# STUDI DAN PERBANDINGAN PERFORMANSI ALGORITMA SIMETRI VIGENERE CHIPPER BINNER DAN HILL CHIPPER BINNER

Ivan Nugraha – NIM : 13506073

*Program Studi Teknik Informatika, Institut Teknologi Bandung*
*Jl. Ganesha 10, Bandung*
E-mail : if16073@students.if.itb.ac.id

## Abstrak

Sebelum adanya komputer, kriptografi dilakukan berbasis karakter dengan hanya menggunakan kertas dan pena. Algoritma kriptografi yang digunakan saat itu termasuk dalam sistem kriptografi kunci simetri dan digunakan jauh sebelum ditemukannya sistem kunci publik. Algoritma kriptografi yang berbasis karakter biasanya termasuk dalam salah satu dari cipher substitusi, cipher transposisi, atau super enkripsi yang merupakan gabungan dari cipher substitusi dan cipher transposisi.

Salah satu algoritma kriptografi kunci simetri berbasis karakter yang cukup lama digunakan adalah algoritma kriptografi Vigènere Cipher. Algoritma ini dapat dikatakan baik karena termasuk cipher abjad-majemuk yang merupakan salah satu bagian dari cipher substitusi. Namun, saat ini algoritma ini sudah tidak digunakan lagi karena pada tahun 1863 Freidrich Kasiski telah menemukan cara memecahkan Vigènere Cipher.

Algoritma kriptografi kunci simetris lainnya adalah *Hill Cipher*. Algoritma *Hill Cipher* menggunakan matriks berukuran m x m sebagai kunci untuk melakukan enkripsi dan dekripsi. Dasar teori matriks yang digunakan dalam *Hill Cipher* antara lain adalah perkalian antar matriks dan melakukan invers pada matriks. Pada Algoritma *Hill Cipher*, chippertext-only attack sangat sulit untuk melakukan kriptanalis. Namun, dengan *know-plaintext attack*, plaintext masih mungkin ditemukan.

Makalah ini membahas mengenai perbandingan antara kriptografi Vigènere dan kriptografi *Hill Cipher* yang keduanya diperluas dengan menggunakan bilangan biner. Dibandingkan pula performansi dari kedua teknik tersebut dengan teknik kriptanalis terhadap keduanya. Dalam makalah ini, algoritma kriptografi Vigenere yang dirancang akan diujikan dengan metode yang biasanya digunakan untuk melakukan kriptanalis terhadap ciphertext hasil enkripsi dari algoritma kriptografi Vigènere, yaitu metode Kasiski. Sedangkan metode *Hill Cipher* diujikan dengan metode *know-plaintext attack*.

***Kata kunci**: Vigenere Cipher, Hill Cipher, biner, kasiski, chippertext-only attack, know-plaintext attack.*

## 1. Pendahuluan

Kriptografi berasal dari Bahasa Yunani: *"cryptós"* artinya rahasia, sedangkan *"gráphein"* artinya tulisan. Jadi, secara morfologi kriptografi berarti tulisan rahasia. [1]

Ada beberapa definisi kriptografi yang telah dikemukakan di dalam berbagai literatur. Definisi yang kita pakai di dalam makalah ini: Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Kata "seni" di dalam definisi di atas berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal.

## 2. Teknik *Hill Cipher*

*Hill Cipher* merupakan penerapan aritmatika modulo pada kriptografi. Teknik kriptografi ini menggunakan sebuah matriks persegi sebagai kunci yang digunakan untuk melakukan enkripsi dan dekripsi.

*Hill Cipher* diciptakan oleh Lester S. Hill pada tahun 1929 [2]. Teknik kriptografi ini diciptakan dengan maksud untuk dapat menciptakan *cipher*

(kode) yang tidak dapat dipecahkan menggunakan teknik analisis frekuensi. *Hill Cipher* tidak mengganti setiap abjad yang sama pada *plaintext* dengan abjad lainnya yang sama pada *ciphertext* karena menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya.

*Hill Cipher* termasuk kepada algoritma kriptografi klasik yang sangat sulit dipecahkan oleh kriptanalis apabila dilakukan hanya dengan mengetahui berkas *ciphertext* saja. Namun, teknik ini dapat dipecahkan dengan cukup mudah apabila kriptanalis memiliki berkas *ciphertext* dan potongan berkas *plaintext*. Teknik kriptanalisis ini disebut *known-plaintext attack*.

**2.1 Dasar Teknik *Hill Cipher***

Dasar dari teknik *Hill Cipher* adalah aritmatika modulo terhadap matriks. Dalam penerapannya, *Hill Cipher* menggunakan teknik perkalian matriks dan teknik invers terhadap matriks.

Kunci pada *Hill Cipher* adalah matriks n x n dengan n merupakan ukuran blok. Matriks *K* yang menjadi kunci ini harus merupakan matriks yang *invertible*, yaitu memiliki inverse $K^{-1}$ sehingga :

$$K \,.\, K^{-1} = I \qquad (1)$$

Kunci harus memiliki invers karena matriks $K^{-1}$ tersebut adalah kunci yang digunakan untuk melakukan dekripsi.

**2.2 Teknik Enkripsi pada *Hill Cipher***

Proses enkripsi pada *Hill Cipher* dilakukan per blok *plaintext*. Ukuran blok tersebut sama dengan ukuran matriks kunci. Sebelum membagi teks menjadi deretan blok-blok, *plaintext* terlebih dahulu dikonversi menjadi angka, masing-masing sehingga A=1, B=2, hingga Y=25. Z diberi nilai 0.

Secara matematis, proses enkripsi pada *Hill Cipher* adalah:

$$C = K \,.\, P \qquad (2)$$

*C = Ciphertext*
*K* = Kunci
*P = Plaintext*

Jika terdapat *plaintext* P:

P = STRIKE NOW

Maka *plaintext* tersebut dikonversi menjadi:

P = 19 20 18 9 11 5 14 15 23
*Plaintext* tersebut akan dienkripsi dengan teknik *Hill Cipher*, dengan kunci K yang merupakan matriks 2x2.

$$K = \begin{bmatrix} 5 & 6 \\ 2 & 3 \end{bmatrix}$$

Karena matriks kunci K berukuran 2, maka *plaintext* dibagi menjadi blok yang masing-masing bloknya berukuran 2 karakter. Karena karakter terakhir tidak ada memiliki pasangan, maka diberi pasangan karakter yang sama yaitu W. P menjadi STRIKENOWW. Blok pertama dari *plaintext* P adalah :

$$P_{1,2} = \begin{bmatrix} 19 \\ 20 \end{bmatrix}$$

Blok *plaintext* ini kemudian dienkripsi dengan kunci K melalui persamaan (2).

$$C_{1,2} = \begin{bmatrix} 5 & 6 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 19 \\ 20 \end{bmatrix} = \begin{bmatrix} 215 \\ 98 \end{bmatrix}$$

Hasil perhitungan menghasilkan angka yang tidak berkorespondensi dengan huruf-huruf, maka lakukan modulo 26 pada hasil tersebut. Sehingga, $C_{1,2}$ menjadi:

$$C_{1,2} = \begin{bmatrix} 215 \\ 98 \end{bmatrix} = \begin{bmatrix} 7 \\ 20 \end{bmatrix} (\text{mod } 26)$$

Karakter yang berkorespondensi dengan 7 dan 20 adalah G dan T. maka S menjadi G dan T menjadi T. Setelah melakukan enkripsi semua blok pada *plaintext*
P maka dihasilkan *ciphertext* C sebagai berikut:

P = STRIKENOW
C = 7 20 14 11 7 11 4 21 19 11
C = GTNKGKDUSK

Dari *ciphertext* yang dihasilkan terlihat bahwa *Hill Cipher* menghasilkan *ciphertext* yang tidak memiliki pola yang mirip dengan *plaintext*nya.

**2.3. Teknik Dekripsi pada *Hill Cipher***

Proses dekripsi pada *Hill Cipher* pada dasarnya sama dengan proses enkripsinya. Namun matriks kunci harus dibalik (invers) terlebih dahulu. Secara matematis, proses dekripsi pada *Hill Cipher* dapat diturunkan dari persamaan (2).

$$C = K.P$$

$$K^{-1}.C = K^{-1}.K.P$$
$$K^{-1}.C = I.P$$
$$P = K^{-1}.C$$

Menjadi persamaan proses dekripsi:

$$P = K^{-1}.C \quad (3)$$

Dengan menggunakan kunci

$$K = \begin{bmatrix} 5 & 6 \\ 2 & 3 \end{bmatrix}$$

, maka proses dekripsi diawali dengan mencari invers dari matriks *K*. Mencari invers dapat dilakukan dengan menggunakan metode operasi baris (row operation) atau metode determinan [3].

Setelah melakukan perhitungan, didapat matriks $K^{-1}$ yang merupakan invers dari matriks *K*, yaitu :

$$K^{-1} = \begin{bmatrix} 27 & -54 \\ -18 & 45 \end{bmatrix} = \begin{bmatrix} 1 & 24 \\ 8 & 19 \end{bmatrix} (\text{mod } 26)$$

Kunci $K^{-1}$ yang digunakan untuk melakukan dekripsi ini telah memenuhi persamaan (1) karena:

$$K.K^{-1} = \begin{bmatrix} 53 & 234 \\ 26 & 105 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} (\text{mod } 26) = I$$

*Ciphertext* C = GTNKGKDUSK, akan didekripsi dengan menggunakan kunci dekripsi $K^{-1}$ dengan persamaan (3). Proses dekripsi ini dilakukan blok per blok seperti pada proses enkripsi. Pertama-tama ubah huruf-huruf pada *ciphertext* menjadi urutan numerik.

C = 7 20 14 11 7 11 4 21 19 11

Proses dekripsi dilakukan sebagai berikut:

$$P_{1,2} = K^{-1}.C_{1,2}$$

$$P_{1,2} = \begin{bmatrix} 1 & 24 \\ 8 & 19 \end{bmatrix} . \begin{bmatrix} 7 \\ 20 \end{bmatrix} = \begin{bmatrix} 487 \\ 436 \end{bmatrix} = \begin{bmatrix} 19 \\ 20 \end{bmatrix} (\text{mod } 26)$$

dan blok kedua:

$$P_{3,24} = K^{-1}.C_{3,4}$$

$$P_{3,4} = \begin{bmatrix} 1 & 24 \\ 8 & 19 \end{bmatrix} . \begin{bmatrix} 14 \\ 11 \end{bmatrix} = \begin{bmatrix} 278 \\ 321 \end{bmatrix} = \begin{bmatrix} 18 \\ 9 \end{bmatrix} (\text{mod } 26)$$

Setelah semua blok selesai didekripsi, maka didapatkan hasil *plaintext*:
P = 19 20 18 9 11 5 14 15 23
P = STRIKENOW

### 2.4 Kriptanalis pada *Hill Cipher*

Kriptanalisis terhadap *Hill Cipher* sangat sulit jika dilakukan dengan *ciphertext-only attack*, terlebih apabila matriks kunci yang digunakan berukuran besar. Kesulitan ini disebabkan oleh *ciphertext Hill Cipher* yang tidak memiliki pola dan setiap karakter dalam satu blok saling mempengaruhi karakter lainnya [2].

Teknik yang dapat digunakan untuk melakukan kriptanalisis terhadap *Hill Cipher* adalah *knownplaintext attack*. Jika kriptanalisis memiliki pecahan *plaintext* dan *ciphertext* yang saling berkorespondensi, maka *Hill Cipher* dapat dipecahkan. Namun proses yang cukup sulit adalah untuk menentukan panjang kunci yang digunAkan. Hal ini menjadi salah satu kekuatan yang dimiliki oleh *Hill Cipher*. Cara yang dapat dilakukan hanya dengan mencari tahu panjang kunci atau dengan melakukan perkiraan dan percobaan.

Kemungkinan terburuk yang dimiliki oleh *Hill Cipher* adalah ketika seorang kriptanalis memiliki potongan *plaintext* dan *ciphertext* yang berkorespondensi serta mengetahui panjang kunci yang digunakan. Dengan informasi ini, kriptanalis dapat memecahkan *Hill Cipher* dengan sangat mudah.

## 3 Algoritma Vigènere

Algoritma enkripsi ini dinamakan Vigènere Cipher atas nama Blaise de Vigènere, meskipun Giovan Batista Belaso telah menemukan algoritma enkripsi ini terlebih dahulu.

### 3.1 Metode Penerapan Algoritma Vigèner

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: sony
Plainteks: this plaintext
Kunci : sony sonysonys
```

Maka dapat dilihat pada tabel Vigènere bahwa untuk huruf pertama, T yang berkorespondensi

dengan huruf S pada kunci akan menghasilkan huruf L.

Hasil enkripsi seluruhnya adalah:

```
Plainteks : THIS PLAINTEXT
Kunci : sony sonysonys
Cipherteks : LVVQ HZNGFHRVL
```

Pada dasarnya, setiap enkripsi huruf adalah Caesar cipher dengan kunci yang berbeda-beda.
c('T') = ('T' + 's') mod 26 = L
c('H') = ('H' + 'o') ,mod 26 = V, dst

### 3.2 Metode Kasiski

Metode Kasiski membantu menemukan panjang kunci Vigènere Cipher. Metode Kasiski memanfaatkan keutungan bahasa inggris tidak hanya mengandung perulangan huruf, tetapi juga perulangan pasangan huruf atau tripel huruf seperti TH, THE, dsb. Perulangan huruf ini memungkinkan menghasilkan kriptogram yang berulang.

Contoh:

```
Plainteks:
CRYPTO IS SHORT FOR CRYPTOGRAPHY
Kunci :
abcdab cd abcda bcd abcdabcdabcd
Cipherteks:
CSASTP KV SIQUT GQU CSASTPIUAQJB
```

Pada contoh tersebut, CRYPTO dienkripsi menjadi kriptogram yang sama, yaitu CSATP. Hal ini dikarenakan jarak antara dua buah string yang berulang di dalam plainteks merupakan kelipatan dari panjang junci, maka string yang sama tersebut akan muncul menjadi kriptogram yangsama pula di dalam cipherteks. Tujuan dari metode Kasiski adalah mencari dua atau lebih kriptogram yang berulang untuk menentukan panjang kunci.

## 4. *Hill Cipher* Biner

Kemungkinan terburuk yang dimiliki oleh *Hill Cipher* adalah ketika seorang kriptanalis memiliki potongan *plaintext* dan *ciphertext* yang berkorespondensi serta mengetahui panjang kunci yang digunakan. Dengan informasi ini, kriptanalis dapat memecahkan *Hill Cipher* dengan sangat mudah.
Dengan memanfaatkan sifat bit dan modulo bilangan biner untuk menambah kerumitan dari *Hill Cipher*, maka hal tersebut bisa dihindari. Sifat bilangan biner dipakai di awal dan di akhir proses cipher text. Karakter diubah menjadi bilangan biner minimal 5 bit, lalu setelah itu diubah ke angkat dengan pengambilan jumlah bit yang berbeda.

### 4.1 Cara Enkripsi dan Dekripsi *Hill Cipher* Biner

Di bawah ini akan diberikan contoh penggunaan bilangan 5 bit yang diubah menjadi 4 bit.

1. Sebelum memasuki proses perkalian matriks, setiap karakter diubah menjadi bentuk biner 5 bit.
   Contoh:

| 10011 | 00001 | 10100 | 10101 | 10100 | 01001 | 00111 |
|---|---|---|---|---|---|---|
| s | a | t | u | t | i | g |

| 00001 | 01100 | 01001 | 01101 | 00001 | 00110 |
|---|---|---|---|---|---|
| a | l | i | m | a | f |

2. Setelah itu, string disusun dari bit-bit tersebut:
   100110000110100101011010001000011000010110001001011010001001 10
3. String tersebut kembali diubah ke dalam bentuk angka desimal, namun diambil dalam bentuk 4 bit.
   Contoh:
   1001 = 9, 1000 = 8, dst
4. Angka-angka itulah yang nantinya akan dipakai dalam perkalian matriks. Setelah perkalian matriks dilakukan, maka bit dikembalikan kepada bit awalnya ketika diubah menjadi chipertext.

Metode pengubahan bit biner ini juga dapat dipakai setelah melakukan perkalian matriks. Dekripsi dilakukan dengan melakukan proses sebaliknya.

### 4.2 Pengujian Metode-Metode Serangan terhadap *Hill Cipher* Biner

Seperti pada *Hill Cipher* biasa, jika kriptanalis hanya mengetahui chipertext saja, maka kriptanalis akan kesulitan untuk mengetahui plaintext dari pesan tersebut. Namun, berbeda dengan *Hill Cipher* biasa, jika kriptanalisis mengetahui chipertext dan potongan plaintext, dia tetap akan kesulitan untuk mengetahui plaintext dan kunci. Hal ini karena pengacakan yang dilakukan saat memakai perubahan bit biner membuat sangat sulit untuk menebak kunci dari cipher tersebut.
Satu-satunya cara yang dapat dilakukan adalah jika potongan plaintext tersebut dengan

mengetahui perpindahan jumlah bit dan mengetahui panjang kunci yang digunakan.

**5. Vigènere Biner**

Secara umum algoritma Vigènere Biner yang dirancang memiliki algoritma yang sama dengan algoritma Vigènere biasa. Yang membuatnya berbeda adalah bahwa pada algoritma yang dirancang ini, pesan terlebih dahulu diubah menjadi bentuk biner, kemudian proses enkripsi dilakukan terhadap pesan yang berbentuk biner.
Karena pesan yang akan dienkripsi berbentuk biner, maka tabel Vigènere yang akan digunakan juga berbeda dengan tabel Vigènere biasa. Tabel Vigènere yang digunakan dalam Vigènere Biner dapat dilihat pada Lampiran dari makalah ini.

**5.1 Algoritma Enkripsi Vigènere Biner**

Contoh:

```
Plainteks: satu tiga lima
Kunci:informatika
```

Langkah-langkahnya adalah

1. Buang spasi dari plainteks

```
satutigalima
p: satutigalima
k: informatikai
```

2. Tambahkan satu karakter pada bagian belakang pesan

```
satutigalimaf
p: satutigalimaf
k: informatikain
```

3. Ubah ke dalam bentuk biner dalam 5 bit

| 10011 | 00001 | 10100 | 10101 | 10100 | 01001 | 00111 |
|---|---|---|---|---|---|---|
| s | a | t | u | t | i | g |

| 00001 | 01100 | 01001 | 01101 | 00001 | 00110 |
|---|---|---|---|---|---|
| a | l | i | m | a | f |

4. Hilangkan bit 0 yang berada di depan

| 10011 | 1 | 10100 | 10101 | 10100 | 1001 |
|---|---|---|---|---|---|

| 111 | 1 | 1100 | 1001 | 1101 | 1 | 110 |
|---|---|---|---|---|---|---|

5. Sisipkan dengan 00000

| 10011 | 00000 | 1 | 00000 | 10100 | 00000 |
|---|---|---|---|---|---|

| 10101 | 00000 | 10100 | 00000 | 1001 |
|---|---|---|---|---|

| 00000 | 111 | 00000 | 1 | 00000 |
|---|---|---|---|---|

| 1100 | 00000 | 1001 | 00000 | 1101 |
|---|---|---|---|---|

| 00000 | 1 | 00000 | 110 | 00000 |
|---|---|---|---|---|

6. Rampatkan ke depan

| 10011 | 00000 | 10000 | 01010 | 00000 |
|---|---|---|---|---|

| 01010 | 10000 | 01010 | 00000 | 01001 |
|---|---|---|---|---|

| 00000 | 11100 | 00010 | 00001 | 10000 |
|---|---|---|---|---|

| 00010 | 01000 | 00110 | 10000 | 01000 | 00110 |
|---|---|---|---|---|---|

7. Ubah tiap karakter kunci ke dalam bentuk biner

| 01001 | 01110 | 00110 | 01111 | 10010 |
|---|---|---|---|---|

| 01101 | 00001 | 10100 | 01001 | 01011 |
|---|---|---|---|---|

| 00001 | 01001 | 01110 | 00110 | 01111 |
|---|---|---|---|---|

| 10010 | 01101 | 00001 | 10100 | 01001 | 01011 |
|---|---|---|---|---|---|

8. Cocokan dengan tabel

| 11100 | 01110 | 10110 | 11001 | 10010 |
|---|---|---|---|---|

| 11110 | 10001 | 11110 | 01001 | 10100 |
|---|---|---|---|---|

| 00001 | 00101 | 10000 | 00111 | 11111 |
|---|---|---|---|---|

| 10100 | 10101 | 00111 | 00100 | 10001 | 10001 |
|---|---|---|---|---|---|

9. Kembalikan ke bentuk alfabetik

```
p: satutigalimaf
k: informatikain
c: .NVYR”Q”ITAEPG-TUGDQQ
```

Dapat dilihat bahwa pada hasil dari enkripsi dengan menggunakan algoritma enkripsi Vigènere Biner, jumlah karakter pada cipherteks tidak sama dengan jumlah karakter pada plainteks. Jumlah karakter pada cipher teks

hampir dua kali dari jumlah karakter pada plainteks. Hal ini akan meninngkatkan keamanan pada cipherteks hasil enkripsi dengan menggunakan Vigènere Biner.

**5.2 Algoritma Dekripsi Vigènere Biner**

Contoh dekripsi:

`Cipherteks:.NVYR”Q”ITAEPG-TUGDQQ`
`Kunci : informatika`

1. Ubah bentuk cipherteks ke dalam bentuk biner.

| | | | | |
|---|---|---|---|---|
| 11100 | 01110 | 10110 | 11001 | 10010 |

| | | | | |
|---|---|---|---|---|
| 11110 | 10001 | 11110 | 01001 | 10100 |

| | | | | |
|---|---|---|---|---|
| 00001 | 00101 | 10000 | 00111 | 11111 |

| | | | | | |
|---|---|---|---|---|---|
| 10100 | 10101 | 00111 | 00100 | 10001 | 10001 |

2. Ubah kunci ke dalam bentuk biner.

| | | | | |
|---|---|---|---|---|
| 01001 | 01110 | 00110 | 01111 | 10010 |

| | | | | |
|---|---|---|---|---|
| 01101 | 00001 | 10100 | 01001 | 01011 |

| | | | | |
|---|---|---|---|---|
| 00001 | 01001 | 01110 | 00110 | 01111 |

| | | | | | |
|---|---|---|---|---|---|
| 10010 | 01101 | 00001 | 10100 | 01001 | 01011 |

3. Cocokan dengan tabel

| | | | | |
|---|---|---|---|---|
| 10011 | 00000 | 10000 | 01010 | 00000 |

| | | | | |
|---|---|---|---|---|
| 01010 | 10000 | 01010 | 00000 | 01001 |

| | | | | |
|---|---|---|---|---|
| 00000 | 11100 | 00010 | 00001 | 10000 |

| | | | | | |
|---|---|---|---|---|---|
| 00010 | 01000 | 00110 | 10000 | 01000 | 00110 |

4. Cari kelompok bit 0 yang terdiri dari lima bit. Pisahkan kelompok biner sebelum kelompok bit 0. Apabila ada kelompok bit 0 yang terdiri lebih dari 5 bit, maka ambil lima bit paling belakang, lalu pisahkan sebelum dengan kelompok bit sebelumnya.

| | | | | | |
|---|---|---|---|---|---|
| 10011 | 00000 | 1 | 00000 | 10100 | 00000 |

| | | | | |
|---|---|---|---|---|
| 10101 | 00000 | 10100 | 00000 | 1001 |

| | | | | |
|---|---|---|---|---|
| 00000 | 111 | 00000 | 1 | 00000 |

| | | | | |
|---|---|---|---|---|
| 1100 | 00000 | 1001 | 00000 | 1101 |

| | | | |
|---|---|---|---|
| 00000 | 1 | 00000 | 110 |

5. 5.Hilangkan kelompok biner yang terdiri dari 00000.

| | | |
|---|---|---|
| 10011 | 1 | 10100 |

| | | |
|---|---|---|
| 10101 | 10100 | 1001 |

| | |
|---|---|
| 111 | 1 |

| | | |
|---|---|---|
| 1100 | 1001 | 1101 |

| | |
|---|---|
| 1 | 110 |

6. Tambahkan bit 0 di depan masing-masing kelompok hingga masing-masing kelompok terdiri dari 5 bit

| | | |
|---|---|---|
| 10011 | 00001 | 10100 |

| | | |
|---|---|---|
| 10101 | 10100 | 01001 |

| | |
|---|---|
| 00111 | 00001 |

| | | |
|---|---|---|
| 01100 | 01001 | 01101 |

| | |
|---|---|
| 00001 | 00110 |

7. Ubah kembali ke dalam bentuk alfabet

| | | |
|---|---|---|
| s | a | t |

| | | |
|---|---|---|
| u | t | i |

| | |
|---|---|
| g | a |

| | | |
|---|---|---|
| l | i | m |

| | |
|---|---|
| a | f |

Plainteks: satutigalimaf

8. Buang satu huruf terakhir

`Plainteks: satutigalima`

**5.3 Pengujian Metode Kasiski Terhadap Vigènere Biner**

Metode Kasiski membantu menemukan panjang kunci dengan bergantung pada perulangan kata, bigram, trigram, maupun susunan huruf lainnya dalam pesan. Dengan menggunakan algoritma Vigènere Biner, kemungkinan untuk adanya perulangan menjadi kecil sekali.

Hal ini disebabkan karena satu huruf pada plainteks tidak diubah menjadi satu blok 14

kelompok bilangan biner, melainkan satu atau lebih blok bilangan biner. Dan pencocokan terhadap tabel juga bukan terhadap huruf, melainkan terhadap blok kelompok bilangan biner yang telah diolah sebelumnya. Sedangkan satu huruf pada kunci tetap menjadi satu blok kelompok bilangan biner. Hal ini menyebabkan perulangan dengan korespondensi huruf pada kunci menjadi kecil sekali.

Karena itu, dengan metode Kasiski pun sangat sulit untuk dapat menemukan panjang kunci yang digunakan untuk melakukan enkripsi.

Kekurangan yang dimiliki oleh algoritma Vigènere Biner adalah bertambah besarnya hasil enkripsi pesan menjadi hampir dua kali lipat. Hal ini akan menghabiskan memori komputer dalam penyimpanan cipherteks. [4]

**5. Kesimpulan**

*Hill Cipher* biasa kuat dalam menghadapi *ciphertext-only attack* namun lemah jika diserang dengan *knownplaintext attack*. Oleh karena itu, pengambangan *Hill Cipher* yang dimodifikasi dengan angka biner dapat memperumit cipher sehingga dapat mempersulit serangan knownplaintext attack.

Algoritma Vigènere biasa merupakan algoritma yang sangat kuat sebelum dipecahkan oleh Kasiski. Oleh karena itu, dengan menemukan sebuah algoritma berdasarkan pada Vigènere yang diolah sedemikian rupa hingga menyulitkan menemukan panjang kunci dengan menggunakan metode Kasiski akan membuat algoritma tersebut menjadi sebuah algoritma enkripsi yang kuat.

Algoritma Vigènere Biner merupakan perluasan dari algoritma Vigènere biasa. Dengan algoritma Vigènere Biner, pencarian panjang kunci dengan metode Kasiski menjadi sulit. Oleh karena itu dapat dikatakan algoritma enkripsi Vigènere Biner adalah algoritma enkripsi yang kuat.

Kekurangan yang dimiliki oleh algoritma Vigènere Biner adalah bertambah besarnya hasil enkripsi pesan menjadi hampir dua kali lipat. Hal ini akan menghabiskan memori komputer dalam penyimpanan cipherteks.

Namun, bila plainteks yang akan dienkripsi tidak terlalu panjang, maka bertambahnya besar cipherteks merupakan suatu keuntungan. Karena dengan bertambah besarnya hasil enkripsi, maka orang yang melihat cipherteks tidak akan mengetahui panjang sebenarnya dari plainteks yang dienkripsi.

Karena keuntungan tersebut, maka algoritma ini akan semakin kuat bila digunakan untuk melakukan enkripsi terhadap pesan yang tidak terlalu panjang.

Bila dibandingkan dengan *Hill Cipher* biner, maka *Hill Cipher* biner lebih efisien jika dipakai untuk pesan-pean yang berukuran sangat panjang karena dalam *Hill Cipher* biner tidak ada perubahan dalam jumlah memori yang dipakai.

**DAFTAR PUSTAKA**


[1] Munir, Rinaldi, Diktat Kuliah IF5054 Kriptografi, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2006.

[2] Forouzan, Behrouz, Cryptography and Network Security, McGraw-Hill, 2006.

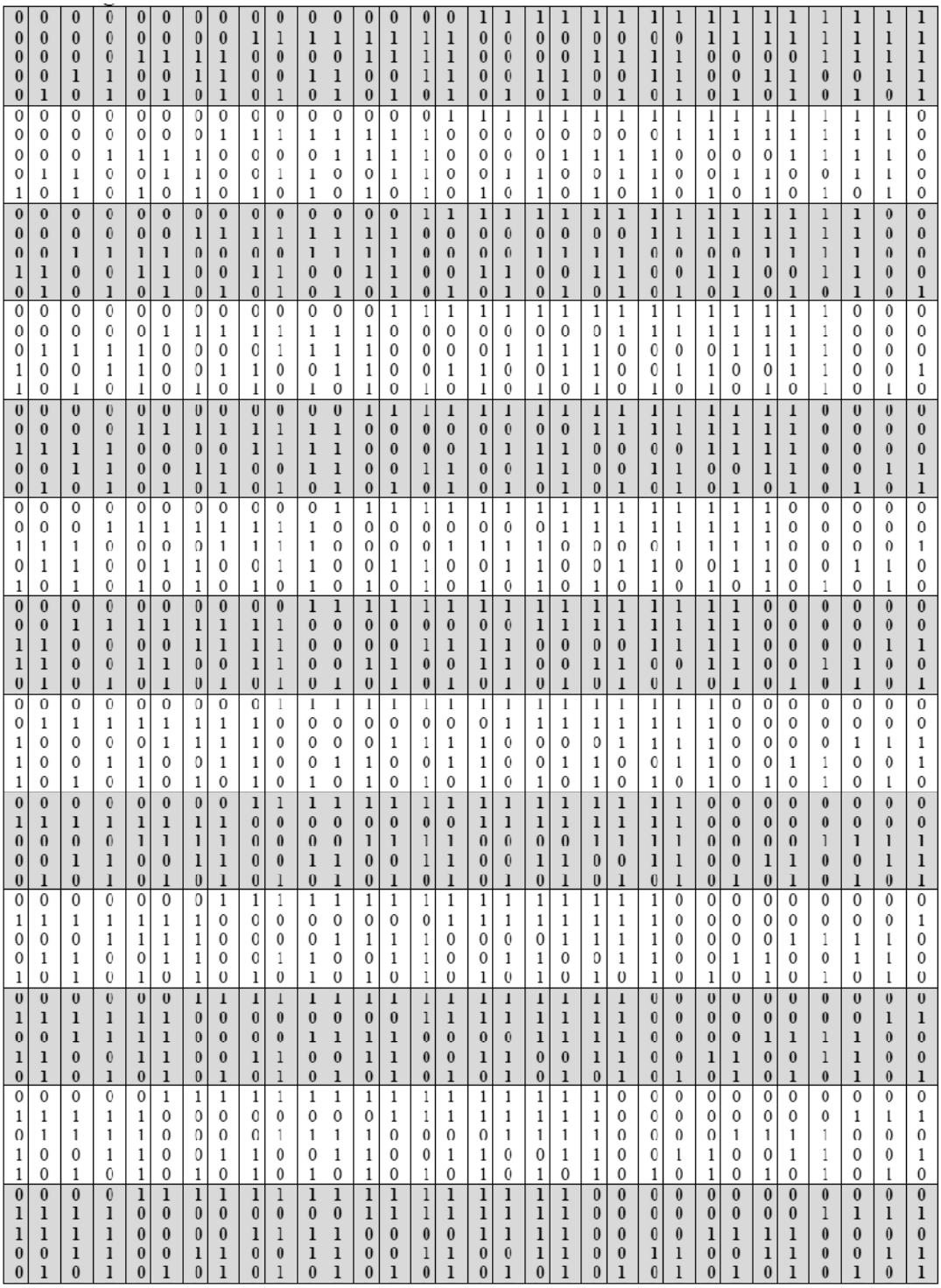
[3] H. Anton, C. Rorres, Elementary Linear Algebra, John Wiley & Sons, 2000

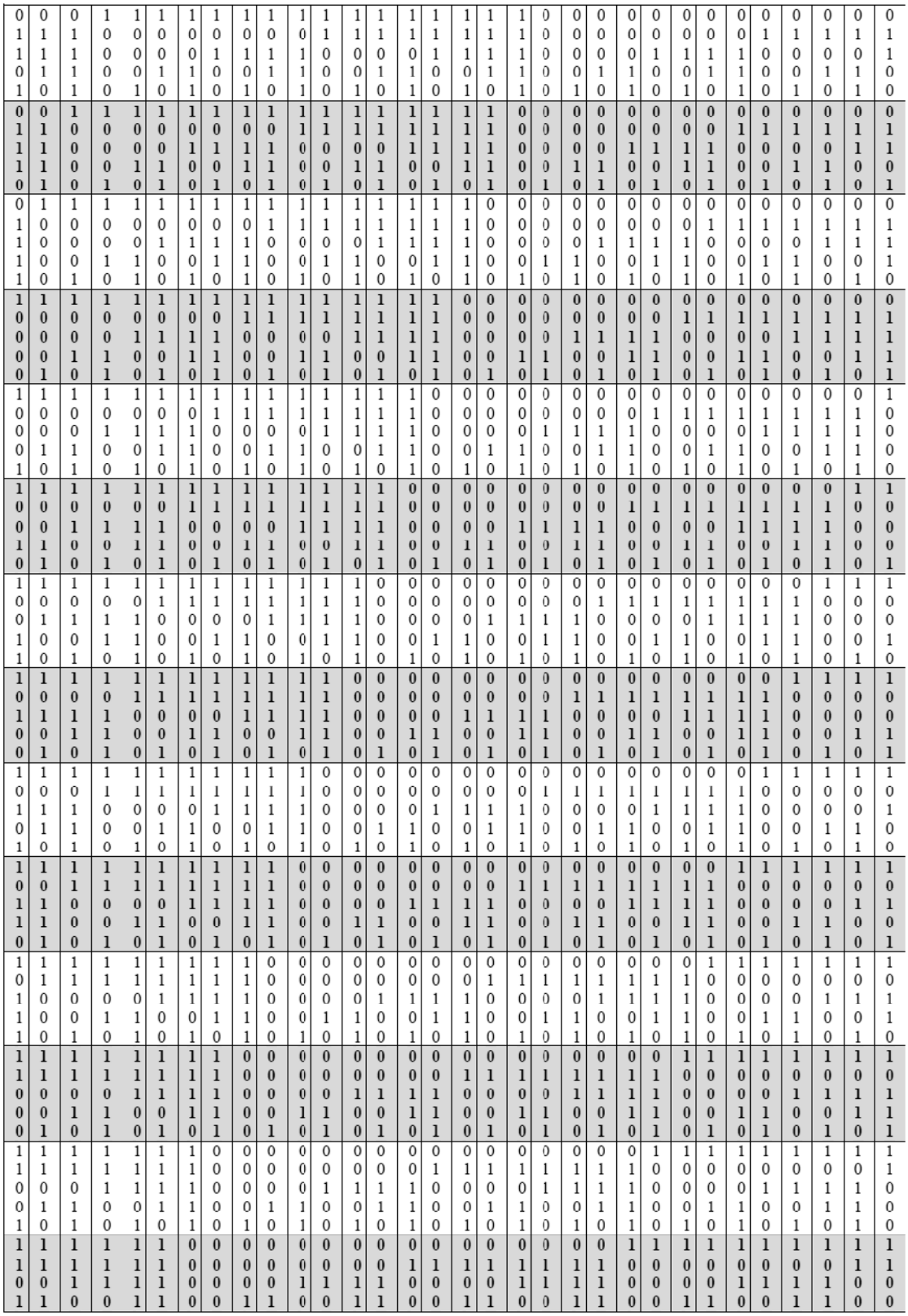
[4] Ridwan, Merancang Algoritma Kriptografi Kunci Simetri dengan Memperluas Algoritma Vigenere dan Analisis Metode Kasiski Terhadap Algoritma Tersebut, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2006.

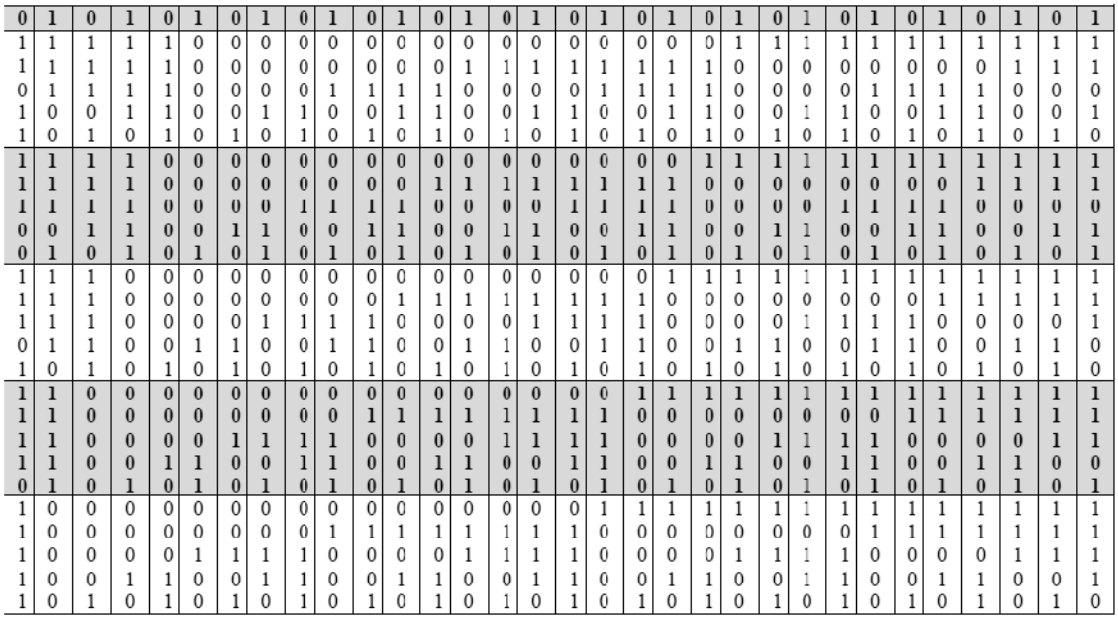
[5] Ivan Nugraha, Studi dan Analisis Mengenai Aplikasi Matriks dalam Kriptografi *Hill Cipher*. Kasiski Terhadap Algoritma Tersebut, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2007.

**LAMPIRAN**

**Tabel Vigenere Biner**







## Tabel Vigenere Chiper (biasa)

| | 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 |