih dahulu. Lalu lakukan proses yang serupa dengan penyiapan plainteks sebelum dienkripsi. Setelah itu lakukan seperti halnya dengan proses enkripsi hanya saja jika arah pergerakannya dibalik kecuali untuk kasus ketiga dimana pasangan huruf tidak berada di kolom dan baris yang sama.

## 2.2. Digraph Cipher

Digraph Cipher merupakan Caesar Cipher yang diterapkan pada pasangan huruf. Jika dibandingkan dengan Caesar Cipher biasa dimana setiap huruf hanya memiliki kemungkinan untuk diganti oleh 26 huruf alfabet, jumlah kemungkinan yang ditawarkan oleh Digraph Cipher memang jauh lebih banyak yaitu 676 kemungkinan untuk setiap pasangan huruf.

Untuk mempermudah proses enkripsi dan dekripsi dengan menggunakan algoritma ini, biasanya dibentuk sebuah tabel acuan yang berukuran 26 x 26 yang berisikan seluruh pasangan untuk setiap kombinasi pasangan huruf dalam alfabet.

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

Gambar 1 Contoh tabel acuan Digraph Cipher

Seperti halnya Playfair Cipher, di dalam Digraph Cipher juga melakukan operasi terhadap pesan yang akan dienkripsi. Hanya saja operasi yang dilakukan tidak sebanyak operasi yang dilakukan pada Playfair Cipher. Dalam Digraph Cipher hal yang perlu dilakukan hanyalah menuliskan pesan yang akan dienkripsi ke dalam bentuk pasangan huruf. Dan bila jumlah huruf di dalamnya ganjil maka ditambahkan huruf "z" di akhir.

Pesan :

JALAN DAAN MOGOT MACET

Pesan setelah diolah:

JA LA ND AA NM OG OT MA CE TZ

Digraph Cipher melakukan proses enkripsi yang sangat sederhana yaitu melakukan substitusi terhadap pasangan huruf yang menjadi masukan sesuai dengan tabel acuan yang tersedia. Adapun cara penggunaan tabel acuan tersebut, yaitu :

- Huruf pertama mengacu kepada kolom yang merepresentasikan huruf tersebut
- Huruf kedua menggunakan baris yang merepresentasikan huruf tersebut
- Persilangan antara baris dan kolom yang diperoleh dari huruf pertama dan kedua. Akan

menghasilkan nilai yang akan menggantikan huruf pertama dan kedua tersebut

Pesan :

JALAN DAAN MOGOT MACET

Pesan setelah diolah:

JA LA ND AA NM OG OT MA CE TZ

Pesan setelah dienkripsi :

WG YG AD NG AU BA BN ZG PC GH

Untuk proses dekripsi dapat dilakukan dengan melakukan pencocokan terhadap kombinasi pasangan huruf yang terdapat dalam tabel dengan pasangan huruf dari berkas cipher text. Setelah itu lalu dilakukan penelusuran terhadap nilai yang diperoleh guna mengetahui nilai yang sebenarnya yang dimiliki baik huruf pertama maupun huruf kedua.

### 3. KOMBINASI PLAYFAIR CIPHER DAN DIGRAPH CIPHER

#### 3.1. Overview

Pengombinasian algoritma Playfair Cipher dan Digraph Cipher, yaitu penerapan algoritma Digraph Cipher terhadap hasil yang diperoleh dari penerapan algoritma Playfair Cipher. Pengombinasian ini juga dapat diterapkan dengan alur yang berlawanan yaitu menerapkan algoritma Playfair Cipher pada hasil yang diperoleh dari penerapan algoritma Digraph Cipher. Hanya saja, peniadaan huruf "j" pada Playfair Cipher dapat mempengaruhi hasil dekripsi sehingga tidak sesuai dengan teks awal yang mengalami proses enkripsi.

#### 3.2 Lingkungan Pengembangan Aplikasi

Program aplikasi pengkombinasian Playfair Cipher dan Digraph Cipher ini dikembangkan pada lingkungan sebagai berikut :

- AMD® Athlon™ 64 3000+
- Memory DDR 512 Mb
- Windows XP SP2 with Framework 2.0
- Kakas pembangunan Notepad++ dengan bahasa Java

#### 3.3 Algoritma Umum

##### 3.3.1 Pembuatan Tabel Acuan

Kedua algoritma di atas, baik Digraph Cipher maupun Playfair Cipher menggunakan tabel sebagai acuan dalam melakukan enkripsi dan dekripsi terhadap sebuah plainteks.

Berikut adalah algoritma yang digunakan dalam pembuatan kedua tabel acuan tersebut

```
public static char[][]
genPlayfairTable(String Key)
{
    int counter = 0;
    char tabel[][] = new char [5][5];
    Key = keyProcess(Key);
    while(counter < Key.length())
    {
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<5;j++)
            {
                tabel[j][i] =
                Key.charAt(counter);
                counter++;
            }
        }
    }
    Return tabel;
}
```

```
public static char[][]
genDigraphTable(int deltaX, int
deltaY)
{
    String tabel[][] = new String
[26][26];
    String huruf =
    "abcdefghijklmnopqrstuvwxy";
    char tabHuruf =
    huruf.toCharArray();
    int x = deltaX;
    int y = deltaY;

    for(int i=0;i<26;i++)
    {
        for(int j=0;j<26;j++)
        {
            if(x + j > 25)
                x = x + j -25;
            if(i - y < 0)
                y = i - y + 25;

            tabel[j][i] = "" +
            tabHuruf[x] + "" +
            tabHuruf[y];
        }
    }
    return tabel;
}
```

Seperti dijelaskan sebelumnya, tabel acuan pada Playfair Cipher terbentuk dari kata kunci yang dimasukkan oleh user. Pada awal tahap ini pertama-tama dilakukan pemrosesan kata kunci dengan menghilangkan huruf "j" lalu kemudian membuang kemunculan huruf yang berulang. Lalu pada akhirnya menambahkan huruf-huruf yang tidak terdapat di dalam kunci pada bagian akhir dari kunci tersebut.

Kata kunci yang telah diproses tersebut lalu dimasukkan ke dalam tabel yang berukuran 5 x 5. Setelah tabel tersebut telah terisi maka akan dikembalikan sebagai hasil dari fungsi `genPlayfairTable`.

Sedangkan dalam kasus Digraph Cipher, fungsi menerima masukan dua buah integer yaitu `deltaX` dan `deltaY` yang menandakan pergeseran huruf pada sumbu X dan `deltaY` yang merupakan nilai pergeseran huruf terhadap sumbu Y. nilai `deltaX` dan `deltaY` inilah yang menentukan kombinasi pasangan huruf yang akan disubsitusi. Pada tahap iterasi dilakukan pengisian tabel yang berukuran 26 x 26 tersebut dengan kombinasi pasangan huruf yang tersedia. Pengisian dilakukan dengan mengacu hasil perhitungan nilai baris dengan `deltaY` serta nilai kolom dengan `deltaX`. Contoh : jika `deltaX=13` dan `deltaY=6`, maka pasangan huruf "AB" akan disubsitusi dengan "NF" yang terdapat pada baris kedua kolom pertama.

### 3.3.2. Proses Enkripsi

Enkripsi pada kedua algoritma ini dilakukan dalam bentuk substitusi pasangan huruf sesuai dengan tabel acuan yang telah digenerate sebelumnya. Dalam proses enkripsi ini, yang berperan penting adalah kata kunci yang akan digunakan oleh algoritma Playfair Cipher dan juga simpangan terhadap sumbu X dan Y yang digunakan oleh Digraph Cipher.

Berikut adalah algoritma enkripsi yang saya ajukan sebagai metoda pengkombinasian Playfair Cipher dan Digraph Cipher.

```
public static string
encrypt(String text, String Key,
int deltaX, int deltaY)
{
String result = "";
String step1 = "";
step1 =
encryptPlayfair(text,Key);
result = encryptDigraph(step1,
deltaX,deltaY);
return result;
}
```

```
public static String
encryptPlayfair(String text,
String Key)
{
char tabel[][] =
genPlayfairTable(Key);
String src[] =
processTextPlayfair(text);
String result = "";

for(int i=0;i<src.length();i++)
{
if(sameRow(src[i],tabel))
result = result +
rowShiftEnc(src[i],tabel);
else if(sameLine(src[i],tabel))
result = result +
lineShiftEnc(src[i],tabel);
else
result = result +
lineRowShiftEnc(src[i],tabel);
}
return result;
}
```

```
public static String
encryptDigraph(String text, int
deltaX, int deltaY)
{
char tabel[][] =
genDigraphTable(deltaX,deltaY);
String src[] =
processTextDigraph(text);
String result = "";

for(int i=0;i<src.length();i++)
{
result = result +
subtituteDigraphEnc(src[i],tabel)
;
}
return result;
}
```

Proses enkripsi dimulai dengan mengenkripsi teks dengan menggunakan algoritma Playfair Cipher. Setelah itu hasil dari enkripsi Playfair akan dienkripsi kembali dengan menggunakan Digraph Cipher.

Enkripsi pada Playfair dimulai dengan mengubah teks asal menjadi teks yang telah siap untuk dienkripsi. Perubahan mencakup penggantian huruf "j" oleh huruf "i", penambahan huruf "z" jika dibutuhkan dan pada akhirnya memilah-milah teks menjadi pasangan-pasangan huruf. Setelah itu lalu mendefinisikan tabel acuan dengan mengacu pada kata kunci. Setelah itu pasangan-pasangan huruf lalu diperiksa dan ditentukan tipenya. Penentuan tipe ini penting karena pada Playfair Cipher terdapat 3 buah tipe enkripsi.

Setelah dienkripsi lalu pasangan-pasangan huruf tersebut disatukan kembali dan nilainya disimpan.

Selanjutnya hasil enkripsi tadi dimasukkan dalam proses enkripsi Digraph Cipher. Seperti halnya Playfair, langkah awal yang diambil dalam algoritma ini adalah pengubahan teks dan pembentukan tabel acuan. Setelah itu pasangan huruf yang terbentuk akan disubsitusikan berdasarkan tabel acuan. Setelah semuanya pasangan huruf selesai dienkripsi, maka hasilnya akan disatukan kembali dan disimpan.

### 3.3.3 Proses Dekripsi

Pada dasarnya proses dekripsi di dalam kasus ini hanyalah kebalikan dari fungsi enkripsi. Seperti halnya enkripsi, untuk mendekripsi pesan maka diperlukan tabel acuan yang sama dengan yang dipergunakan pada saat mengenkripsi pesan. Tapi dalam kasus ini hanya perlu diketahui kunci, deltaX dan deltaY.

Karena dilakukan dalam urutan terbalik, maka yang paling pertama dilakukan adalah pendekripsian dengan algoritma Digraph Cipher lalu hasilnya akan didekripsi oleh algoritma Playfair Cipher.

```
public static string
decrypt(String text, String Key,
int deltaX, int deltaY)
{
String result = "";
String step1 = "";
step1 =
decryptPlayfair(text,Key);
result = decryptDigraph(step1,
deltaX,deltaY);
return result;
}
```

```
public static String
decryptDigraph(String text, int
deltaX, int deltaY)
{
char tabel[][] =
genDigraphTable(deltaX,deltaY);
String src[] =
processTextDigraph(text);
String result = "";

for(int i=0;i<src.length();i++)
{
result = result +
subtituteDigraphDec(src[i],tabel)
;
}
return result;
}
```

```
public static String
encryptPlayfair(String text,
String Key)
{
char tabel[][] =
genPlayfairTable(Key);
String src[] =
processTextPlayfair(text);
String result = "";

for(int i=0;i<src.length();i++)
{
if(sameRow(src[i],tabel))
result = result +
rowShiftDec(src[i],tabel);
else if(sameLine(src[i],tabel))
result = result +
lineShiftDec(src[i],tabel);
else
result = result +
lineRowShiftDec(src[i],tabel);
}
return result;
}
```

Pada pendekripsian Digraph Cipher, urutan kerjanya sama seperti pada tahap enkripsinya. Hanya saja pada tahap pencarian kombinasi huruf awal, yang dilakukan adalah merunut balik langkah-langkah yang ditempuh dengan tetap mengacu pada tabel acuan yang sama. Selanjutnya hasil dekripsi ini diteruskan untuk di dekripsi oleh pendekripsian Playfair.

Seperti pendekripsian Digraph Cipher, urutan kerja yang dilakukan secara umum sama persis dengan proses enkripsi. Bahkan perlakuan yang berbeda untuk ketiga tipe pasangan huruf tersebut tetap berlaku. Hanya saja arah pencarian dilakukan secara terbalik. Sebagai contoh pada saat enkripsi jika pasangan huruf berada di baris yang sama maka kedua huruf tersebut akan disubsitusi oleh huruf pertama di sebelah kanannya. Sedangkan pada saat mendekripsi, maka subsitusi akan dilakukan terhadap huruf pertama di sebelah kiri masing-masing huruf.

Karena Playfair Cipher meleburkan huruf "j" dengan huruf "i". Maka sering kali perlu dilakukan penalaran secara manual terhadap hasil dekripsi yang bernilai "i". Apakah nilai sebenarnya adalah "i" atukah "j".

### 3.4 Kekuatan Algoritma

Pengombinasian kedua algoritma ini memberikan jumlah kemungkinan pasangan subsitusi yang lebih banyak dibandingkan apabila kedua algoritma tersebut hanya dipergunakan secara sendiri-sendiri. Selain itu, faktor penentu hasil enkripsi tidak hanya satu buah input, melainkan memerlukan tiga buah input sehingga apabila salah satu dari ketiganya bocor, hasil enkripsi yang dicapai masih sulit untuk dipecahkan

oleh pihak ketiga tersebut.

### 3.5 Kelemahan Algoritma

Seperti halnya cipher-cipher klasik lainnya, kedua algoritma ini masih tergolong sangat mudah untuk dipecahkan. Bahwa sifat tersebut pun tidak akan secara otomatis menghilang walaupun sudah dikombinasikan satu sama lainnya. Hal ini terjadi karena pada dasarnya pada algoritma cipher klasik pada umumnya hanya bertujuan untuk mensubstitusikan huruf dengan huruf lain setelah melalui tahap-tahap tertentu. Dengan mengetahui tabel acuan yang dipergunakan ataupun langkah-langkah enkripsi dari algoritma kombinasi ini, pihak yang tidak bertanggung jawab tetap dapat memecahkannya.

## 4. KESIMPULAN DAN SARAN

Dari pembahasan diatas maka kesimpulan yang dapat diambil, antara lain:

1. Proses enkripsi dan dekripsi Digraph Cipher dapat dilakukan walaupun tanpa membentuk tabel acuan, selama mengetahui seberapa besar simpangan yang terjadi terhadap huruf-huruf baik di kolom maupun baris.
2. Urutan pelaksanaan enkripsi pada algoritma kombinasi sangatlah penting karena perbedaan dari domain tabel acuan yang digunakan oleh

kedua algoritma ini.

3. Pengkombinasian dua atau lebih algoritma cipher klasik dapat menambahkan tingkat kerumitan hasil enkripsi

Adapun saran antara lain:

1. Penggunaan tabel acuan di dunia nyata memang cukup memudahkan. Hanya saja jika diterapkan dalam bentuk program maka perlu diperhatikan dimensi dari tabel tersebut karena semakin besar dimensinya, waktu yang dibutuhkan untuk memproses isinya pun akan meningkat.
2. Algoritma cipher klasik sudah tidak lagi dapat diterapkan di dunia nyata dikarenakan kemudahannya untuk dipecahkan. Hanya saja konsep-konsep dasar dari Algoritma Cipher klasik masih dapat diterapkan dalam Algoritma cipher modern.

## DAFTAR REFERENSI

- [1]Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.
- [2][http://en.wikipedia.org/Playfair\\_cipher](http://en.wikipedia.org/Playfair_cipher)
- [3][http://simonsingh.net/The\\_Black\\_Chamber/playfair\\_cipher.com](http://simonsingh.net/The_Black_Chamber/playfair_cipher.com)
- [4][http://simonsingh.net/The\\_Black\\_Chamber/digraph\\_cipher.com](http://simonsingh.net/The_Black_Chamber/digraph_cipher.com)