# IMPLEMENTASI ALGORITMA VIGENERE DAN METODE LSBMR PADA CITRA DIAM

**Fauzus Sa'id[1], Wijanarto[2]**
[1,2]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Nakula I No. 5-11, Semarang, Jawa Tengah 50131 - (024) 3517261
E-mail : fasa.zepiert@gmail.com[1], wijanarto.udinus@gmail.com[2]

***Abstrak***

*Kemajuan teknologi komputer yang sangat bermanfaat pada kehidupan manusia sekarang adalah kecepatan dalam menyampaikan informasi dari tempat yang jauh yaitu melalui Internet. Dalam pengiriman informasi tersebut terdapat masalah yang mengganggu keamanan yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab yaitu dengan mengubah bahkan mengganti informasi data dalam sebuah media data citra yang disampaikan. Penulis menggunakan kriptografi dengan algoritma Vigenere untuk mengacak pesan dan steganografi dengan metode modifikasi Least Significant Bit Matching Revisited sebagai media yang akan menyembunyikan informasi berupa setiap nilai bit data pesan ke dalam nilai bit media citra. Setelah dianalisis dan diimplementasikan maka diperoleh bahwa citra yang digunakan untuk cover-image masih tampak seperti normal sehingga tidak menimbulkan kecurigaan bagi orang yang melihatnya, dari hasil pengujian 3 buah citra didapatkan hasil rata-rata PSNR sebesar 65,18773622225307dB pada citra yang belum mengalami serangan noise salt & paper. Kemudian jika diekstraksi dan didekripsi maka akan didapat kembali pesan asli yang telah dienkripsi dan disisipkan tersebut secara utuh. Dengan demikian, kriteria steganografi yang baik yaitu imperceptibility, fidelity dan recovery dapat terpenuhi.*

***Kata Kunci:*** *Citra Digital, Kriptografi, Steganografi, Vigenere, LSBMR (Least Significant Bit Matching Revisited).*

***Abstract***

*The advances of computer technology that very useful for human life now is the speed in conveying information from distant places namely Internet. The transmission of information contained several problems that can be distrubed on security committed by parties who are not responsible to change even change the data information in a media image data is delivered. The author uses cryptography with Vigenere algorithm to scramble the message and steganography by a modified method Least Significant Bit Matching Revisited as a medium that will hide information in the form of any values of bits of data messages into bit values image media. Having analyzed and implemented the obtained that the image used for the cover-image still looked like normal so as not to arouse suspicion to the viewer, from the test results obtained three pieces of the image of the average yield for 65,18773622225307dB PSNR in the image that has not experienced an attack noise salt & paper. Then if extracted and described it will get back the original message that has been encrypted and is inserted as a whole. Thus, the criteria for good steganography is imperceptibility, fidelity and recovery can be fulfilment.*

***Keywords:*** *Digital Image, Cryptography, Steganography, Vigenere, LSBMR (Least Significant Bit Matching Revisited)*

## 1. PENDAHULUAN

Komunikasi sudah digunakan oleh manusia sejak dulu untuk bertukar informasi ataupun bersilaturahmi. Perkembangan jaringan internet yang memungkinkan setiap orang untuk saling bertukar data atau informasi melalui jaringan internet tersebut[1] [2].

Pertukaran informasi digital menggunakan internet sangat riskan mengalami pencurian yang kemudian diubah sedemikian rupa lalu disebarkan kembali. Dengan semakin banyak serangan yang mungkin terjadi dalam proses pertukaran data menyebabkan perlunya suatu metode agar dapat meningkatkan keamanan informasi [3].

Kriptografi adalah ilmu yang mempelajari cara untuk menjaga keamanan data agar tetap aman saat dikirimkan, tanpa mengalami gangguan dari pihak ketiga. *Vigenere Chiper* termasuk dalam cipher abjad majemuk (*polyalphabetic substitution Chiper*) yang dipublikasikan oleh diplomat sekaligus seorang kriptologis Perancis, Blaise de Vigenere pada abad 16 (tahun 1586). *Vigenere Chiper* adalah metode menyandikan teks *alfabet* dengan menggunakan deretan sandi *Caesar* berdasarkan huruf-huruf pada kata kunci [8].

Steganografi adalah suatu metode penyembunyian informasi pada suatu media, dapat berupa image, audio, ataupun video. Metode ini dibuat sedemikian rupa sehingga selain pengirim dan penerima, keberadaan informasi tidak diketahui. Aspek utama dari metode ini adalah seberapa tinggi tingkat keamanannya agar pihak lain kesulitan dalam mendeteksi keberadaan informasi yang disembunyikan [1].

Banyak metode yang berhubungan dengan steganografi sampai sekarang. Terutama metode *Least Significant Bit* (LSB)[5][6] adalah salah satu metode yang sederhana dan efektif dalam implementasi konsep steganografi. Dalam skema ini, metode LSB hanya mengganti bidang gambar *cover* dengan aliran bit rahasia. Hal ini sangat mudah untuk mendeteksi keberadaan pesan tersembunyi bahkan pada tingkat *embedding* rendah menggunakan algoritma *steganalytic*. *Least Significant Bit Matching* (LSBM) adalah modifikasi kecil untuk pengganti LSB. Pada LSBM, setiap bit-bit data yang disembunyikan dibandingkan dengan bit terakhir dari bit cover image yang berkoresponden. Jika cocok jangan lakukan apapun, jika tidak cocok, byte pada nilai piksel cover image ditambah satu atau dikurang satu secara acak (kecuali untuk byte yang ukurannya 0 tidak dapat dikurangi dan byte yang ukurannya 255 tidak dapat ditambah). Dalam *Least Significant Bit Matching Revisited* (LSBMR) menggunakan sepasang piksel sebagai *unit embedding*, di mana LSB dari piksel pertama membawa satu bit pesan rahasia dan hubungan dari nilai-nilai dua piksel membawa sedikit dari pesan rahasia lain [4] [7].

Penggunaan citra digital sebagai media penampung pada proses steganografi memiliki kapasitas yang terbatas, tidak seperti pada penggunaan stream media sebagai media penampung. Untuk mencegah deteksi ataupun serangan, kerapatan penyisipan informasi pada media penampung harus merata [1]. Oleh karena itu, penulis ingin melakukan penelitian mengenai penyisipan teks yang sudah di enkripsi dengan kriptografi algoritma *Vigenere Cipher* ke dalam sebuah citra dengan metode penyisipan *Least Significant Bit*

*Matching Revisited* atau yang sering disingkat LSBMR.

## 2. METODE

### 2.1 Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: "*cryptos*" artinya "*secret*" (rahasia), sedangkan "*graphein*" artinya "*writing*" (tulisan), sehingga kriptografi berarti "*secret writing*" (tulisan rahasia). Kriptografi didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Kriptografi memiliki dua konsep utama, yaitu enkripsi (*encryption*) dan dekripsi (*decryption*). Enkripsi adalah proses penyandian *plainteks* menjadi *cipherteks*, sedangkan dekripsi adalah proses mengembalikan *cipherteks* menjadi *plainteks* semula. Enkripsi dan dekripsi membutuhkan kunci sebagai parameter yang digunakan untuk transformasi [8].

### 2.2 Steganografi

Steganografi adalah suatu teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya. Media yang digunakan umumnya merupakan suatu media yang berbeda dengan media pembawa informasi rahasia, dimana disinilah fungsi dari teknik steganografi yaitu sebagai teknik penyamaran menggunakan media lain yang berbeda sehingga informasi rahasia dalam media awal tidak terlihat secara jelas [3].

Kata steganografi berasal dari bahasa Yunani, yaitu dari kata *Stegos* (*covere*d / tersembunyi) dan *Graptos* (*writing* / tulisan). Steganografi di dunia modern biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam suatu arsip citra digital, audio, atau video. Teknik Steganografi ini telah banyak digunakan dalam strategi peperangan dan pengiriman sandi rahasia sejak jaman dahulu kala. Dalam perang Dunia II, teknik Steganografi umum digunakan oleh tentara Jerman dalam mengirimkan pesan rahasia dari atau menuju Jerman [3][11].

### 2.3 Algoritma Kriptografi Vigenere Chiper

Vigenere cipher termasuk dalam *cipher* abjad majemuk (*polyalphabetic substitution cipher*) yang dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Vigenere pada abad 16 (tahun 1586). *Vigenere Cipher* adalah metode menyandikan teks *alfabet* dengan menggunakan deretan sandi *Caesar* berdasarkan huruf-huruf pada kata kunci. *Vigenere Cipher* menggunakan tabel seperti pada tabel 1, *Vigenere Cipher* dengan angka dalam melakukan enkripsi.

Teknik dari substitusi *vigenere cipher* bisa dilakukan dengan dua cara [9][10]:

a. *Vigenere Cipher* dengan Angka

**Tabel 1:** Vigenere cipher dengan angka

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Jika ditukar dengan angka, maka kunci dengan huruf "FASA"
K = (5, 0, 18, 0)
Dan plainteks nya " VIGENERE " akan menjadi
P = (21,9,6,4,13,4,17,4).

Dari contoh tabel, maka dapat disimpulkan bahwa rumus dari enkripsi dan dekripsi data *vigenere cipher* adalah [10]:
Enkripsi :

ci = (pi+ ki) mod 26
Deskripsi :
pi = (ci – ki ) mod 26 ; untuk ci > = ki

pi = (ci + 26 – ki ) mod 26 ; untuk ci < = ki
Keterangan :
c = *Ciphertex*t p = *Plaintext*
k = Kunci

b. *Vigenere Cipher* dengan Huruf
*Vigenere Cipher* dengan huruf berisi *alfabet* yang dituliskan dalam 26 baris, masing masing baris digeser ke kiri dari baris sebelumnya membentuk ke-26 kemungkinan sandi *Caesar* setiap huruf disediakan dengan menggunakan baris yang berbeda-beda sesuai kunci yang diulang [10].

**Tabel 2:** Vigenere Cipher dengan Huruf

| | 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 |

## 2.4 Least Significant Bit

Metode LSB merupakan metode steganografi yang paling sederhana dan mudah diimplementasikan. Metode ini menggunakan citra digital sebagai *covertext*. Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), ada bit yang paling berarti (*most significant bit* atau MSB) dan bit yang paling kurang berarti (*least significant bit* atau LSB). Sebagai contoh byte 11010010, angka bit 1 (pertama, digaris-bawahi) adalah bit MSB, dan angka bit 0 (terakhir, digaris-bawahi) adalah bit LSB. Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai byte satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan bit tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti. Mata manusia tidak dapat membedakan perubahan kecil tersebut [8].

Dalam skema ini, metode LSB hanya mengganti bidang gambar *cover* dengan aliran bit rahasia. Hal ini dilakukan dengan membandingkan setiap bit pesan dengan LSB dari masing-masing gambar piksel. Rata-rata, LSB membutuhkan hanya setengah bit dalam suatu perubahan gambar. Hal ini sangat mudah untuk mendeteksi keberadaan pesan tersembunyi bahkan pada tingkat *embedding* rendah menggunakan algoritma *steganalytic. Least Significant Bit Matching* (LSBM) adalah modifikasi kecil untuk pengganti LSB. Pada LSBM, setiap bit-bit data yang disembunyikan dibandingkan dengan bit terakhir dari bit cover image yang berkoresponden. Jika cocok jangan lakukan apapun, jika tidak cocok, bit pada *cover image* ditambah satu atau dikurang satu secara acak (kecuali untuk byte yang ukurannya 0 tidak dapat dikurangi dan byte yang ukurannya 255 tidak dapat ditambah). Dalam *Least Significant Bit Matching Revisited* (LSBMR) menggunakan sepasang piksel sebagai *unit embedding*, di mana LSB dari piksel pertama membawa satu bit pesan rahasia dan hubungan dari nilai-nilai dua piksel membawa sedikit dari pesan rahasia lain [4][7].

## 3. HASIL DAN PEMBAHASAN

### 3.1 Enkripsi Pesan

Dalam proses yang pertama ini enkripsi menggunakan algoritma *vigenere cipher* dengan persamaan ci= (pi + ki) mod 128.

**Tabel 3:** Algoritma Enkripsi

| | |
|---|---|
| Input | Data *.txt yang akan dienkripsi |
| Output | Data *.txt yang telah dienkripsi |
| Proses | Panjangtekslength(p) // memeriksa panjang plainteks<br>For 1 to panjangText // menyamakan panjang k dengan panjang p<br>Huruf[i] M[i]<br>nilaiASCII[i]<br>For 1 to panjangText // menyamakan panjang k dengan panjang p<br>i mod 128 =1?<br>c = (p + k) mod128<br>Cetak char (c)<br>c = Cipherteks |

Berikut adalah proses enkripsi *vigenere cipher*:

a. Pesan yang dijadikan p di baca panjang elementnya.
b. Kemudian kunci dibaca panjang elementnya k dan di looping sesuai panjang element p.
c. Setelah panjang element k sama dengan p, maka k di ubah ke bilangan ascii lalu di susun menjadi seperti p.
d. Lalu p dan k di oprasikan dengan persamaan enkripsi *vigenere cipher*. Pada tabel 4 dan tabel 5 dibawah ini adalah p dan k pada f (1,1) sampai f (1,8).

**Tabel 4:** Karakter p, k, dan c pada f (1,1) sampai f (1,8)

| | *f*(1,1) | *f*(1,2) | *f*(1,3) | *f*(1,4) | *f*(1,5) | *f*(1,6) | *f*(1,7) | *f*(1,8) |
|---|---|---|---|---|---|---|---|---|
| *p* | v | i | g | e | n | e | r | e |
| *k* | f | a | s | a | f | a | s | a |

**Tabel 5:** Nilai ASCII p, k, dan c pada f (1,1) sampai f (1,8)

| | *f*(1,1) | *f*(1,2) | *f*(1,3) | *f*(1,4) | *f*(1,5) | *f*(1,6) | *f*(1,7) | *f*(1,8) |
|---|---|---|---|---|---|---|---|---|
| *p* | 118 | 105 | 103 | 101 | 110 | 101 | 104 | 101 |
| *k* | 102 | 97 | 115 | 97 | 102 | 97 | 115 | 97 |

Dari p dan k diatas maka c dapat diketahui dengan pengoprasian sebagai berikut :

c (1,1) = (118 + 102) mod 128 = 92
c (1,2) = (105 + 97) mod 128 = 74
c (1,3) = (103 + 115) mod 128 = 90
c (1,4) = (101 + 97) mod 128 = 70
c (1,5) = (110 + 102) mod 128 = 84
c (1,6) = (101 + 97) mod 128 = 70
c (1,7) = (104 + 115) mod 128 = 101
c (1,8) = (101 + 97) mod 128 = 70

Hasil dari operasi dengan persamaan *vigenere cipher* dalam bentuk kode *ascii* akan dikembalikan lagi dalam bentuk karakter adalah sebagai berikut :

**Tabel 6:** Hasil Operasi dengan Persamaan Vigenere Cipher

| | *f*(1,1) | *f*(1,2) | *f*(1,3) | *f*(1,4) | *f*(1,5) | *f*(1,6) | *f*(1,7) | *f*(1,8) |
|---|---|---|---|---|---|---|---|---|
| *c* | 92 | 74 | 90 | 70 | 84 | 70 | 101 | 70 |
| | \ | J | Z | F | T | F | e | F |

### 3.2 Merahasiakan Pesan

Untuk dapat lebih memahami proses merahasiakan pesan dari metode yang diusulkan di penelitian ini, maka akan diberikan potongan dari piksel citra sebagai ilustrasi dari proses merahasiakan pesan dengan menggunakan algoritma *LSB Matching Revisited*. Algoritma *LSB Matching Revisited* memodifikasi LSB pada pasangan piksel tertentu dan menerapkan 4 aturan penyisipan. Aturan dalam penyisipan pesan menggunakan algoritma *LSB Matching Revisited* memiliki persamaan sebagai berikut :

a. Jika mi sama dengan xi dan jika mi+1 tidak sama dengan f([xi /2]+xi+1) maka nilai dari piksel yi+1 sama dengan xi+1 ditambah 1 atau dikurang 1 secara acak dengan ketentuan jika 255 tidak bisa ditambah 1 dan jika 0 tidak

dapat dikurangi 1, dan nilai piksel yi sama dengan nilai xi.

b. Jika mi sama dengan xi dan jika mi+1 sama dengan f([xi /2]+xi+1) maka nilai dari piksel yi+1 sama dengan xi+1, dan nilai piksel yi sama dengan nilai xi.
c. Jika mi tidak sama dengan xi dan mi+1 sama dengan xi+1 maka nilai dari piksel yi sama dengan nilai piksel xi dikurangi 1 dan nilai piksel yi+1 sama dengan nilai dari xi+1.
d. Jika mi sama dengan xi dan mi+1 tidak sama dengan xi+1 maka nilai dari piksel yi sama dengan nilai piksel xi ditambah 1 dan nilai piksel yi+1 sama dengan nilai dari xi+1.

Berikut contoh penyisipan menggunakan algoritma *LSB Matching Revisited* :

| $x_i$ | $x_{i+1}$ | $x_i$ | $x_{i+1}$ | $m_i$ | $m_{i+1}$ |
|---|---|---|---|---|---|
| 162 | 150 | 10100010 | 10010110 | 0 | 0 |

| $y_i$ | $y_{i+1}$ | $y_i$ | $y_{i+1}$ |
|---|---|---|---|
| 162 | 151 | 10100010 | 10010111 |

| $x_i$ | $x_{i+1}$ | $x_i$ | $x_{i+1}$ | $m_i$ | $m_{i+1}$ |
|---|---|---|---|---|---|
| 162 | 150 | 10100010 | 10010110 | 0 | 1 |

| $y_i$ | $y_{i+1}$ | $y_i$ | $y_{i+1}$ |
|---|---|---|---|
| 162 | 150 | 10100010 | 10010110 |

| $x_i$ | $x_{i+1}$ | $x_i$ | $x_{i+1}$ | $m_i$ | $m_{i+1}$ |
|---|---|---|---|---|---|
| 162 | 150 | 10100010 | 10010110 | 1 | 0 |

| $y_i$ | $y_{i+1}$ | $y_i$ | $y_{i+1}$ |
|---|---|---|---|
| 161 | 150 | 10100001 | 10010110 |

| $x_i$ | $x_{i+1}$ | $x_i$ | $x_{i+1}$ | $m_i$ | $m_{i+1}$ |
|---|---|---|---|---|---|
| 162 | 150 | 10100010 | 10010110 | 1 | 1 |

| $y_i$ | $y_{i+1}$ | $y_i$ | $y_{i+1}$ |
|---|---|---|---|
| 163 | 150 | 10100011 | 10010110 |

**Gambar 1.** Proses Penyisipan

### 3.3 Pengambilan Pesan

Setelah melakukan proses merahasiakan pesan, maka akan diuji apakah pesan dapat diambil kembali. Untuk lebih memahami proses pengambilan pesan pada metode yang diusulkan di penelitian ini, maka akan diberikan potongan dari piksel citra sebagai ilustrasi dari proses pengambilan pesan dengan menggunakan algoritma *LSB Matching Revisited*. Aturan dalam pengambilan pesan menggunakan algoritma *LSB Matching Revisited* memilki persamaan sebagai berikut :

a. mi sama dengan yi.
b. mi+1 sama dengan f([0,5.yi] + yi+1).

| 162 | 150 |
|---|---|
| $x_i$ | $x_{i+1}$ |

| $y_i$ | $y_{i+1}$ | LSB($y_i$) | f ($y_i$, $y_{i+1}$) |
|---|---|---|---|
| 162 | 151 | 0 | 0 |
| 162 | 150 | 0 | 1 |
| 161 | 150 | 1 | 0 |
| 163 | 150 | 1 | 1 |

**Gambar 2.** Proses Pengambilan

### 3.4 Dekripsi Pesan

Dalam proses deskripsi yang ini dengan algoritma *vigenere cipher* yang menggunakan persamaan pi = (ci – ki) mod 128 atau pi = ((ci + 128) – ki) mod 128.

**Tabel 7:** Algoritma untuk Melakukan Proses Deskripsi

| | |
|---|---|
| Input | Data *.txt yang akan dideskripsi |
| Output | Data *.txt yang telah dideskripsi |
| Proses | Panjangtekslength(M)<br>For 1 to panjangText // menyamakan panjang k dengan panjang c<br>Huruf[i]M[i]<br>nilaiASCII[i]<br>For 1 to panjangText // menyamakan panjang k dengan panjang c<br>i mod 128 =1?<br>p = (c - k) mod 128<br>Cetak chr (c)<br>Pi= Plainteks |

Berikut adalah proses dekripsi *vigenere cipher*:

a. Disini k = "fasa" dibaca panjang elementnya dan di looping sesuai panjang element c.
b. Setelah panjang element k sama dengan c, maka k di ubah ke kode ascii lalu di susun menjadi seperti c.
c. Lalu c dan k di oprasikan dengan persamaan dekripsi *vigenere cipher*. Pada tabel 8 dan tabel 9 dibawah ini adalah c dan k *vigenere cipher* pada f (1,1) sampai f (1,8).

**Tabel 8:** Karakter p, k, dan c pada f (1,1) sampai f (1,8)

| | f(1,1) | f(1,2) | f(1,3) | f(1,4) | f(1,5) | f(1,6) | f(1,7) | f(1,8) |
|---|---|---|---|---|---|---|---|---|
| c | \ | J | Z | F | T | F | e | F |
| k | f | a | s | a | f | a | s | a |

**Tabel 9:** Nilai ASCII p, k, dan c pada f (1,1) sampai f (1,8)

| | f(1,1) | f(1,2) | f(1,3) | f(1,4) | f(1,5) | f(1,6) | f(1,7) | f(1,8) |
|---|---|---|---|---|---|---|---|---|
| c | 92 | 74 | 90 | 70 | 84 | 70 | 101 | 70 |
| k | 102 | 97 | 115 | 97 | 102 | 97 | 115 | 97 |

Berikut ilustrasi penghitungan dekripsi, misalkan kita ingin mengembalikan pesan rahasia dengan kunci ke dalam pesan asli :

pi = (ci – ki) mod 128 ; untuk Ci > = Ki
pi = ((ci + 128) – ki) mod 128 ; untuk Ci < = Ki
pi = ((92 + 128) - 102) mod 128 = 118
pi = ((74 + 128) - 97) mod 128 = 105
pi = ((90 + 128) - 115) mod 128 = 103
pi = ((70 + 128) - 97) mod 128 = 101
pi = ((84 + 128) - 102) mod 128 = 110
pi = ((70 + 128) - 97) mod 128 = 101
pi = ((101 + 128) - 115) mod 128 = 104
pi = ((70 + 128) - 97) mod 128 = 101

Pada tabel 9 di bawah ini menunjukkan hasil dari operasi dengan persamaan *vigenere cipher* dalam bentuk kode *ascii* akan dikembalikan lagi dalam bentuk karakter.

**Tabel 10:** Hasil Operasi dengan Persamaan Vigenere Cipher

| | f(1,1) | f(1,2) | f(1,3) | f(1,4) | f(1,5) | f(1,6) | f(1,7) | f(1,8) |
|---|---|---|---|---|---|---|---|---|
| p | 118 | 105 | 103 | 101 | 110 | 101 | 104 | 101 |
| | v | i | g | e | n | e | r | e |

### 3.5 Hasil Pengujian

Pengujian non-attack merupakan pengujian dengan menghitung nilai MSE dan PSNR pada *stego-image* agar dapat mengetahui kualitas *stego-image* bila dibandingkan dengan *cover-image* aslinya. Sedangkan pengujian attack bertujuan untuk mengetahui dampak serangan (*attack*) terhadap pesan yang terdapat dalam *stego-image*. Teknik attack yang digunakan adalah *salt & pepper*. *Stego-image* yang sudah diserang akan dicoba untuk diambil kembali pesan di dalamnya dan dilakukan penghitungan MSE dan PSNR pada *stego-image* agar dapat diketahui kualitasnya. Semakin rendah nilai MSE dan semakin tinggi nilai PSNR berarti semakin bagus kualitas citra. Berikut adalah nilai MSE dan PSNR dari hasil pengujian metode dengan pengujian *non-attack* dan pengujian *attack* dengan memberikan *noise salt & paper pada stego-image* :

**Tabel 11:** Pengujian pada Citra non-Attack

| Citra yang diuji | Ukuran Citra | MSE (db) | PSNR (db) | Hasil Stego |
|---|---|---|---|---|
| imagee1.bmp | 256x256 | 0,011215209960938 | 67,666725221632035 |  |
| imagee2.tif | 256x256 | 0,002090454101563 | 74,962392940909922 |  |
| imagee3.jpg | 256x256 | 0,333480834960938 | 52,934090504217266 |  |

**Tabel 12:** Pengujian pada Citra Attack dengan Salt and Pepper

| Format Citra yang diuji | Ukuran Citra | MSE (db) | PSNR (db) | Hasil Stego |
|---|---|---|---|---|
| salt&pepper-stegimagee1.bmp | 256x256 | 2.028326416015625e+02 | 25.093420838740890 |  |
| salt&pepper-stegimagee2.tif | 256x256 | 2.539497528076172e+02 | 24.117321361971808 |  |
| salt&pepper-stegimagee3.jpg | 256x256 | 2.491920776367188e+02 | 24.199456995922205 |  |

## 4. KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh peneliti, maka dapat disimpulkan beberapa hal sebagai berikut:

a. Penggabungan teknik kriptografi dan steganografi menggunakan algoritma *Vigenere Cipher* dan *LSB Matching Revisited* dapat digunakan untuk merahasiakan pesan ke dalam citra dengan baik dan kualitas *stego-image non-attack* yang dihasilkan cukup tinggi dengan nilai rata-rata PSNR 65,18773622225307dB.
b. Penggabungan teknik kriptografi dan steganografi dapat membantu menjaga kerahasiaan pesan karena orang yang tidak mengetahui kunci rahasia yang digunakan akan kesulitan untuk mendapatkan pesan yang terdapat pada *stego-image*. Walaupun orang tersebut dapat mengambil pesan pada *stego-image*, namun pesan tersebut masih berupa *ciphertext* sehingga perlu mencari kunci untuk mendekripsikannya.
c. Metode yang diusulkan pada penelitian ini cukup bagus, karena pesan asli berhasil di ambil kembali dari *stego-image* tanpa ada perubahan yang signifikan pada teks pesan asli ketika didekripsi walaupun telah dilakukan pengujian *attack* dengan *noise seperti salt & pepper*. Sementara pada pengujian *non-attack*, nilai MSE dan PSNR *stego-image* lebih baik daripada pengujian *attack dengan noise salt & pepper.*

### 4.2 Saran

Dalam penelitian ini terbukti bahwa teknik kriptografi dan steganografi dapat digabungkan dengan baik. Dengan menggunakan algoritma kriptografi *Vigenere Cipher* dan algoritma steganografi *LSB Matching Revisited* kualitas *stego-image* yang dihasilkan cukup tinggi. Untuk penelitian selanjutnya, peneliti memberikan saran sebagai berikut:

a. Penelitian dapat dilanjutkan dengan menggunakan citra RGB sebagai *cover-image* sehingga memiliki kapasitas lebih banyak yang dapat digunakan untuk menyisipkan pesan.
*b.* Menggunakan lebih banyak format file, *.docx, *.pdf, *.xml, dan sebagainya untuk pesan yang akan dienkripsi, disisipkan, diekstraksi, dan didekripsi pada citra *cover-image.*
c. Mengembangkan metode yang diusulkan pada penelitian ini agar dapat lebih tahan terhadap attack serta mencoba menerapkan jenis attack lainnya, seperti kompresi citra dan sebagainya dan mengganti algoritma kriptografi *vigenere* dan algoritma steganografi *LSB Matching Revisited* dengan algoritma steganografi lainnya.

## DAFTAR PUSTAKA

[1] Ratna Astuti Nugrahaeni, R. Rumani M., Ir., Drs., MSEE, Surya Michrandi Nasution, ST., MT, "Implementasi Kriptografi dan Steganografi untuk Teks pada Media Citra Digital dengan Algoritma AES dan F5", 2012.

[2] Jhoni Verlando Purba, Marihat Situmorang, Dedy Arisandi, "Implementasi Steganografi Pesan

Text Ke Dalam File Sound (.Wav) Dengan Modifikasi Jarak Byte Pada Algoritma Least Significant Bit (Lsb)", Jurnal Dunia Teknologi Informasi Vol. 1, No. 1, (2012) 50-55.

[3] Ari Septayuda Dr., Ir. Bambang Hidayat, DEA Hilal Hudan Nuha,MT, "ANALISIS STEGANOGRAFI CITRA DIGITAL MENGGUNAKAN METODE SPREAD SPECTRUM BERBASIS ANDROID", Bandung, 2013.

[4] B. Sharmila1 and R. Shanthakumari, "EFFICIENT ADAPTIVE STEGANOGRAPHY FOR COLOR IMAGES BASED ON LSBMR ALGORITHM", Ictact Journal on Image and Video Processing, Vol. 02, ISSUE: 03, 2012.

[5] Sarita Poonia, Mamtesh Nokhwal, Ajay Shankar, "A Secure Image Based Steganography and Cryptography with Watermarking", International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-8, 2013.

[6] Mielikainen J. "LSB matching revisited", IEEE signal Processing, Vol.13, No. 5, pp.285-287, 2006.

[7] Shauma Hayyu Syakura, "STUDI ANALISIS PERBANDINGAN METODE STEGANALISIS TERHADAP LSBIMAGE STEGANOGRAPHY", Bandung, 2010.

[8] Basuki Rakhmat dan Muhammad Fairuzabadi, M.Kom., "STEGANOGRAFI MENGGUNAKAN METODE LEAST SIGNIFICANT BIT DENGAN KOMBINASI ALGORITMA KRIPTOGRAFI VIGENÈRE DAN RC4", Jurnal Dinamika Informatika, vol. 5, no. 2, 2010.

[9] M. Wahid, "Steganografi Citra Digital Dengan Discrete Wavelet Transform (DWT) dan Discrete Cosine Transform (DCT)," 2014.

[10] Putu H. Arjana, Tri Puji Rahayu, Yakub, Hariyanto, "IMPLEMENTASI ENKRIPSI DATA DENGAN ALGORITMA VIGENERE CHIPER" Seminar Nasional Teknologi Informasi dan Komunikasi, ISSN: 2089-9815, Yogyakarta, 10 Maret 2012.

[11] D. K. Budiarsyah, "Pengujian Beberapa Teknik Proteksi Watermark Terhadap Penyerangan," 2013.