ISSN 1693 - 227

FAHMA

JURNAL TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

Volume 10, Nomor 1 Januari 2012

Diterbitkan oleh:
Lembaga Penelitian dan Pengabdian Masyarakat
STIMIK EL RAHMA YOGYAKARTA

Jurnal FAHMA Volume 10 Nomor 1 Januari 2012

Jurnal FAHMA merupakan jurnal di bidang teknologi informasi dan ilmu komputer beserta rumpun keilmuannya. Diterbitkan oleh LP2M STMIK EL RAHMA dengan frekuensi terbit setahun tiga kali pada bulan Januari, Mei dan September.

Alamat redaksi: Unit LP2M (Lembaga Penelitian dan Pengabdian Masyarakat)
STMIK EL RAHMA Jl. Sisingamangaraja No. 76 Yogyakarta
e-mail: lp2m@stmikelrahma.ac.id Telepon/fax: 0274 – 377982

DAFTAR ISI

IMPLEMENTASI ALGORITMA KRIPTOGRAFI KLASIK KE DALAM BAHASA PEMROGRAMAN PHP

Asih Winantu
STMIK EL RAHMA
e-mail: asihwinantu@gmail.com

Abstract

Before computers existed, cryptography-based algorithms performed by the characters. The algorithm used belongs to the symmetry and the cryptographic system used long before the public key cryptographic system is found. There are a number of algorithms that are recorded in the history of cryptography so-called classical cryptographic algorithm, but now the algorithm are not widely used because they are very easily to be solved. Classical system algorithm that will be discussed in this paper are: Shift Cipher, Hill Cipher, Affine Cipher, Substitution Cipher and Vigenere Cipher.

Keywords: *Cryptography, The Classical System, Cipher*

PENDAHULUAN

Masalah keamanan (*security*) pada komputer menjadi isu penting pada era teknologi informasi saat ini. Banyak kejahatan *cyber* yang pernah kita dengar dari media massa (terutama kita baca beritanya didalam portal berita di internet). Kriptografi merupakan dasar untuk memahami keamanan pada komputer. Begitu pentingnya kriptografi untuk keamanan informasi, sehingga jika berbicara mengenai masalah keamanan yang berkaitan dengan penggunaan komputer, maka orang tidak bisa memisahkannya dengan kriptografi.

Kriptografi merupakan seni dan ilmu menyembunyikan informasi dari penerima yang tidak berhak. Algoritma kriptografi klasik termasuk kedalam sistem kriptografi simetri dan digunakan jauh sebelum sistem kriptografi kunci publik ditemukan. Dalam algoritma kriptografi klasik terdapat metode cipher substitusi. Di dalam metode cipher substitusi setiap unit plainteks diganti dengan satu unit cipherteks. Satu unit disini bisa berarti satu huruf, pasangan huruf, atau kelompok lebih dari dua huruf. Algoritma cipher substitusi tertua yang diketahui adalah Caesar Cipher yang digunakan oleh kaisar Romawi, Julius Caesar untuk menyandikan pesan yang ia kirim kepada para gubernurnya.

METODE PENELITIAN

Algoritma Kriptografi Klasik

Sebelum komputer ada, kriptografi dilakukan dengan menggunakan pensil dan kertas. Algoritma kriptografi (*cipher*) yang digunakan saat itu, dinamakan juga algoritma klasik, adalah berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Pada dasarnya, algoritma kriptografi klasik dapat dikelompokkan kedalam dua macam *cipher* yaitu :

1. *Cipher* Substitusi (*Substitution Ciphers*)
2. *Cipher* Transposisi (*Transposition* Ciphers)

Cipher Substitusi

Di dalam *Cipher* Substitusi setiap unit plainteks diganti dengan satu unit cipherteks. satu unit disini bisa berarti satu huruf, pasangan huruf, atau kelompok lebih dari dua huruf (Munir, 2006). Algoritma substitusi tertua yang diketahui adalah *Caesar cipher* yang digunakan oleh kaisar Romawi, Julius Caesar untuk menyandikan pesan yang ia kirim kepada para gubernurnya.

Cipher substitusi dapat dikelompokkan kedalam 4 (empat) jenis yaitu : cipher alphabet-tunggal, cipher alphabet-majemuk, cipher substitusi homofonik dan cipher substitusi poligram.

1. *Cipher* Alphabet-tunggal : Pada *cipher* alphabet-tunggal (*monoalphabetic cipher*) atau disebut juga *cipher* substitusi sederhana, satu huruf di plainteks diganti dengan tepat satu huruf cipherteks. Jadi fungsi *ciphering*-nya adalah fungsi satu-ke-satu (Munir, 2006).
2. *Cipher* Alphabet-majemuk : Pada *cipher* alphabet-majemuk (*polyalphabetic cipher*) merupakan cipher substitusi ganda yang melibatkan penggunaan kunci berbeda (Munir, 2006). *Cipher* alphabet-majemuk dibuat dari sejumlah cipher alphabet-tunggal, masing-masing dengan kunci yang berbeda.
3. *Cipher* Substitusi Homofonik *: Cipher* substitusi homofonik (*homophonic substitution cipher*) adalah seperti *cipher* alphabet-tunggal, kecuali bahwa setiap huruf didalam plainteks dapat dipetakan kedalam salah satu dari unit cipherteks yang mungkin (Munir, 2006). Maksudnya, setiap huruf plainteks dapat memiliki lebih dari satu kemungkinan unit cipherteks. Huruf yang paling sering muncul dalam teks mempunyai lebih banyak pilihan unit cipherteks. Jadi fungsi *ciphering*-nya memetakan satu-ke-banyak (*one-to-many*).
4. *Cipher* Substitusi Poligram *: Cipher* substitusi poligram (*polygram substitution cipher*), setiap unit huruf disubstitusi dengan unit huruf cipherteks (Munir, 2006). Jika unit huruf plainteks/cipherteks panjangnya 2 huruf maka disebut *digram*, jika 3 huruf disebut *trigram,* dan seterusnya (blok cipherteks tidak perlu harus sama panjang dengan blok plainteks). Keuntungannya, distribbusi kemunculan poligraf menjadi *flat* (datar), dan hal ini menyulitkan analisis frekuensi.

Monoalphabetic cipher

Shift cipher

Dalam sejarahnya, shift cipher pernah digunakan pada masa Romawi kuno dalam pemerintahan Julius Caesar. Metode yang digunakan sangatlah sederhana, yaitu dengan menggeser setiap huruf dalam plainteksnya. Misalkan x adalah plainteks dalam bentuk bilangan, K adalah kunci dengan $0 \leq K \leq 25$ dan y adalah cipherteks dalam bentuk bilangan. Proses enkripsi diberikan dalam fungsi

$$eK(x) = (x + K) \bmod 26$$

dan proses dekripsi diberikan dalam fungsi

$$dK(y) = (y - K) \bmod 26.$$

Untuk kunci K = 3, shift cipher sering disebut dengan Caesar Cipher, dan untuk K = 13 sering disebut dengan Rot-13 cipher. Sebagai contoh, enkripsi plainteks "saya" menggunakan K = 3 menghasilkan cipherteks "vdbd".

Affine cipher

Sandi Affine (Affine Cipher) merupakan kejadian khusus dari Sandi Substitusi. Pada sandi ini fungsi enkripsi didefinisikan sebagai berikut.

$$e(x) = ax + b \bmod 26$$

dengan kuncinya $a, b \in \mathbb{Z}_{26}$. Fungsi seperti ini disebut dengan fungsi affine. Perhatikan bahwa jika nilai a=1, maka fungsi enkripsinya merupakan sandi geser (shift cipher). Selanjutnya, bagaimana proses dekripsinya? Diberikan $y \in \mathbb{Z}_{26}$ maka persamaan:

$$ax + b \equiv y (\bmod 26)$$

haruslah mempunyai penyelesaian tunggal untuk x. Perhatikan bahwa persamaan di atas ekuivalen dengan:

$$ax \equiv y - b (\bmod 26)$$

Dari persamaan terakhir, diperoleh bahwa a haruslah mempunyai invers (pergandaan) yaitu a^{-1}, atau dengan kata lain nilai $\gcd(a, 26) = 1$. Jadi, secara umum Sandi Affine dituliskan sebagai berikut.

$$K = (a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26}, \gcd(a, 26) = 1$$
$$e_K(x) = ax + b \bmod 26$$
$$d_K(y) = a^{-1}(y - b) \bmod 26$$
$$x, y \in \mathbb{Z}_{26}$$

Untuk mencari nilai invers dapat digunakan algoritma Euclide yang diperluas (*Extended Euclidean Algorithm*). Misal diberikan kunci K = (7,3). Dapat diperoleh bahwa invers dari 7 mod 26 adalah 15. Maka fungsi enkripsinya adalah.

$$e_K(x) = 7x + 3 \bmod 26$$
$$d_K(y) = 15(y - 3) \bmod 26 = 15y - 19 \bmod 26$$

Misal diberikan plainteks "hot", h=7, o=14, dan t=19. Maka proses enkripsinya adalah:

$$7 \times 7 + 3 \bmod 26 = 52 \bmod 26 = 0$$
$$7 \times 14 + 3 \bmod 26 = 101 \bmod 26 = 23$$
$$7 \times 19 + 3 \bmod 26 = 136 \bmod 26 = 6$$

Diperoleh cipherteks 0-23-6 atau "AXG".

Polyalphabetic cipher
Hill cipher

Hill Cipher diperkenalkan pertama kali pada tahun 1929 oleh Lester S. Hill. Proses enkripsi dan dekripsi pada Hill Cipher menggunakan operasi perkalian matriks atas ring Z26. Ide dasar dari Hill Cipher adalah untuk membuat kombinasi linear dari plainteks untuk mendapatkan cipherteks. Kunci yang digunakan berupa matriks persegi atas Z26 yang determinannya invertibel pada Z26.

Diberikan kunci $K = \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mm} \end{bmatrix}$,

plainteks x = x1x2...xm dan cipherteks y = y1y2...ym. Proses enkripsi diberikan dalam fungsi

$$e_K(x) = \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

dan proses dekripsi diberikan dalam fungsi

$$d_K(y) = \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mm} \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

Syarat bahwa determinan kunci harus merupakan elemen yang invertibel dalam ring Z26 yaitu agar matriks kunci tersebut mempunyai invers atas Z26.

Vigenere cipher

Vigenere Cipher merupakan pengembangan dari shift cipher. Dalam vigenere cipher, kunci yang digunakan berupa beberapa pergeseran yang direpresentasikan dengan suatu kata kunci. Diberikan kata kunci K = k1k2...km, plainteks x = x1x2...xm dan cipherteks y = y1y2...ym. Proses enkripsi diberikan dalam fungsi

eK(x) = ((x1 + k1) mod 26, (x2 + k2) mod 26,..., (xm + km) mod 26)

dan proses dekripsi diberikan dalam fungsi

dK(y) = ((y1 - k1) mod 26, (y2 - k2) mod 26,..., (ym - km) mod 26)

Sebagai contoh, diberikan plainteks "selamat belajar kriptografi" dan kunci "rahasia", maka diperoleh cipherteks "jesaeit sesabir brpplwgiami".

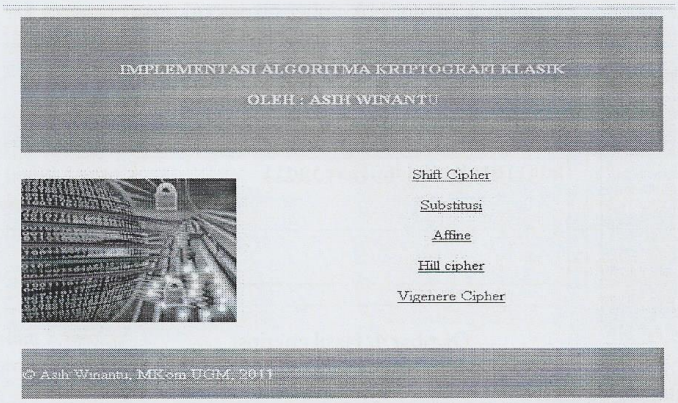
HASIL DAN PEMBAHASAN

Penjelasan program

Aplikasi cipher algoritma klasik dalam tugas ini dibuat dalam bahasa pemrograman PHP. Program disajikan dalam bentuk aplikasi berbasis web. Aplikasi ini mempunyai 6 halaman utama, yaitu :

a. Halaman awal (index.php)

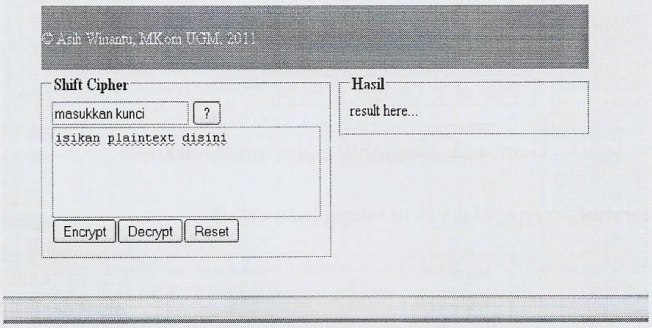
Halaman awal ini berisi menu-menu yang dapat dipilih oleh pengguna program. Menu adalah berupa list dari aplikasi algoritma cipher klasik, jika pengguna program mengklik salah satu menu, maka akan tampil aplikasi cipher sesuai pilihan, dibawah menu utama.



Gambar 1. Halaman awal alggoritma kriptografi klasik

b. Shift cipher (shift.php)

Halaman ini menampilkan aplikasi sistem shift cipher.

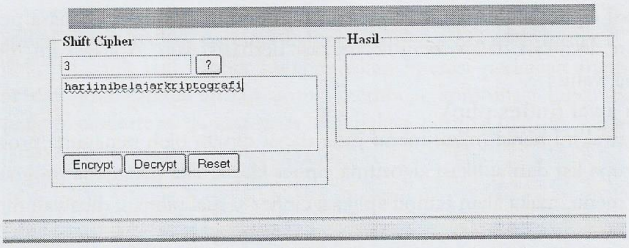


Gambar 2. Menu shift cipher

Untuk melakukan enkripsi :

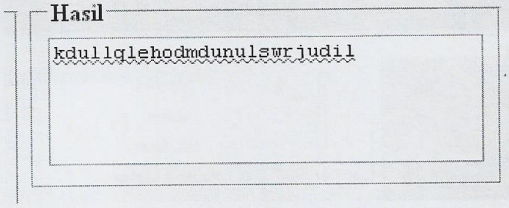
1. Isikan kunci, misalkan kita masukkan kunci = 3

2. Masukkan plaintext, misalkan plaintext = hariinibelajarkriptografi,



Gambar 3. Menu shift cipher proses enkripsi

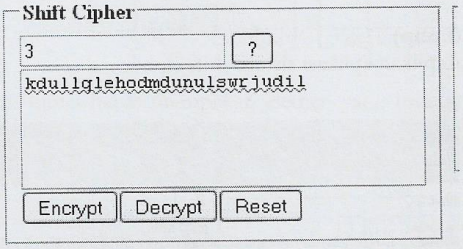
3. Lalu klik **encrypt.** Hasil enkripsi akan tampil di kotak sebelah kanan.



Gambar 4. Hasil enkripsi

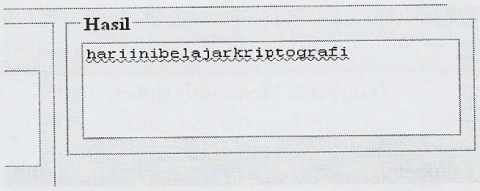
Untuk melakukan dekripsi :

1. Masukkan ciphertext yang akan di dekripsikan



Gambar 5. Menu shift cipher proses dekripsi

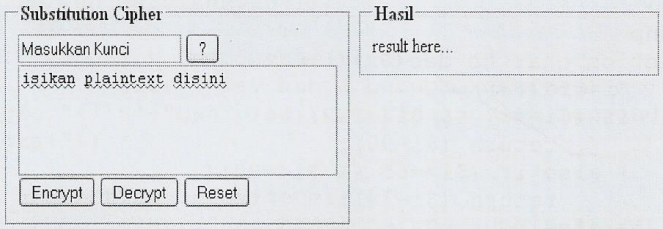
2. Klik tombol decrypt, Maka akan tampil hasil dekripsi



Gambar 6. Hasil dekripsi

c. Substitusi cipher

Untuk menggunakan aplikasi cipher substitusi, langkah yang digunakan sama dengan shift cipher:



Gambar 7. Menu substitusi cipher

Enkripsi :

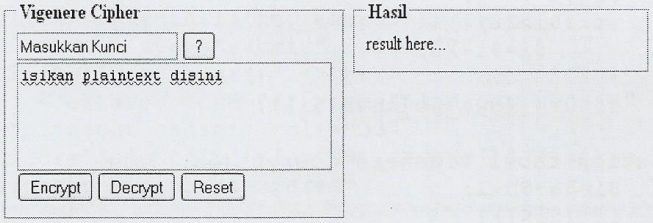
1. isikan kunci
2. masukkan plainteks
3. klik tombol *encrypt*

Dekripsi :

1. masukkan *ciphertext*
2. klik tombol *decrypt*

d. Vigenere cipher

Untuk menggunakan aplikasi *cipher Hill*, langkah yang digunakan:



Gambar 8. Menu Vigenere cipher

Enkripsi :

1. isikan kunci
2. masukkan plainteks
3. klik tombol encrypt

Dekripsi :

1. masukkan ciphertext
2. klik tombol decrypt

Listing program

Berikut ini adalah contoh listing program yang dipakai dalam implementasi algoritma sistem klasik.

Convert.Php

```
<?php
function char_to_dec($a){
     $i=ord($a);
     if ($i>=97 && $i<=122){
          return ($i-96);
     } else if ($i>=65 && $i<=90){
          return ($i-38);
     } else {
          return null;
     }
}

function dec_to_char($a){
     if ($a>=1 && $a<=26){
          return (chr($a+96));
     } else if ($a>=27 && $a<=52){
          return (chr($a+38));
     } else {
          return null;
     }
}

function tabel_vigenere_encrypt($a, $b){
     $i=$a+$b-1;
     if ($i>26){
          $i=$i-26;
     }
     return (dec_to_char($i));
}
function tabel_vigenere_decrypt($a, $b){
     $i=$a-$b+1;
     if ($i<1){
          $i=$i+26;
     }
     return (dec_to_char($i));
}

?>
```

Shift.php

```
     <script type="text/javascript">
     function SelectAll(id){
          document.getElementById(id).focus();
```

```
          document.getElementById(id).select();
     }
     function Info(){
          alert("Original code by :"+'\n\n'+"Ahmad
Zafrullah Mardiansyah");
     }
     function InfoCaesar(){
          alert("Key hanya berupa kombinasi
angka,"+'\n'+"dan plan text tidak boleh mengandung
angka!");
     }
     function InfoVigenere(){
          alert("Key hanya berupa kombinasi kata,
tidak boleh mengandung angka,"+'\n'+"dan plan text
tidak boleh mengandung angka!");
     }
     </script>
</head>

<body>
     <center>
     </center>
     <table width="600" align="center">
     <tr><td width="50%" valign="top">
     <fieldset>
     <legend><b>Shift Cipher </b></legend>
     <form action="" method="post">
     <input type="text" name="key_caesar"
id="key_caesar" value="masukkan kunci"
onClick="SelectAll('key_caesar')" />
     <input type="submit" value="?"
onClick="InfoCaesar()" /><br/>
     <textarea rows="4" name="plantext_caesar"
id="plantext_caesar" cols="33"
onClick="SelectAll('plantext_caesar')" >isikan
plaintext disini</textarea>
     <br/>
     <input type="submit" name="encrypt_caesar"
value="Encrypt" /><input type="submit"
name="decrypt_caesar" value="Decrypt" /><input
type="reset" value="Reset" />
     </form>
     </fieldset>
     </td><td valign="top" colspan="3">
     <fieldset>
     <legend><b>Hasil</b></legend>
<?php
     //-----------------------------------------------
-----------------//
```

```
     // caesar
                                   //
     //-----------------------------------------------
-----------------//
          if((isset($_POST['key_caesar'])) &&
(isset($_POST['plantext_caesar'])) &&
isset($_POST['encrypt_caesar'])){
               $key=$_POST['key_caesar'];
               $plantext=$_POST['plantext_caesar'];
               $split_key=str_split($key);
               $i=0;
               $split_chr=str_split($plantext);
               while ($key>52){
                    $key=$key-52;
               }
               foreach($split_chr as $chr){
                    if (char_to_dec($chr)!=null){

     $split_nmbr[$i]=char_to_dec($chr);
                    } else {
                         $split_nmbr[$i]=$chr;
                    }
                    $i++;
               }
               echo '<textarea rows="4" id="result"
cols="33" onclick="SelectAll(\'result\')" >';
               foreach($split_nmbr as $nmbr){
                    if (($nmbr+$key)>52){
                         if
(dec_to_char($nmbr)!=null){
                              echo
dec_to_char(($nmbr+$key)-52);
                         } else {
                              echo $nmbr;
                         }
                    } else {
                         if
(dec_to_char($nmbr)!=null){
                              echo
dec_to_char($nmbr+$key);
                         } else {
                              echo $nmbr;
                         }
                    }
               }
               echo '</textarea><br/>';
          } else if ((isset($_POST['key_caesar'])) &&
(isset($_POST['plantext_caesar'])) &&
isset($_POST['decrypt_caesar'])){
```

```
          } else if ((isset($_POST['key_vigenere']))
&& (isset($_POST['plantext_vigenere'])) &&
(isset($_POST['encrypt_vigenere']))){
               $key=$_POST['key_vigenere'];
               $plantext=$_POST['plantext_vigenere'];
               $len_key=strlen($key);
               $len_plantext=strlen($plantext);
               $split_key=str_split($key);
               $split_plantext=str_split($plantext);

               echo '<textarea rows="4" id="result"
cols="33" onclick="SelectAll(\'result\')" >';
               $i=0;
               for($j=0;$j<$len_plantext;$j++){
                    if ($i==$len_key){
                         $i=0;
                    }
                    $split_key2[$j]=$split_key[$i];
                    $i++; }             }
               for($k=0;$k<$len_plantext;$k++){
                    $a=char_to_dec($split_key2[$k]);

     $b=char_to_dec($split_plantext[$k]);
                    if (($a && $b)!=null){
                         echo
(tabel_vigenere_encrypt($a, $b));
                    } else {
                         echo $split_plantext[$k];
     }
               }
               echo '</textarea><br/>';
          } else if ((isset($_POST['key_vigenere']))
&& (isset($_POST['plantext_vigenere'])) &&
(isset($_POST['decrypt_vigenere']))){
               $key=$_POST['key_vigenere'];
               $plantext=$_POST['plantext_vigenere'];
               $len_key=strlen($key);
               $len_plantext=strlen($plantext);
               $split_key=str_split($key);
               $split_plantext=str_split($plantext);

               echo '<textarea rows="4" id="result"
cols="33" onclick="SelectAll(\'result\')" >';
               $i=0;
               for($j=0;$j<$len_plantext;$j++){
                    if ($i==$len_key){
                         $i=0;
                    }
                    $split_key2[$j]=$split_key[$i];
```

```
                    $i++;
                }

                for($k=0;$k<$len_plantext;$k++){
                    $a=char_to_dec($split_key2[$k]);

    $b=char_to_dec($split_plantext[$k]);
                    if (($a && $b)!=null){
                        echo
(tabel_vigenere_decrypt($b, $a));
                    } else {
                        echo $split_plantext[$k];
                    }
                }

                echo '</textarea><br/>';

            } else {
                echo "result here...";
            }
        ?>
        </fieldset>
```

KESIMPULAN

Sebelum komputer ada, kriptografi dilakukan dengan algoritma berbasis karakter. Algoritma yang digunakan termasuk ke dalam sistem kriptografi simetri dan digunakan jauh sebelum sistem kriptografi kunci publik ditemukan.

Terdapat sejumlah algoritma yang tercatat dalam sejarah kriptografi sehingga dinamakan algoritma kriptografi klasik, namun sekarang algoritma tersebut sudah tidak banyak dipakai karena ia sangat mudah dipecahkan.

Tiga alasan mempelajari algoritma kriptografi klasik:
1. Untuk memberikan pemahaman konsep dasar kriptografi.
2. Dasar dari algoritma kriptografi modern.
3. Dapat memahami potensi-potensi kelemahan sistem chiper.

DAFTAR PUSTAKA

Munir, R., 2006, Kriptografi, Penerbit Informatika, Bandung

Mahyudin, R., 2009, Studi dan Perbandingan Algoritma ADFGVX Cipher Dengan Algoritma Playfair Cipher Pada Perang Dunia I, ITB, Bandung

Stalling, W., 2005, Cryptography and Network Security Principles and Practices, Prentice Hall, New Jersey