JURNAL MANTIK PENUSA

Volume 1 No 2 Desember 2017

p-ISSN 2088-3943
e-ISSN 2580-9741

IMPLEMENTASI PENYEMBUNYIAN PESAN PADA CITRA DIGITAL DENGAN MENGGABUNGKAN ALGORITMA *HILL CIPHER* DAN METODE *LEAST SIGNIFICANT BIT (LSB)*

Jane Irma Sari, Sulindawaty, Hengki Tamando Sihotang

Teknik Informatika
STMIK Pelita Nusantara Medan, Jl. Iskandar Muda No.1, Medan, Indonesia 20154

janejiss.jj@gmail.com, sulindawaty@gmail.com, hengki_tamando@yahoo.com

Abstrak

Pesatnya perkembangan teknologi di era saat ini, menyembunyikan pesan yang dirahasiakan dari orang yang tidak bertanggung jawab atau yang dapat mengakses pesan tersebut diperlukan suatu cara untuk menyembunyikan pesan tersebut. Salah satu cara penyembunyian pesan dalam pengiriman adalah merubah data menjadi yang tidak dimengerti dengan penyandian dan penyisipan menggunakan teknik kriptografi dan steganografi. Tujuan dari skripsi ini adalah untuk menghasilkan suatu aplikasi yang dapat menjaga dan memberikan keamanan yang berlapis tanpa mengurangi atau merusak pesan teks pada citra digital yang akan dikirm. Aplikasi ini dibangun menggunakan algoritma Hill Cipher dan metode Least Significant Bit, algoritma ini merupakan salah satu algoritma kunci simetris yang memiliki beberapa kelebihan dalam enkripsi data. Algoritma Hill Chiper menggunakan matriks berukuran m x m sebagai kunci untuk melakukan enkripsi dan dekripsi. Sedangkan steganografi dilakukan dengan menggantikan bit-bit terakhir pada gambar dengan bit pesan teks. Hasil dari penelitian ini adalah suatu aplikasi yang dapat melakukan proses encode dan decode citra digital dengan format bitmap.

Abstact

The rapid development of technology in the current era, hiding undisclosed messages from irresponsible people or who can access the message required a way to hide the message. One way of concealment of messages in transmission is to change the data to be not understood by encryption and insertion using cryptography and steganography techniques. The purpose of this thesis is to produce an application that can maintain and provide layered security without reducing or damaging text messages on digital images to be sent. This application is built using Hill Cipher algorithm and Least Significant Bit method, this algorithm is one of symmetric key algorithm which has some advantages in data encryption. Hill Chiper algorithm uses an mx m sized matrix as the key for encryption and decryption. While steganography is done by replacing the last bits of the image with text message bits. The result of this research is an application that can do the process of encode and decode digital image with bitmap format.

***Kata Kunci**: Kriptografi, Steganografi, Algoritma Hill Cipher, Citra Digital Bitmap, Metode Least Significant Bit.*

I. PENDAHULUAN

Seiring dengan majunya perkembangan teknologi, banyak tersedia beberbagai macam teknik untuk dapat melindungi pesan atau informasi yang dirahasiakan dari orang yang tidak berhak untuk mengakses pesan tersebut seperti pencurian data dan percobaan *hacking*. Maka dari itu agar dapat mempersulit para pihak yang tidak bertanggug jawab akan kejahatan komputer, penulis menggabungkan metode kriptografi yaitu algoritma *Hill Cipher* untuk enkripsi pesan dengan metode steganografi yaitu *Least Significant Bit* (LSB) yang mana bisa menambah keamanan dalam sebuah pesan.

Pada saat ini telah banyak teknik dilakukan untuk menjaga keamanan data dan informasi, seperti kriptografi dan steganografi. Steganografi adalah seni untuk menyisipkan pesan rahasia kedalam suatu media, dimana pesan rahasia yang akan disembunyikan tidak diubah bentuknya, melainkan disisipkan pada sebuah citra digital, sehingga orang lain tidak mengetahui bahwa di dalam citra digital tersebut ada pesan rahasia. Metode steganografi yang digunakan adalah metode *Least Significant Beast* (LSB). Metode ini merupakan penyembunyian pesan yang dilakukan mengganti bit-bit data yang kurang berarti dalam segmen citra dengan bit-bit rahasia pada bit terakhir. Sedangkan kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Proses enkripsi dilakukan menggunakan suatu algoritma dengan

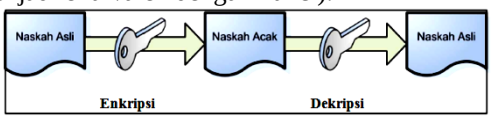
beberapa parameter. Algoritma yang digunakan pada penyusunan Penelitian ini yaitu *Hill Cipher*. Algoritma *Hill Cipher* merupakan salah satu algoritma kunci simetris yang memiliki beberapa kelebihan dalam enkripsi data.

Dalam penelitiani ini, penulis akan membahas lebih lanjut tentang implementasi penyembunyian pesan pada citra digital dengan menggabungkan algoritma *Hill Chiper* dan metode *Least Significant Bit* (LSB).

II. TEORI

A. Kriptografi

Apakah sebenarnya kriptografi itu? Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci).



Gambar 1. Proses enkripsi dan dekripsi

Gambar diatas menunjukkan efek dari proses enkripsi dan proses dekripsi. Secara garis besar, proses enkripsi adalah proses pengacakan “naskah asli" (*plaintext*) menjadi “naskah acak" (*ciphertext*) yang “sulit untuk dibaca" oleh seseorang yang tidak mempunyai kunci dekripsi. Yang dimaksud dengan “sulit untuk dibaca" disini adalah probabilitas mendapat kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu yang tidak terlalu lama adalah sangat kecil. Jadi suatu proses enkripsi yang baik menghasilkan naskah acak yang memerlukan waktu yang lama (contohnya satu juta tahun) untuk didekripsi oleh seseorang yang tidak mempunyai kunci dekripsi. [2]

B. Algoritma *Hill Cipher*

Hill Cipher diciptakan oleh Lester S. Hill pada tahun 1929. Teknik kriptografi ini diciptakan dengan maksud untuk dapat menciptakan *cipher* (kode) yang tidak dapat dipecahkan menggunakan teknik analisis frekuensi. *Hill Cipher* merupakan salah satu algoritma kriptografi kunci simetris yang memiliki beberapa kelebihan dalam enkripsi data. Untuk menghindari matrik kunci yang tidak *invertible*, matrik kunci dibangkitkan menggunakan *koefisien binomial newton*. Proses enkripsi dan deskripsi menggunakan kunci yang sama, *plaintext* dapat menggunakan media gambar atau text. 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. *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* 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* yang merupakan polyalphabetic cipher dapat dikategorikan sebagai *block cipher* karena teks yang akan diproses akan dibagi menjadi blok-blok dengan ukuran tertentu. Setiap karakter dalam satu blok akan saling mempengaruhi karakter lainnya dalam proses enkripsi dan dekripsinya, sehingga karakter yang sama tidak dipetakan menjadi karakter yang sama pula. *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*. [1]

C. Steganografi

Steganografi merupakan seni untuk menyembunyikan pesan di dalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan di dalam media tersebut. Kata steganografi (*steganography*) berasal dari bahasa Yunani *steganos* yang artinya ”tersembunyi/terselubung” dan *graphein* ”menulis” sehingga kurang lebih artinya “menulis (tulisan) terselubung”. Steganografi membutuhkan dua property, yaitu wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, audio, teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, audio, teks, atau video. Gambar dibawah ini adalah ilustrasi untuk menggambarkan proses penyimpanan (penyisipan) pesan ke dlaam media digital dengan teknik steganografi. [3]

Gambar 2. Proses Penyimpanan data rahasia ke dalam media digital dengan teknik steganografi

D. Citra Digital (*Bitmap*)

Citra adalah representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televise, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra digital adalah citra yang dapat diolah oleh komputer. Istilah citra digital sangat popular pada masa sekarang. Banyak peralatan elektronik, misalnya *scanner*, kamera digital, mikroskop digital, dan *fingerprint reader* (pembaca sidik jari), yang menghasilkan citra digital juga sangat popular digunakan oleh pengguna untuk mengolah foto atau untuk berbagai keperluan lain. Sebagai contoh, *Adobe Photoshop* dan GIMP (*GNU Image Manipulation Program*) menyajikan berbagai fitur untuk memanipulasi citra digital. [4]

Citra digital yang digunakan untuk menyembunyikan pesan yang adalah *Bitmap*. Yang paling penting dari kriteria ini adalah kedalaman warna (berapa banyak bit per *pixel* yang didefinisikan dari sebuah warna) sebagai berikut :

4 bit = 16 warna (16 *gray scales*).
8 bit = 256 warna (256 *gray scales*).
24 bit = 16.777.216 warna.

Secara umum semakin banyaknya warna, maka akan diperlukan keamanan yang ketat atau tinggi dikarenakan *bitmap* memiliki area yang sangat luas dalam sebuah warna yang seharusnya dihindarkan. Dilihat dari kedalaman atau kejelasan dari sebuah warna, *bitmap* dapat mengambil sejumlah data tersembunyi dengan perbandingan sebagai berikut (ukuran ratio dari *bitmap* dalam *byte* = ukuran dari data yang disembunyikan) :

4 bit = 16 warna : 4 : 1
8 bit = 256 warna : 8 : 1
24 bit = 16.777.216 warna : 8 : 1

Manipulasi pada *bitmap* tidak dapat dikonvert atau diubah ke dalam bentuk *form*at grafik yang lain karena data tersembunyi dalam *file* tersebut akan hilang. *Form*at menggunakan metode komperesi yang lain (seperti JPEG) tidak dapat digunakan. Mengurangi ukuran dari *file* pembawa sangatlah penting untuk melakukan transmisi *on-line*, yaitu dengan menggunakan utilitas kompresi (seperti : ARZ, LZH, PKZIP, WinZip), dikarenakan kerja mereka tidak terlalu berat. Untuk dapat menyisipkan pesan rahasia pada *file bitmap* maka terlebih dahulu harus diketahui struktur *file Bitmap*.

E. Metode *Least Significant Bit* (LSB)

Least Significant Bit adalah salah satu metode untuk menyembunyikan pesan dalam media digital dengan cara menyisipkan pesan tersebut pada satu bit paling kanan ke pixel file obyek. Dalam menyisipkan data pesan ke dalam berkas citra digital dengan menggunakan metode *Least Significan Bit* (LSB) *Modification*. Misalkan untuk menyisipkan suatu segmen pesan hasil dan modulasi sebesar 4 byte dengan modifikasi 1 bit LSB, maka dibutuhkan 32 data citra digital untuk menampungnya. dari segmen pesan ' 1 0 1 0 ' dengan 4 byte data citra digital sebagai berikut:
'0 1101110 00100011 01000010 01101101'
Maka dengan operasi penggantian bit terakhir dengan 4 bit segmen pesan secara berurutan menjadi sebagai berikut:
Data citra digital:
'0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 1 0 10 0 0 0 1 0 0 1 1 0 1 1 0 1'
Pesan:
1 0 1 0
Hasil:
'0 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 0'
Dengan sedikit modifikasi ini, maka efek dari perubahan nilai warna yang terjadi akibat perubahan bit tersebut tidak terlalu berpengaruh terhadap kualitas gambar. Perhatikan contoh untuk menyisipkan sebuah karakter A ke dalam citra *grayscale*. Sebuah pesan huruf A akan disisipkan ke dalam citra *grayscale* 8 bit ukuran 10x10 piksel.

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Gambar 3. Citra *grayscale* 8 bit 10x10 piksel

Langkah pertama adalah mengunbah kedua data tersebut (huruf A dan citra) menjadi biner. Nilai biner untuk A adalah 10000011. Karena jumlah digit biner huruf A hanya 8 bit maka jumlah piksel citra *grayscale* yang dibutuhkan cukup 8 piksel saja. Perhatikan 8 piksel pertama dari citra yang diubah menjadi biner.

8 piksel pertama diambil

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Gambar 4. Citra *grayscale* 8 piksel yang diambil
Langkah kedua adalah mengganti bit terakhir (LSB) dari piksel citra dengan bit-bit dari huruf A.

TABEL 1
PIKSEL CITRA YANG DIAMBIL

Piksel Citra		A	Piksel Citra yang berubah	
Decimal	Biner		Decimal	Biner
1	0000001	1	1	0000001
6	0000110	0	6	0000110
5	0000101	0	4	0000100
3	0000011	0	2	0000010
7	0000111	0	6	0000110
4	0000100	0	4	0000100
7	0000111	1	7	0000111
4	0000100	1	4	0000100

Perhatikan bit-bit yang ditandai dengan kotak. Bit-bit piksel citra mengalami perubahan (dalam hal ini yang berubah hanya 4 piksel saja) sehingga citra berubah menjadi:

8 piksel pertama diambil

1	6	4	2	6	4	7	5	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Tampak bahwa piksel-piksel yang mengakami perubahan hanya ± 1intensitas saja. Maka, secara kasad mata hal ini tidak begitu berpengaruh. Selain itu, tidak semua ppiksel mengalami perubahan intensitas.

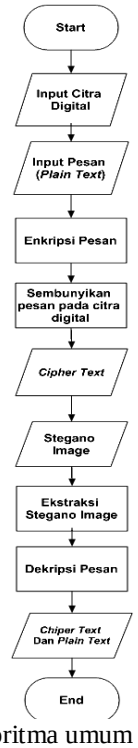
Gambar 5. Gambar *grayscale* piksel yang berubah

Ukuran data maksimum yang bisa disembunyikan dengan metode LSB yaitu seperti contoh yang berikut ini:
Media penampung : Citra *grayscale* 8 bit berukuran 64x32piksel.
Ukuran media penampung = 64 x 32 x 8 bit
= 16384 bit
1 piksel media penampung = 8 bit

Untuk menampung 1 bit data pesan diperlukan 1 piksel citra media penampung berukuran 8 bit karena setiap 8 bit hanya bisa menyembunyikan satu bit di LSB-nya. Oleh karena itu, citra ini hanya mampu menampung data pesan sebesar maksimum 16384/8 = **2048 bit dikurangi panjang nama filenya** karena penyembunyian data rahasia tidak hanya menyembunyikan isi data tersebut, tetapi juga nama filenya. Semakin besar data yang disembunyikan di dalam citra, semakin besar pula kemungkinan data tersebut rusak akibat manipulasi pada citra penampung.[3]

III. HASIL

Media yang digunakan untuk sistem steganografi yaitu citra digital, sedangkan pada kriptografi menggunakan algoritma *hill cipher* untuk enkripai pesan. Kemanan data ini dikombinasikan dengan teknik pengamanan data kriptografi dan teknik steganografi yang menjadikan data bersifat rahasia. Penggabungan ini akan memberikan keamanan berlapis sebagai tand kepemilikan tanpa mengurangi atau merusak pesan tersebut. Pesan yang sudah dienkripsi disisipkan, sehingga memberikan keadaan sesuatu yang tidak dapat dilihat (*invisibility*) yang cukup tinggi pada citra digital oleh kasat mata manusia. Untuk algoritma umum perangkat lunak penyembunyian pesan pada citra digital dapat dilihat pada gambar 6 dibawah ini :



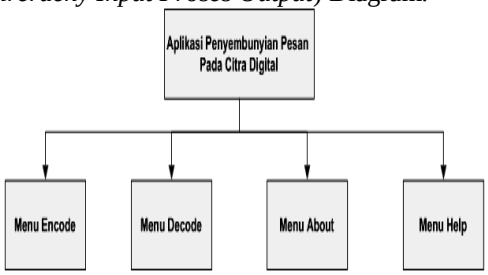
Gambar 6. Algoritma umum perangkat lunak

Tiap modul pada gambar 6. merupakan modul secara umum yang didalammya banyak terdapat sub modul yang memiliki fungsi lebih spesifik yang membangun keseluruhan program. Modul proses *input* data akan menggabungkan citra

digital dan pesan rahasia dengan fungsi stego menjadi suatu *stegano image* serta algoritma *hill cipher* untuk enkripsi pesan. Modul proses *output* data akan memisah kembali antara *file* citra digital dan data pesan rahasia pada suatu *stegano image* serta melakukan dekripsi pesan menjadi *ciphertext*. Detail dari proses ini akan dijelaskan pada sub bahasan berikutnya.

A. Perancanga Antarmuka

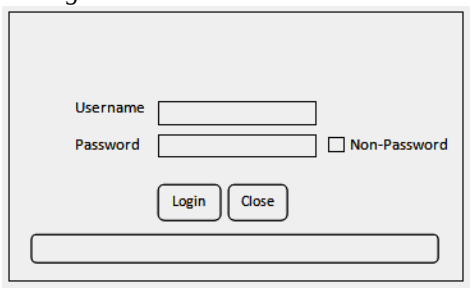
Perlunya kenyamanan dan kemudahan dalam mengakses atau menggunakan sebuah aplikasi, maka dari itu dirancang tampilan-tampilan yang mudah dipahami oleh pengguna atau biasanya disebut *user friendly*. Perancangan antarmuka berikut ini akan digambarkan dengan HIPO (*Hirerachy Input Proses Output*) Diagram.



Gambar 7. HIPO Diagram Rancangan Antarmuka

1. Halaman Menu *Login*

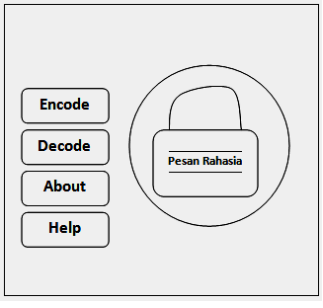
Halaman menu *login* merupakan halaman yang pertama kali muncul pada saat sistem dijalankan. Dimana halaman *login* ini terdiri dari menu *username* dan *password*, *checkbox Non-password*, tombol *login* dan *cancel*.



Gambar 8. Menu Login

2. Halaman *Form* Utama

Halaman *form* utama merupakan halaman yang digunakan pengguna untuk memilih pilihan yang diinginkan, di *form* utama ada 4 menu tombol button yaitu. Pertama *button encode* untuk melakukan proses penyembunyian pesan pada citra digital, yang kedua *button decode* untuk melakukan proses ekstraksi pesan dari steganografi *image*.



Gambar 9. Menu Utama

B. Pembahasan

Membahas tentang hasil dari sistem berdasarkan studi kasus yang diangkat sehingga dapat diketahui tingkat keberhasilan dari sistem yang dibangun, apakah hasil sudah sesuai dengan yang diharapkan sehingga permasalahan terpecahkan.

1. Pengujian Algoritma *Hill Cipher*

Sebelum masuk kedalam proses penyandian terlebih dahulu ditetapkan pesan yang akan disandikan dan kunci matriks 2x2. Pesan yang disandikan maksimal 66 karakter tiap karakter harus berada diantara A-Z yang berjumlah 26 huruf dalam proses penyandian ini huruf besar dan kecil tidak dibedakan. Dan juga dalam penyandian ini tidak menggunakan kode ascii huruf tetapi dengan kode angka berdasarkan urutan huruf yaitu A=0, B=1, ... Z=25 dan spasi tidak dihitung dalam penyandian. Berdasarkan persamaan (4) maka kunci yang telah ditetapkan harus kita gunakan di dalam proses enkripsi pesan berikut ini adalah langkah-langkah penyandian pesan mengunakan kunci matriks 2x2, yaitu sebagai berikut :

a) Sisipkan pesan dan kunci matrik

Pesan :

P = Halo

Kunci :

$K = \begin{bmatrix} 1 & 5 \\ 9 & 8 \end{bmatrix}$

Untuk mengetahui kode-kode dari pesan diatas dibawah ini akan diberikan tabel kode masing-masing huruf dari A sampai dengan Z, dapat dilihat pada gambar 10.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 10. Konversi Abjad Menjadi Angka

b) Mengubah pesan menjadi kode dan matrik 2x2

Setelah diketahui masing-masing kode huruf maka pesan teks yang akan di sandikan diubah kedalam kode-kode angka dari tabel diatas, yaitu :

Kode pesan :

P = 7, 0, 11, 16

Kemudian setelah didapat kode untuk masing-masing huruf kemudian setiap dua kode diubah kedalam matriks ordo 1 x 2, agar dapat

dikalikan dengan kunci yang mempunyai matriks 2x2.

$$P = \begin{bmatrix} 7 \\ 0 \end{bmatrix} \begin{bmatrix} 11 \\ 14 \end{bmatrix}$$

Setelah matrik disusun maka Rumus : C=(P*K) mod 26 digunakan sesuai dengan persamaan (6).

c). Mengalikan matriks pesan dan matriks kunci

Matriks pesan dikalikan dengan matriks kunci, dan hasil perkalikan tersebut diubah lagi kedalam huruf dengan referensi tabel kode masing-masing huruf. Dibawah ini adalah proses perkalian matriks kunci dengan matriks pesan, yaitu sebagai berikut:

$$C = \begin{bmatrix} 1 & 5 \\ 9 & 8 \end{bmatrix} x \begin{bmatrix} 7 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 63 \end{bmatrix}$$

Karena hasil perkalian melebihi 25 maka hasil perkalian ini harus di mod 26, yaitu sebagai berikut:

$$C = \begin{bmatrix} 7 \\ 63 \end{bmatrix} Mod\ 26 = \begin{bmatrix} 7 \\ 11 \end{bmatrix}$$

Untuk K*P (LO)

$$C = \begin{bmatrix} 1 & 5 \\ 9 & 8 \end{bmatrix} x \begin{bmatrix} 11 \\ 14 \end{bmatrix} = \begin{bmatrix} 81 \\ 211 \end{bmatrix} mod\ 26 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Setelah semua matriks pesan dikalikan dengan matriks kunci dan lakukan modulus dengan 26 maka hasil matrik *cipher* adalah sebagai berikut :

$$\begin{bmatrix} 7 \\ 11 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

d). Mengubah matriks menjadi deret pesan

Setelah didapat hasil matriks dari perkalian antara matriks kunci dan pesan kemudian matriks disusun kembali berurutan, yaitu sebagai berikut :

C = 7, 11, 3, 3

e). Mengubah kode pesan menjadi huruf (karakter)

Kemudian kode diatas disusun kembali kedalam bentuk huruf dengan menggunakan gamba 10. Dan hasil ini adalah chiper hasil dari penyandian pesan yang akan di gunakan dalam proses steganografi, yaitu sebagai berikut :

Cipherteks : HLDD

Setelah proses enkripsi selesai dilakukan, maka untuk mendeskripsi cipherteks menjadi pesan kembali sebenarnya hampir sama dengan cara enkripsi. Tetapi kunci yang digunakan harus di invers terlebih dahulu.

Untuk lebih jelasnya tentang proses dekskripsi chiperteks diatas, dibawah ini dijelaskan secara rinci tentang tahap-tahap deksripsi yang dimaksud yaitu :

A. Invers matriks kunci

Didalam pengembalian pesan terdapat ketetapan yang telah ditentukan, tahap pertama yaitu dengan melakukan invers terhadap matriks kunci yang digunakan dalam penyandian pesan. Dibawah ini adalah rumus untuk menginvers matriks yang ber ordo 2x2. Proses untuk mengetahui invers matriks kunci invers tersebut adalah sebagai berikut:

$$K^{-1} = \begin{bmatrix} 4 & 17 \\ 15 & 7 \end{bmatrix}$$

Setelah invers matriks kunci selesai dilakukan maka didapat matriks. Kunci-1 untuk mendeskripsikan chiper menjadi pesan kembali matrik invers kunci dikalikan dengan matriks chipperteks. Tetapi terlebih dahulu mengubah chipperteks ke dalam bentuk kode kembali sama persis seperti proses pengubahan pesan kedalam kode sewaktu proses penyandian.

B. Menyiapkan pesan cipher

Cipherteks : HLDD

Setelah diketahui cipherteks, setelah itu diubah kedalam bentuk kode berdasarkan urutan angka yang ada dalam gambar 10 diatas.

C. Mengubah chiper menjadi kode dan matriks

C = 7, 11, 3, 3

Setelah diketahui masing-masing kode dari cipherteks, kemudian diubah kedalam bentuk matriks *cipher*, sebagai berikut :

$$C = \begin{bmatrix} 7 \\ 11 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Setelah diketahui matriks chipper kemudian chiper dikalikan dengan invers matrik kunci dan modulus dengan 26 dan hasil itu adalah pesan asli dari penyandian yang telah dilakukan sebelumnya. Berikut adalah rumus untuk menentukan pesan teks dari chiperteks.

P = K-1*C

D. Mengalikan matriks pesan chipper dengan invers matriks kunci

Berikut ini adalah proses perkalian antara invers matriks kunci dengan matriks chipperteks, yaitu sebagai berikut :

untuk kata HA:

$$P = \begin{bmatrix} 4 & 17 \\ 15 & 7 \end{bmatrix} x \begin{bmatrix} 7 \\ 11 \end{bmatrix} = \begin{bmatrix} 215 \\ 182 \end{bmatrix} mod\ 26 = \begin{bmatrix} 7 \\ 0 \end{bmatrix}$$

untuk kata LO:

$$P = \begin{bmatrix} 4 & 17 \\ 15 & 7 \end{bmatrix} x \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 63 \\ 66 \end{bmatrix} mod\ 26 = \begin{bmatrix} 11 \\ 14 \end{bmatrix}$$

Dan seletelah semua matriks chipper kalikan dengan inverst matriks kunci K-1 maka didapat hasil perkalian dan modulus 26 adalah sebagai berikut :

$$P = \begin{bmatrix} 7 \\ 0 \end{bmatrix} \begin{bmatrix} 11 \\ 14 \end{bmatrix}$$

E. Mengubah matriks pesan menjadi deret kode

Setelah diketahui masing-masing matriks pesan kemudian matriks pesan diurutkan kedalam bentuk bilangan bulat biasa seperti dibawa ini:

P = 17, 0, 3, 8

F. Mengubah kode menjadi pesan kembali

Dan langkah terakhir adalah mengubah pesan yang masih berbentuk kode menjadi bentuk huruf dengan berpedoman pada tabel kode

masing-masing huruf yang tertera pada tabel IV.1, dah hasilnya adalah sebagai berikut :
P = HALO

2. Pengujian Metode *Least Significant Bit* (LSB)
Dalam pembasan ini dijelaskan tentang bagaimana proses penyisipan pesan chiper kedalam citra gambar. Gambar yang digunakan adalah gambar berwarna 24 bit, yaitu gambar yang terdiri dari 3 warna R, G, B masing-masing warna mempunyai kedalaman warna sebesar 8 bit. Karena masing-masing warna bernilai 8 bit, maka pesan akan disisipkan kedalam bit R, bit G dan bit B tiap-tiap pixel. Misalkan pesan yang akan disisipkan sebanyak 8 bit, maka pesan yang 8 bit tersebut hanya akan disisipkan pada dua 3 pixel, karena tiap pixel memiliki kapasitas 24 bit dan masing-masing bit pesan hanya disisipkan pada 8 bit citra gambar. Dibawah ini adalah langkah-langkah proses steganografi untuk menyisipkan pesan kedalam citra gambar, Berikut ini merupakan bagaimana cara kerja dari algoritma LSB dimana teks HALO akan disisipkan kedalam gambar, namun terlebih dahulu teks tersebut diubah kedalam biner dengan nilai sebagai berikut:

TABEL 2.
KODE ASCII PESAN YANG AKAN DISISIPI

Teks	Biner
H	01001000
A	01100001
L	01101100
O	01101111

Setelah diubah kedalam biner lalu pesan akan disisipkan kedalam gambar pada warna (RGB) dengan metode lsb dengan nilai biner gambar awal sebagai berikut:

TABEL 3.
KODE MEDIA/GAMBAR YANG AKAN DISISIP

01110111	01110110	01110100	01000111
01110011	01110100	01110110	01110000
00110110	01110111	11110111	01110110
10110111	11110111	01110111	11110111
11010111	01110110	11110111	01110110
11110111	10110111	01111111	01110111
01110100	11000111	11110111	00010111
01111110	11010111	01111111	01110100

Setelah disisipkan pesan maka nilai biner gambar tersebut akan berubah menjadi berikut ini:

TABEL 4.
KODE MEDIA/GAMBAR YANG SUDAH DISISIP

0111011**0**	01110110	01110100	0100011**0**
01110011	0111010**1**	0111011**1**	0111000**1**
00110110	01110111	11110111	0111011**1**
1011011**0**	1111011**0**	0111011**0**	1111011**0**
11010111	01110110	11110111	0111011**1**
1111011**0**	1011011**0**	01111111	01110111
01110100	1100011**0**	1111011**0**	00010111
0111111**1**	11010111	0111111**0**	0111010**1**

Untuk proses ekstraknya adalah mengambil nilai paling kanan dari biner yang disisipkan. Data biner yang telah diambil isi pesannya dimana nilai tersebut adalah sebagai berikut:

TABEL 5.
KODE ASCII PESAN YANG AKAN DISISIPI

Teks	Biner
H	01001000
A	01100001
L	01101100
O	01101111

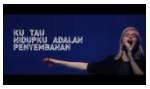
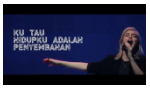
Untuk menganalisa hasil, penulis menggunakan beberapa pengujian yaitu pengujian hasil teori dan hasil praktek (pengaplikasian), lalu pengujian perbandingan gambar sebelum dan sesudah pesan disisipkan. Adapun analisa hasil pada enkripsi *Hill Cipher* ini, dapat dilihat pada tabel 6:

TABEL 6.
ANALISA HASIL PADA ENKRIPSI *HILL CIPHER*

Analisa Hasil Pada Enkripsi Hill Cipher		
Hill Cipher	Teori	Praktek
Pesan	HALO	HALO
Hasil	Huruf H menjadi H Huruf A menjadi L Huruf L menjadi D Huruf O menjadi D Sehingga kata HALO menjadi kata HLDD	Dalam prakteknya dengan menekan tombol submit maka hasilnya adalah sebagai berikut: 

Pengujian selanjutnya adalah analisa hasil sebelum dan sesudah enkripsi/disisipkan pesan pada citra BMP. Adapun pesan yang disisipkan adalah “Proses mengubah citra analog menjadi citra digital disebut digitalisasi citra.
Adapun hasilnya dapat dilihat pada tabel 7:

TABEL 7.
ANALISA HASIL PADA CITRA BMP

Analisa Hasil Pada Citra BMP		
Detail	Citra Asli	Citra Stegano
Nama Citra	j.bmp	Halo.bmp
Ukuran(Mb)	2.63	2.63
Dimensi(Pixel)	1297x720	1297x720
Citra Bmp		

IV. KESIMPULAN

Berdasakan dari analisa, perancangan dan implementasi pada aplikasi penyembunyian pesan pada citra digital dengan menggabungkan algoritma *hill cipher* dan metode *least significant bit* (LSB), dapat diambil kesimpulan sebagai berikut:

1. Proses penyembunyian pesan pada citra digital aman dan tidak diketahui secara kasat mata, karena besar *bitmap* hasil steganografi tidak mengalami perubahan setelah proses penyisipan biner teks ke dalam biner *bitmap* menggunakan metode *least significant bit* (LSB) yaitu penggantian bit terakhir sehingga kapasitas *bitmap* sebelum dan sesudah disteganografi tidak mengalami perubahan yang berarti .
2. Pengujian pesan teks menggunakan algoritma *hill cipher* berhasil dilakukan sesuai tepat dengan alur atau langkah-langkah sehingga menghasilkan cipherteks yang berupa pengacakan huruf abjad.
3. Pesan yang akan diambil dari *bitmap* dapat dilanjutkan ke proses dekripsi yang bertujuan untuk mengembalikan pesan cipherteks ke bentuk semula (*plaintext*) melalui proses dekripsi algoritma *hill cipher* yang sesuai.
4. Pengujian dilakukan dengan menjalankan aplikasi *encode* dan *decode*, dapat diambil kesimpulan bahwa pengungkapan data kembali berhasil dan pesan teks dapat diambil.
5. Pengujian dilakukan dengan menggunakan *Visual Basic* 2010, pesan yang disembunyikan pada citra digital dengan format *bitmap* tersebut menggunakan metode *least significant bit* (LSB) dapat terdeteksi.

V. REFERENSI

[1] Muamal Khoerudin (2015, Maret 22-29). *Algoritma Hill Cipher (Sandi Hill)* Materi Perkuliahan Pada Jurusan Teknik Informatika. Ditemukenali 24 Agustus 2017, dari https://muamalkhoerudin.wordpress.com/2015/03/22/algoritma-hill-cipher-sandi-hill/

[2] Sentot Kromodimoeljo. 2010. *Teori & Aplikasi Kriptografi*, SPK IT Consulting, Jakarta, p.5-6.

[3] T. Sutoyo et al. *Teori Pengolahan Citra Digital*, Andi Yogyakarta dengan UDINUS Semarang, Semarang, p.245-249, 2009.

[4] Abdul Kadir dan Adhi Susanto, *Teori dan Aplikasi Pengolahan Citra*, Andi Yogyakarta, Yogyakarta, p.2, 2013.