

MODIFIKASI HILL CIPHER MOD 36 KOMBINASI TRANSPOSISI DAN XOR KUNCI TABEL PERIODIK DENGAN LSB UNTUK PENYEMBUNYIAN PESAN

Eka Yulia Sari¹, Ahmad Rifki Harir², Dony Ariyus³

^{1,2,3} Universitas Amikom Yogyakarta

Jalan Ringroad Utara, Condongcatur, 55582

¹ekasari2107@gmail.com, ²ahmad.1105@students.amikom.ac.id, ³dony.a@amikom.a.cid

Abstrak— Saat ini Indonesia memasuki industri 4.0, dimana apapun berhubungan dengan teknologi. Pertukaran data dan informasi secara digital lebih efektif dan efisien sehingga memudahkan penggunaannya. Tentunya data dan informasi ada yang dapat diakses publik dan ada yang bersifat rahasia. Saat terjadi pertukaran data, tidak dapat dipungkiri bahwa pencurian data dan manipulasi data oleh orang yang tidak berhak dapat terjadi. Peningkatan keamanan data dapat dilakukan dengan kriptografi dan steganografi. Pada penelitian ini, algoritma modifikasi hill cipher mod 36 dengan transposisi pergeseran, operasi xor dan transposisi yang dikombinasikan dengan penyisipan pesan kedalam gambar menggunakan LSB digunakan untuk mengamankan pesan teks. Hasil yang diperoleh dari modifikasi hill cipher mod 36 yang diusulkan yaitu dapat digunakan untuk mengenkripsi pesan dengan karakter a sampai z dan 0 sampai 9. Jumlah pesan sebesar 975 disisipkan kedalam gambar berekstensi .jpg dengan hasil MSE sebesar 0,00498 dan PSNR sebesar 0,00498.

Kata Kunci— Modifikasi Hill Cipher, Hill Cipher Mod 36, Steganografi LSB, Least Significant Bit

Abstract— Currently, Indonesia enters the 4.0 industry, where anything relates to technology. Data exchange and information digitally more effectively and efficiently to facilitate the user. Obviously, data and information are publicly accessible and some are confidential. When data exchange occurs, it is undeniable that data theft and data manipulation by unauthorized persons can occur. Improved data security can be done with cryptography and steganography. In this study, the algorithm modified Hill cipher mod 36 with a transposition shift, XOR operation and transposition combined with the insertion of the message into the image using LSB used to secure the text message. The result obtained from the proposed Hill cipher MoD 36 Modification is that it can be used to encrypt messages with characters A to Z and 0 to 9. The number of messages 975 is inserted into the image with an extension of .jpg with MSE result of 0.00498 and PSNR of 0.00498.

Keywords— Modification Hill Cipher, Hill Cipher Mod 36, Steganography LSB, Least Significant Bit

I. PENDAHULUAN

Era teknologi saat ini sangat memudahkan dalam pertukaran data secara digital sehingga lebih efektif dan efisien. Data merupakan informasi rahasia yang sangat penting untuk perorangan, kelompok maupun perusahaan [1]. Biasanya data yang dikirim merupakan data rahasia yang tidak boleh diakses oleh orang yang tidak memiliki kepentingan atas data tersebut. Pertukaran data yang terjadi harus dapat dijamin keamanan dan keutuhannya. Saat pertukaran data terjadi, tidak dipungkiri bahwa pencurian data oleh orang yang tidak berhak dapat terjadi. Banyak orang yang melakukan segala cara untuk mendapatkan data secara ilegal tanpa melalui prosedur resmi.

Pencurian data termasuk kedalam masalah keamanan data yang harus menjadi perhatian khusus. Salah satu pengamanan data dapat dilakukan dengan teknik kriptografi. Kriptografi merupakan seni dalam keamanan suatu pesan/informasi ketika dikirimkan dari satu tempat ke tempat lain dengan menjaga privasi dan

keamanan data dari gangguan orang yang tidak berhak menerima informasi/pesan yang ingin disampaikan dengan cara disembunyikan [2]. Selain itu, metode kriptografi bertujuan untuk membuat pesan yang dikirim berubah sehingga keamanan pesan akan meningkat [3] [4]. Proses yang dilakukan dalam kriptografi adalah enkripsi dan dekripsi. Enkripsi dilakukan oleh pengirim ketika data akan dikirimkan dan dekripsi akan dilakukan setelah data diterima oleh penerima.

Beberapa metode dalam kriptografi untuk mengamankan data sudah banyak digunakan, termasuk teknik hill cipher. Teknik hill cipher merupakan kriptosistem polialfabetik yang menggunakan aritmatika modulo [5]. Teknik hill cipher menggunakan matrik sebagai tempat pertukaran informasi. Hill cipher termasuk kedalam algoritma kriptografi kunci simetris karena hanya menggunakan satu kata kunci untuk enkripsi dan dekripsi. Penerima suatu pesan yang dienkripsi menggunakan metode Hill cipher harus

diberitahukan kunci yang digunakan. Serangan dalam algoritma Hill cipher bisa dilakukan oleh cryptanalysis dengan brute force, untuk mengatasi hal tersebut perlu dilakukan modifikasi dari metode hill cipher.

Modifikasi Hill cipher sudah banyak dilakukan, seperti penambahan metode transposisi atau operasi XOR. Modifikasi terhadap suatu algoritma diharapkan mampu meningkatkan keamanan dari pesan yang akan dikirim. Seperti yang dilakukan oleh [6] memodifikasi hill cipher dengan teknik penyembunyian pesan metode LSB. Pada penelitian [7] memodifikasi hill cipher dengan kunci matrix persegi panjang dengan fungsi XOR dan fungsi XNOR. Pada penelitian [7] menyimpulkan bahwa hill cipher dapat dimodifikasi dengan menambahkan operasi biner dan pembagian bit. Sedangkan penelitian yang dilakukan oleh [8] yang memodifikasi hill cipher dengan kunci kode wilayah telepon dan algoritma twofish.

Berdasarkan penelitian terdahulu, Peneliti tertarik untuk meningkatkan keamanan pesan dengan mengenkripsi pesan dengan modifikasi hill cipher modulus 36 yang dimodifikasi dengan transposisi dan operasi XOR. Pada penelitian ini, pemanfaatan algoritma LSB untuk menyembunyikan pesan kedalam gambar untuk meningkatkan keamanan dari pesan tersebut. LSB digunakan karena mempunyai kemampuan dalam menyembunyikan pesan yang tidak dapat dirasakan oleh persepsi visual manusia dan kemampuannya dalam menampung jumlah data [9]. Penggunaan metode penyembunyian pesan kedalam gambar bertujuan untuk mengurangi kecurigaan terhadap pesan teks yang terenkripsi karena pada umumnya pesan teks yang terenkripsi berbentuk pesan acak dan tidak bermakna [10].

II. LANDASAN TEORI

A. Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *crypto* (*secret/* rahasia) dan *graphia* (*writung/tulisan*). Kriptografi merupakan seni dalam menjadi keamanan suatu pesan/informasi ketika dikirimkan dari satu tempat ke tempat lain. Kriptografi dapat digunakan untuk menjaga privasi dan keamanan data dari gangguan orang yang tidak berhak menerima informasi/pesan yang ingin disampaikan dengan tujuan untuk menjaga kerahasiaan data agar tidak bocor sehingga tidak merugikan bagi pengirim maupun penerima pesan.

B. Hill Cipher

Merupakan *cryptosystem* polyalphabetic yang ditemukan pada tahun 1929 oleh Lester S. Hill. Dalam teknik ini, setiap karakter alfabet dapat dipetakan ke lebih dari satu macam karakter alfabet. Kunci yang digunakan untuk enkripsi dan dekripsi diambil dari rumus yang sama. Kunci untuk mengenkripsi diinvers sebelum digunakan untuk mendekripsi ciphertext. Teknik hill cipher menggunakan matriks sebagai

tempat pertukaran informasi baik enkripsi maupun dekripsi. Secara umum, matriks yang digunakan pada hill cipher adalah perkalian antar matriks dan invers dari matriks. Proses enkripsi pada hill cipher terlihat pada persamaan 1 berikut(1):

$$\begin{bmatrix} C1 \\ C2 \\ C3 \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ CP3 \end{bmatrix} \text{ mod } TC$$

$$\begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix} \begin{bmatrix} C1 \\ C2 \\ C3 \end{bmatrix} \text{ mod } TC$$

C merupakan ciphertext, P adalah plaintext dan K adalah kunci, serta TC adalah total penggunaan karakter. Nilai C1 diperoleh dari perkalian antara K11, K12, K13 dan P1, P2, P3. Hasil perkalian tersebut akan mengalami modulo terhadap total karakter yang digunakan. Pada persamaan 2, terlihat proses dekripsi yang mana berkebalikan dengan proses enkripsi pada persamaan 1. Pada proses dekripsi, nilai K harus diubah menjadi K inverse terlebih dahulu dengan ketentuan nilai determinan dari matriks kunci harus bernilai 1. Jika tidak maka ciphertext tidak bisa berubah menjadi plaintext atau teks asli.

C. Steganografi

Steganography merupakan cabang ilmu yang mempelajari cara menyembunyikan informasi rahasia kedalam informasi lainnya [2]. Berbeda dengan kriptografi, steganography menyembunyikan data ke dalam data atau informasi lain bisa berupa gambar, dokumen, suara dan video. Jika kriptografi melakukan pengacakan data asli sehingga menghasilkan data yang sudah berbentuk ciphertext (terenkripsi) yang benar-benar acak dan sangat berbeda dengan aslinya, sedangkan steganography menyembunyikan ke dalam data lain yang tanpa merubah data yang ditumpanginya sehingga data yang ditumpanginya sebelum dan sesudah ditumpanginya bentuknya hampir sama. Kelebihan steganography dari pada kriptografi adalah dapat merahasiakan pesan tanpa diketahui bahwa pesan sedang dikirim, namun steganography memerlukan banyak ruang untuk dapat menyembunyikan beberapa bit pesan.

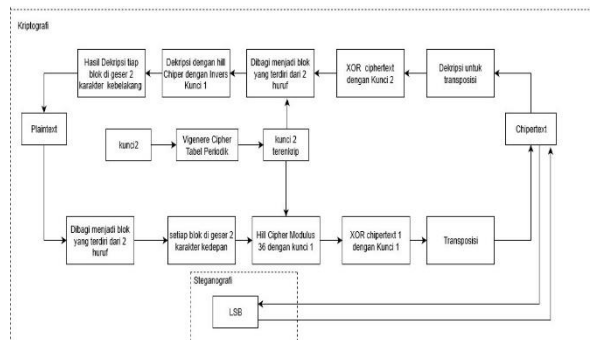
D. Metode LSB (Least Significant Bit)

Metode LSB atau least significant bit merupakan salah satu teknik penyembunyian pesan pada media digital seperti file image dengan menyisipkan pada bit rendah atau bit yang paling kanan. File image tersusun atas bit-bit dengan susunan angka 0 dan angka 1, sehingga teknik LSB ini berbasis dengan biner dengan angka 0 dan 1. Teknik LSB mengubah bit pada posisi least significant bit pada image yang akan disisipi dengan bit text yang akan disembunyikan. Bit yang dilakukan perubahan merupakan bit yang paling akhir, sehingga mata manusia sulit mengenali adanya

perbedaan image yang belum tersisipi dan yang sudah tersisipi.

III. METODE PENELITIAN

Pada penelitian ini, metode kriptografi dan steganografi di gabungan untuk mengamankan sebuah pesan. Metode kriptografi yang digunakan adalah hill cipher modulus 36 yang dimodifikasi dengan transposisi dan operasi XOR. Matrik kunci yang digunakan pada hill cipher adalah 2x2. Modifikasi dilakukan dengan merubah modulo 26 menjadi modulo 36. Gambar 1 berikut merupakan alur enkripsi dan dekripsi algoritma dalam penelitian ini..



Gbr. 1 Alur Enkripsi dan Dekripsi

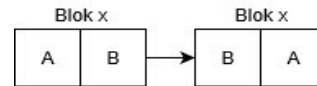
Penelitian ini menggunakan beberapa langkah dalam memodifikasi hill cipher. 2 kunci digunakan untuk mengenkripsi plaintext yaitu kunci untuk hill cipher dan kunci untuk operasi XOR. Keamanan kunci untuk operasi XOR dilakukan dengan mengkonversi kunci dengan unsur dalam tabel periodik. Modifikasi algoritma yang diusulkan yaitu *plaintext* dibagi menjadi blok-blok yang terdiri dari 2 huruf, selanjutnya masing-masing blok dilakukan pergeseran antar huruf. Masing-masing blok yang sudah dilakukan pergeseran, selanjutnya dilakukan enkripsi dengan Hill Cipher mod 36 menggunakan kunci 1. Setelah hasil enkripsi pertama menggunakan hill cipher didapatkan, kemudian dilakukan operasi XOR dengan kunci 2 yang sudah terenkripsi. Selanjutnya Hasil dari operasi XOR akan diproses lagi dengan transposisi cipher.

Pesan teks yang terenkripsi selanjutnya akan disembunyikan kedalam dokumen gambar dengan metode LSB. Sebelum pesan disisipkan kedalam tiap-tiap bit piksel dalam gambar, perlu dilakukan konversi pesan kedalam bit-bit. Pesan yang sudah dalam bentuk bit selanjutnya disisipkan kedalam bit piksel dengan metode LSB. Dimana penyisipan pesan dengan teknik LSB adalah proses mengganti bit pada posisi LSB yang paling akhir atau yang paling kanan. Setelah pesan sudah disisipkan, kami melakukan pengujian terhadap sistem yang sudah dibuat.

A. Proses Enkripsi Transposisi 2 huruf

Pada proses ini, plaintext awal dilakuk kan penghapusan spasi dan merubah huruf besar menjadi huruf kecil semua. Plaintext awalnya berupa beritipe

string huruf dan angka dikonversi menjadi huruf bertipe double. Konversi dilakukan untuk mempermudah perhitungan dalam proses hill cipher. Selanjutnya pembagian plaintext yang sudah dikonversi kedalam blok-blok yang berisi 2 huruf. Apabila ada blok yang tidak berisi 2 huruf penuh, dapat ditambahkan dengan huruf "x" sehingga setiap blok harus berisi 2 huruf. Pembagian 2 huruf perblok dikarenakan pada penelitian ini matrix kunci yang digunakan berordo 2x2. Setelah blok-blok di dapatkan, huruf-huruf pada setiap blok digeser, ilustrasi bisa dilihat pada gambar 2 berikut :



Gbr. 2 Ilustrasi Pergeseran Huruf

B. Proses Enkripsi Hill Cipher

Kelanjutan dari proses transposisi 2 yaitu proses enkripsi hill cipher yang menggunakan modulus 36. Hill cipher pada umumnya hanya sebatas a sampai z, pada penelitian ini hill cipher tidak hanya a sampai z melainkan a sampai z dan 0 sampai 9 sehingga modulus yang digunakan sebesar 36. Gambar 3 merupakan *screen shoot* source code proses enkripsi hill cipher modulus 36 yang digunakan pada penelitian ini :

```

24 trans_awal=double(trans_awal)/konversi ke double dari variabel trans_awal
25 [~,lk]=size(trans_awal);
26 for i=0:lk-1
27     i=i+1
28     if trans_awal(i,i)>=97
29         konvert_h(i,i)=trans_awal(i,i)-97;
30     elseif trans_awal(i,i)<=97
31         konvert_h(i,i)=trans_awal(i,i)-22;
32     end
33 end
34 end
35 p= [2 3; 3 5]; %kunci Hill Cipher
36 pb=lk/2;
37 H= reshape(konvert_h,2,pb);
38 E=M^H;
39 hasil_hill=mod(E,36);
40 hsl_enkrip_h= reshape(hasil_hill,1,lk);
41 for i=0:lk-1
42     i=i+1
43     if hsl_enkrip_h(i,i)>=27
44         konvert_hsl(i,i)=hsl_enkrip_h(i,i)+22;
45     elseif hsl_enkrip_h(i,i)<=26
46         konvert_hsl(i,i)=hsl_enkrip_h(i,i)+97;
47     end
48 end
49

```

Gbr. 3 Screen Shoot Enkripsi Hill Cipher Mod 36

Proses dekripsi menggunakan hill cipher mod 36 sama dengan proses dekripsi hill cipher mod 26, namun yang membedakan adalah mod yang digunakan. Matrik kunci pada modulus 36 harus memiliki determinan 1, karena jika tidak 1 akan kesulitan dalam proses dekripsi. Langkah-langkah dalam dekripsi hill cipher mod 36 dengan matrik ber ordo 2x2 sebagai berikut :

a) Dimisalkan matrik kunci dengan ordi 2x2 adalah

$$\text{sebagai berikut : } \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

b) Cari determinan dari maatrik kunci ordo 2x2 dengan persamaan (2):

$$\text{Matrik kunci} = (a * d) - (c * b)$$

c) Cari nilai invers Modulo matrik kunci⁻¹ mod 36 dengan persamaan (3):

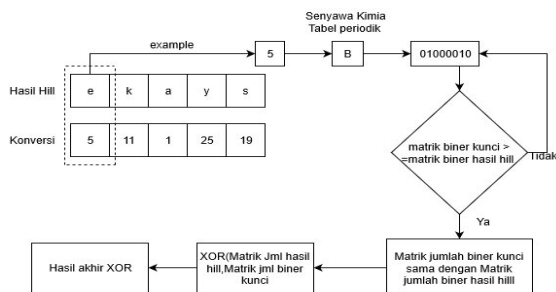
$$\begin{aligned} \text{Modulo matrik kunci mod 36} &= K \\ &= \frac{36(k) + 1}{\text{hasil matrik kunci}} \end{aligned}$$

K adalah percobaan angka untuk menghasilkan nilai invers modulo matrik kunci dengan hasil bilangan bulat positif.

- d) Hasil nilai invers modulo matrik kunci di kali dengan invers matrik sehingga menghasilkan kunci dekripsi. Dimana ketentuan invers dari matrik kunci adalah sebagai berikut : $\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$
- e) Selanjutnya kunci dekripsi di kali dengan masing-masing huruf yang di mod 36. Proses ini sama dengan proses enkripsi, namun yang membedakan kunci yang digunakan merupakan kunci dekripsi yang sudah jelaskan diatas.

C. Proses Operasi XOR

Operasi XOR dilakukan dengan operasi XOR antar kunci 2 terenkripsi dan hasil hill cipher. Kunci 2 dienkripsi dengan mengubah terlebih dahulu kedalam bentuk angka, selanjutnya di konversi kedalam bentuk unsur senyawa pada tabel periodik. Hasil enkripsi kunci 2 dan hasil hill cipher akan diubah kedalam bentuk bilangan biner karena operasi XOR hanya dapat dilakukan dengan biner. Ilustrasi proses operasi xor dapat terlihat pada gambar berikut.



Gbr 4 Ilustrasi Proses Operasi XOR

Pada gambar 4 terlihat bahwa matrik antar biner kunci dan biner hasil hill cipher harus sama, sehingga untuk mendapatkan matrik biner kunci diperlukan perulangan penyimpanan biner sampai jumlah matrik sama dengan jumlah matrik biner hasil hill. Setelah proses perulangan tersebut, operasi XOR dapat dilakukan dan mendapatkan matrik berisi hasil operasi biner. Hasil operasi biner selanjutnya di konversi kedalam bentuk char sehingga dapat dilakukan proses enkripsi ditahap selanjutnya.

Seperti yang sudah dijelaskan diatas, bahwa kunci 2 yang digunakan untuk operasi XOR akan dienkripsi terlebih dahulu dengan mengkonversi kunci kedalam unsur senyawa kimia pada tabel periodik. Aturan pengkonversian kunci terlihat pada gambar 5. Aturan dalam menginput kunci 2 untuk operasi XOR yang akan dikonversi hanya dapat menerima huruf a sampai z saja. Kunci 2 yang mengandung huruf besar atau angka serta simbol lain akan dihapus secara otomatis oleh sistem.

Kunci 2 Awal	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
Aturan Konversi	H	He	Li	Be	B	C	N	O	F	Ne	Na	Mg	Al	Si	P	S	Cl	Ar	K	Ca	Sc	Ti	V	Cr	Mn	Fe

Gbr. 5 Aturan Konversi Kunci 2

D. Proses Enkripsi Transposisi 5

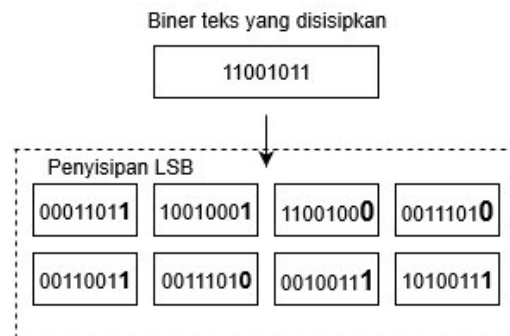
Proses selanjutnya setelah hill cipher adalah proses transposisi 5, dimana setiap kunci dibagi menjadi 5 kolom [1 2 3 4 5]. Proses transposisi ini membutuhkan hasil proses hill cipher kelipatan 5, sehingga apabila jumlah hasil proses hill tidak kelipatan 5 maka akan ditambahkan variabel atau string lain sehingga mendapatkan jumlah hasil proses hill cipher kelipatan 5. Proses transposition dilakukan dengan mengubah urutan dari 5 kolom. Ilustrasi proses transposisi 5 terlihat pada gambar 6.

Plaintext	1	2	3	4	5	1	2	3	4	5
Ciphertext Akhir	2	4	1	3	5	2	4	1	3	5

Gbr. 6 Ilustrasi Proses Transposisi 5

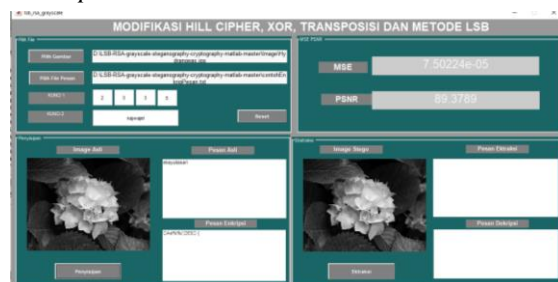
E. Proses Penyembunyian Pesan dengan LSB

Pada proses penyisipan pesan teks kedalam image, sebelumnya pesan teks dikonversi kedalam bentuk biner. Hasil konversi tersebut akan disisipkan ke biner pixel citra. Citra yang akan menjadi wadah penyisipan pesan, sebelumnya diubah kedalam gray scale image. Ilustrasi proses penyembunyian pesan dengan LSB terlihat pada gambar 7 berikut.



Gbr. 7 Ilustrasi Penyembunyian Bit Pesan pada Bit Image

F. Tampilan Sistem



Gbr. 8 Tampilan Sistem

Pada gambar 8 merupakan tampilan sistem kriptografi dan steganografi modifikasi hill cipher dengan transposisi dan operasi xor dan LSB sebagai algoritma penyembunyian pesan. User diminta untuk memilih citra dan memilih dokumen .txt yang akan dilakukan enkripsi dan penyisipan kedalam image. Kemudian user menginputkan kunci1 dan kunci2. Pada sistem yang telah dibuat, user dapat mengenyisipkan dan juga mengekstrak pesan text. Hasil image yang telah disisipi pesan teks diukur menggunakan MSE dan PSNR. MSE mengukur tingkat kesalahan kuadrat rata-rata dari perubahan citra baru dari citra sebelum disisipi [11]. Tingkat kemiripan citra baru dengan citra yang disisipi dapat diukur dengan PSNR, dimana semakin besar nilai parameter PSNR dari citra baru maka semakin mirip dengan citra sebelum disisipi. PSNR adalah alat ukur untuk mengetahui ketahanan pada steganografi dengan mengukur nilai perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang mempengaruhi sinyal tersebut [12]. PSNR atau *peak signal to noise ratio* di manfaatkan untuk menghitung kualitas image yang sudah disisipi pesan dengan mengandalkan nilai standar sistem visual manusia (*human visual system*) dengan nilai 30gB [13]. Nilai PSNR pada gambar yang disisipi pesan jika lebih dari 30gB menyiratkan bahwa pesan yang disisipi tidak terlihat oleh mata manusia [14]

Perhitungan MSE dapat terlihat pada persamaan berikut (4):

$$MSE = \frac{1}{m \times n} \sum_n^m \sum_j^n ||I(i, j) - K(i, j)||^2$$

Keterangan :

MSE = Nilai *Mean Square Error* dari citra
m = Panjang Citra (dalam piksel)
n = Lebar Citra (dalam piksel)
(i,j) = Koordinat masing-masing piksel
I = Nilai intensitas citra asli
K = Nilai intensitas citra yang disisipi.

Sementara rumus perhitungan PSNR pada persamaan berikut (5):

$$PSNR = 20 \cdot \log \left(\frac{MAX_i}{\sqrt{MSE}} \right)$$

Ketarangan :

PSNR = Nilai PSNR citra (dalam dB)
MAX i = Nilai Maksimum Piksel
MSE = Nilai MSE

IV. HASIL DAN PEMBAHASAN

A. Source Code Proses Enkripsi

Sistem modifikasi hill cipher dengan transposisi, operasi xor dan algoritma penyisipan pesan LSB yang diusulkan, dibangun menggunakan MATLAB R2018a. Pada bab ini, akan ditampilkan source code hasil enkripsi dan dekripsi algoritma yang diusulkan. Kemudian dilakukan pengujian dengan pesan yang memiliki panjang tertentu serta image yang disisipi berekstensi .jpg dan .bmp. Source code file enkripsi pesan text disimpan sebagai function dengan nama file enkripMod.m yang akan di panggil di file tampilan GUI

sistem yang sudah dibuat. Berikut source code pada proses enkripsi dengan algoritma yang diusulkan.

```
function[hasil_akhir_cipher,pjg_plaintext]=en
kripMod(pesan,kunci1,kunci2);
kunci1= kunci1
A= pesan+0;
[~,pjg_plaintext]=size(A);
A= A(find(~isspace(A)));
lowA = lower(A);
[~,j]=size(lowA);
if mod(j,2) == 0
    leng=j;
elseif mod(j,2) >0
    leng=j+1;
    lowA(1,leng)="x";
    j=j+1;
end
%bg=leng/2;
bg=leng/2;
l=0;
%%proses transposisi-----
for k=1:bg
    l=l+1;
    trans_awal(1:l+1)=[lowA(1,l+1);lowA(1,l)];
    l=l+1;
end
trans_awal=double(trans_awal);
[~,lk]=size(trans_awal);
for i=0:lk-1
    i=i+1
    if trans_awal(1,i)>=97
        konvert(1,i)=trans_awal(1,i)-97;
    elseif trans_awal(1,i)<=97
        konvert(1,i)=trans_awal(1,i)-22;
    end
end
kunci1=kunci1+0; %kunci Hill Cipher
pb=lk/2;
M= reshape(konvert,2,pb);
E=kunci1*M;
hasil_hill=mod(E,36);
hsl_enkrip_h= reshape(hasil_hill,1,lk);
for i=0:lk-1
    i=i+1
    if hsl_enkrip_h(1,i)>=27
        konvert_hsl(1,i)=hsl_enkrip_h(1,i)+22;
    elseif hsl_enkrip_h(1,i)<=26
        konvert_hsl(1,i)=hsl_enkrip_h(1,i)+97;
    end
end
%hasil dekripsi hill cipher di XOR dengan
binary
[~,ka]=size(konvert_hsl);
kh_dob_new=double(konvert_hsl);
kh_biner_new=de2bi(kh_dob_new);
[kf,kb]=size(kh_biner_new);
kunci2=kunci2+0;
kunci2= kunci2(find(~isspace(kunci2)));
%hilkunci2_konversin spasi
kunci2 = lower(kunci2); % lower case
kunci2=double(kunci2)
[~,lz]=size(kunci2);
for i=0:lz-1
    i=i+1;
    if kunci2(1,i)==97
        kunci2_konversi(1,i)="H";
    elseif kunci2(1,i)==98
        kunci2_konversi(1,i)="He";
    elseif kunci2(1,i)==99
        kunci2_konversi(1,i)="Ci";
    elseif kunci2(1,i)==100
        kunci2_konversi(1,i)="Be";
    elseif kunci2(1,i)==101
        kunci2_konversi(1,i)="B";
```

```

elseif kunci2(1,i)==102
    kunci2_konversi(1,i)="C";
elseif kunci2(1,i)==103
    kunci2_konversi(1,i)="N";
elseif kunci2(1,i)==104
    kunci2_konversi(1,i)="O";
elseif kunci2(1,i)==105
    kunci2_konversi(1,i)="F";
elseif kunci2(1,i)==106
    kunci2_konversi(1,i)="Ne";
elseif kunci2(1,i)==107
    kunci2_konversi(1,i)="Na";
elseif kunci2(1,i)==108
    kunci2_konversi(1,i)="Mg";
elseif kunci2(1,i)==109
    kunci2_konversi(1,i)="Al";
elseif kunci2(1,i)==110
    kunci2_konversi(1,i)="Si";
elseif kunci2(1,i)==111
    kunci2_konversi(1,i)="P";
elseif kunci2(1,i)==112
    kunci2_konversi(1,i)="S";
elseif kunci2(1,i)==113
    kunci2_konversi(1,i)="Cl";
elseif kunci2(1,i)==114
    kunci2_konversi(1,i)="Ar";
elseif kunci2(1,i)==115
    kunci2_konversi(1,i)="K";
elseif kunci2(1,i)==116
    kunci2_konversi(1,i)="Ca";
elseif kunci2(1,i)==117
    kunci2_konversi(1,i)="Sc";
elseif kunci2(1,i)==118
    kunci2_konversi(1,i)="Ti";
elseif kunci2(1,i)==119
    kunci2_konversi(1,i)="V";
elseif kunci2(1,i)==120
    kunci2_konversi(1,i)="Cr";
elseif kunci2(1,i)==121
    kunci2_konversi(1,i)="Mn";
elseif kunci2(1,i)==122
    kunci2_konversi(1,i)="Fe";
end
end
konversi=char(kunci2_konversi);
kunci2_new=double(konversi);
key_biner=de2bi(kunci2_new);
[kuy,ky]=size(key_biner);
jpg=ceil(kf/kuy);
p=0;
for i=0:pjpg-1
    key_baru([p+1:p+kuy, :])=[key_biner];
    p=p+1;
    p=p+kuy;
end
for i=0 : kf-1
    i=i+1;
    key_baru_bgt(i, :) = key_baru(i, :);
end
hasil_xor=bitxor(kh_biner_new, key_baru_bgt);
hasil_xor_new=bi2de(hasil_xor);
hasil_xor_new=hasil_xor_new';
%%convert ke abcd%%convert ke abcd
[~, phx]=size(hasil_xor_new);
hmod=mod(phx, 5);
hmod=5-hmod;
if mod(phx, 5)>=0
    for a=1:hmod
        hasil_xor_new(phx+a)=double('x');
    end
elseif mod(phx, 5)==0
    hasil_xor_new=hasil_xor_new;
end
tp=[2 4 1 3 5];
ltp=numel(tp);

```

```

leng_hXor = numel(hasil_xor_new);
chipertext=char(zeros(1,leng_hXor));
plaintext = [hasil_xor_new
32*ones(1,ltp*ceil(leng_hXor/ltp) -
leng_hXor)];
for idx = 1 : ltp : leng_hXor
    tx = plaintext(idx : idx+ltp-1);
    ciphertext2(idx : idx+ltp-1) = tx(tp);
end
hasil_akhir_cipher=char(ciphertext2);
setappdata(0, 'hasilFunction', hasil_akhir_cipher);

```

B. Source Code Proses Dekripsi

Source code proses dekripsi disimpan pada file function dekripMod.m yang dipanggil kedalam file tampilan GUI sistem. Berikut source code dengan matlab untuk dekripsi menggunakan algoritma yang diusulkan :

```

function[hs1_akhir_dek]=dekripMod(pesan_enkrip, kunci1, kunci2, jpg_plaintext);
cp_text=pesan_enkrip+0;
cp_text1=double(cp_text);
[~,ca]=size(cp_text1);
tp=[3 1 4 2 5];
ltp=numel(tp);
leng_hXor = numel(cp_text1);
dekriptext=char(zeros(1,leng_hXor));
dekriptext1 = [cp_text1
32*ones(1,ltp*ceil(leng_hXor/ltp) -
leng_hXor)];
for idx = 1 : ltp : leng_hXor
    tx = dekriptext1(idx : idx+ltp-1);
    dekriptext1(idx : idx+ltp-1) = tx(tp);
end
%%hasil enkripsi
end
hasil_dekrip_ulang=dekriptext1;
kunci2=kunci2+0;
kunci2= kunci2(find(~isspace(kunci2)));
kunci2 = lower(kunci2); % lower case
kunci2(regexp(kunci2, '[^\,]*$@#&!~\`-;+_=?><|')=[])
[~,lz]=size(kunci2);
for i=0:lz-1
    i=i+1;
    if kunci2(1,i)=="a"
        kunci2_konversi(1,i)=="H";
    elseif kunci2(1,i)=="b"
        kunci2_konversi(1,i)=="He";
    elseif kunci2(1,i)=="c"
        kunci2_konversi(1,i)=="Ci";
    elseif kunci2(1,i)=="d"
        kunci2_konversi(1,i)=="Be";
    elseif kunci2(1,i)=="e"
        kunci2_konversi(1,i)=="B";
    elseif kunci2(1,i)=="f"
        kunci2_konversi(1,i)=="C";
    elseif kunci2(1,i)=="g"
        kunci2_konversi(1,i)=="N";
    elseif kunci2(1,i)=="h"
        kunci2_konversi(1,i)=="O";
    elseif kunci2(1,i)=="i"
        kunci2_konversi(1,i)=="F";
    elseif kunci2(1,i)=="j"
        kunci2_konversi(1,i)=="Ne";
    elseif kunci2(1,i)=="k"
        kunci2_konversi(1,i)=="Na";
    elseif kunci2(1,i)=="l"
        kunci2_konversi(1,i)=="Mg";
    elseif kunci2(1,i)=="m"
        kunci2_konversi(1,i)=="Al";

```

```

elseif kunci2(1,i)=="n"
    kunci2_konversi(1,i)="Si";
elseif kunci2(1,i)=="o"
    kunci2_konversi(1,i)="P";
elseif kunci2(1,i)=="p"
    kunci2_konversi(1,i)="S";
elseif kunci2(1,i)=="q"
    kunci2_konversi(1,i)="Cl";
elseif kunci2(1,i)=="r"
    kunci2_konversi(1,i)="Ar";
elseif kunci2(1,i)=="s"
    kunci2_konversi(1,i)="K";
elseif kunci2(1,i)=="t"
    kunci2_konversi(1,i)="Ca";
elseif kunci2(1,i)=="u"
    kunci2_konversi(1,i)="Sc";
elseif kunci2(1,i)=="v"
    kunci2_konversi(1,i)="Ti";
elseif kunci2(1,i)=="w"
    kunci2_konversi(1,i)="V";
elseif kunci2(1,i)=="x"
    kunci2_konversi(1,i)="Cr";
elseif kunci2(1,i)=="y"
    kunci2_konversi(1,i)="Mn";
elseif kunci2(1,i)=="z"
    kunci2_konversi(1,i)="Fe";
end
end
konversi=char(kunci2_konversi);
kunci2_new=double(konversi);
key_biner_des=de2bi(kunci2_new);
[kuy,ky]=size(key_biner_des);
[kb,kf]=size(hasil_dekrip_ulang);
pjpg=ceil(kf/kuy);
p=0;
for i=0:pjpg-1
    key_baru([p+1:p+kuy,:])=[key_biner_des];
    p=p+1;
    p=p+kuy;
end
for i=0 : kf-1
    i=i+1;
    key_baru_bgt_des(i,:) = key_baru(i,:);
end
kh_biner_new_des=de2bi(hasil_dekrip_ulang);
hasil_xor_des=bitxor(kh_biner_new_des,key_baru_bgt_des);
hasil_xor_new_des=bi2de(hasil_xor_des);
hasil_xor_new_des=hasil_xor_new_des';
isi=("x");
isi=char(isi);
[~,jd]=size(hasil_xor_new_des);
if mod(jd,2) == 0
    jd=jd;
elseif mod(jd,2) > 0
    jd=jd+1;
    hasil_xor_new_des(1,jd)=double(isi);
end
[~,lkd]=size(hasil_xor_new_des);
i=0;
for i=0:lkd-1
    i=i+1
if hasil_xor_new_des(1,i)>=97
    konvert_des(1,i)=hasil_xor_new_des(1,i)-97;
elseif hasil_xor_new_des(1,i)<=97
    konvert_des(1,i)=hasil_xor_new_des(1,i)-22;
end
end
end
kuncil=kuncil+0;
a_det=kuncil(1,1);
b_det=kuncil(1,2);
c_det=kuncil(2,1);
d_det=kuncil(2,2);

```

```

det_key=(a_det*d_det)-(c_det*b_det);
modulo=((36*1)+1)/det_key;
kunci_dek_baru=[d_det -b_det;-c_det a_det];
key_dekrip_akhir=mod(modulo*kunci_dek_baru,36);
[~,cb]=size(konvert_des);
if mod(cb,2) == 0
    cb=cb;
elseif mod(cb,2) > 0
    cb=cb+1;
    konvert_des(1,cb)=double(isi);
end
[~,cb_new]=size(konvert_des);
f=0;
bg=cb_new/2;
for i=0:bg-1
    f=f+1;
    vek=konvert_des(f:f+1);
    E=vek*key_dekrip_akhir;
    hsl_dekrip_hill(f:f+1)=mod(E,36);
    f=f+1;
end
%%konvert hasil dekrip hill
for i=0:lkd-1
    i=i+1
if hsl_dekrip_hill(1,i)>=27
    konvert_hsl_dekrip(1,i)=hsl_dekrip_hill(1,i)+22;
elseif hsl_dekrip_hill(1,i)<=26
    konvert_hsl_dekrip(1,i)=hsl_dekrip_hill(1,i)+97;
end
end
cg=cb/2;
%%proses dekrip transposisi-----
j=0;
for k=1:cg
    j=j+1;
hsl_akhir_dekrip(j:j+1)=[konvert_hsl_dekrip(1,j+1);konvert_hsl_dekrip(1,j)];
j=j+1;
end
%%%-----
%%%-----
hsl_akhir_dek=char(hsl_akhir_dekrip(1:pjpg_plaintext));

```

C. Pengujian Input dan Output Sistem

Pengujian ini dilakukan untuk membuktikan kerja sistem apakah sudah sesuai dengan yang diinginkan. Kesesuaian kerja sistem terlihat ketika hasil pesan asli dan hasil dekripsi ekstraksi image stego sama. Pengujian ini menggunakan file berekstensi .jpg dengan ukuran 1024x768. Karakter yang diuji adalah karakter a sampai b dan 0 sampai 9 dengan kunci1 $\begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$ dan kunci2 "bisadong". Hasil dari pengujian ini terlihat pada tabel 1 berikut :

TABEL I
HASIL PENGUJIAN SISTEM

	Pesan	Hasil Enkripsi	Dekripsi
1	Abcdefghijkl mnopqrstuvwxyz xyz	T++<~%PPLX !(/kH0"?qEOx x2x	abcdefghijklmnop qrstuvwxyz xyz
2	0123456789	PL1~":pHxxxx x	123456789

3	abcdefgh12@ ..	T++<~qPPd6x	abcdefgh12g yw
---	-------------------	-------------	-------------------





Dari tabel diatas, terlihat bahwa dekripsi citra stego dengan pesan yang disipkan kedalam image hasilnya sama. Ini membuktikan bahwa algoritma yang diusulkan dapat digunakan untuk mengamankan pesan digital dengan mengenkripsi dan menyisipkannya kedalam sebuah gambar. Karakter selain a samapai z dan 0 sampai 9 tidak dapat dikenali karena tujuan algoritma ini hanya dapat mengenkripsi karakter huruf dan angka saja.

D. Pengukuran MSE dan PSNR

Pengukuran selanjutnya dilakukan dengan menghitung MSE dan PSNR dari *image* yang telah disisipi pesan teks. Pada tahap pengujian ini, digunakan citra dengan ukuran berbeda serta ekstensi yang digunakan yaitu .jpg dan .bmp saja. Awal proses penyembunyian pesan kedalam citra adalah citra yang digunakan sebagai wadah untuk menyembunyikan pesan terlebih dahulu diubah menjadi citra grayscale. Teknik penyembunyian pesan yang digunakan adalah LSB penggantian yang pertama kali dibahas oleh T. Sharp [15] dimana bit terakhir dari citra diganti dengan setiap bit dari pesan teks yang akan disembunyikan. Penyembunyian pesan kedalam image akan lebih mengurangi kecurigaan terhadap pesan yang dienkripsi karena pesan yang sudah *dihidding* kedalam sebuah citra tidak terlihat perbedaannya secara signifikan oleh mata manusia.

Hasil dari pengujian tahap ini terlihat pada tabel 2 berikut ini :

TABEL II
HASIL EKSPERIMEN PENYISIPAN DAN EKSTRAKSI

Citra	Time Hidding	Time Ekstraksi	MSE	PSNR
 1024 x 768 Koala.jpg	9,592 s	385,41s	0.00498	71,15
 300 x 168 Cat5.jpg	11,85s	2,456 s	0.07311	59.49
 500x480 Baboon.bmp	2,514 s	9,163 s	0.0160	66.08
 320x240 Tiger.bmp	0,472s	1,5705 s	0,050	61.09

Tabel 2 merupakan hasil eksperimen penyisipan pesan berjumlah 975 yang terdiri dari huruf dan angka.

Citra yang digunakan berbeda ukuran dan ekstensinya. Terlihat bahwa citra dengan ekstensi .jpg dengan ukuran 1024x768 mendapatkan nilai MSE paling kecil dan PSNR terbesar. Namun proses ekstraksi citra Koala.jpg membutuhkan waktu proses paling lama. Waktu proses dipengaruhi oleh ukuran citra.

V. KESIMPULAN DAN SARAN

Penelitian ini mengusulkan metode modifikasi hill cipher dengan transposisi dan operasi XOR yang digabungkan dengan metode steganografi LSB. Hill cipher dimodifikasi dengan merubah mod yang digunakan yaitu 36. Pemanfaatan tabel periodik pada pengamanan kunci2 dapat digunakan dan mudah diimplementasikan, karena hanya mengkonversi huruf kedalam kode unsur senyawa kimia pada tabel periodik. Enkripsi pesan teks melalui 5 tahap keamanan yaitu pergeseran, hill cipher, operasi xor, enkripsi kunci 2 untuk operasi XOR, dan transposisi 5. Pesan teks yang terenkripsi selanjutnya disisipkan kedalam bit-bit terakhir atau bit paling rendah pada citra. Sehingga total pengamanan pada algoritma yang diusulkan berjumlah 6 tahap pengamanan. Hasil yang diperoleh pada penelitian ini adalah :

- Modifikasi dengan algoritma kriptografi dan steganografi yang diusulkan yaitu modifikasi hill cipher, pergeseran, operasi xor, transposisi dan penyisipan pesan teknik LSB dapat digunakan untuk melindungi pesan rahasia.
- Hill cipher modulus 36 dapat dimanfaatkan untuk mengenkripsi karakter selain huruf yaitu 0 sampai 9.
- Citra dengan ekstensi .jpg dan ukuran 1024x768 jika digunakan sebagai wadah penyisipan pesan, menghasilkan nilai PSNR tertinggi yaitu sebesar 71,15 dan nilai MSE paling rendah yaitu sebesar 0,00498.

Saran untuk penelitian selanjutnya adalah :

- Modifikasi algoritma yang diusulkan dapat disisipkan kedalam citra dengan teknik yang berbeda.
- Mod pada hill cipher dapat ditambah lagi sehingga lebih banyak karakter yang dapat dienkripsi, seperti .(titik), : (titik dua), ;(titik koma), dan ,(koma). Sehingga dapat mengenkripsi dokumen pesan yang lebih panjang.

REFERENSI

- H. F. M. I. Adi W., "Teknik Keamanan Data Menggunakan Vigenere Cipher dan Electronic Code Book (ECB)," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 3 No. 2, pp. 393-400, 2019.
- A. Dony, *Pengantar Kriptografi : Teori, Analisis dan Implementasi*, Yogyakarta: Andi Offset, , 2008.
- J. e. a. Katz, *Buku pegangan kriptografi terapan*, Pers.CRC., 1996.

- [4] C. d. J. P. Paar, Memahami kriptografi:buku teks untuk siswa dan praktisi, Springer Science & Business Media,, 2009.
- [5] J. &. D. M. Chase, Extending the Hill Cipher., (2010).
- [6] S. H. T. S. Jane Irma Sari., "Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma Hill Cipher dan Metode Least Significant Bit (LSB)," *Jurnal Mantik Penusa*, Vol. 21, No. 2, pp.1-8, 2017.
- [7] A. B. H. Tuti Alawiyah, "Modifikasi Kriptografi Hill Cipher Kunci Matriks Persegi Panjang Menggunakan Fungsi Xor Dan Fungsi Xnor," *Indonesian Journal on Computer and Information Technology*, vol. 1, pp.68-82, 2016.
- [8] P. H. D. A. Selviana Yunita., "Modifikasi Algoritma Hill Cipher dan Twofish Menggunakan Kode Wilayah Telepon," *Jurnal Ilmiah SISFOTENIKA*, Vol. 29, No.2, pp. 213-224, 2019.
- [9] E. Z. e. al, "Flipping the Message Bits to Increase Imperceptibility in the Least Significant Bit Image Steganography," *International Conference on Electronics Representation and Algorithm (ICERA 2019)*, 2019.
- [10] A. K. Marsela Sutikno Dibiyo., "Implementasi Vernam Cipher dan Steganografi End Of File (EOF) Untuk Enkripsi Pesan PDF," *Techno.COM*, vol. 15, pp.66-71, 2016.
- [11] R. Wissarto, "Implementasi Slantlet Transform (SLT) Dan Huffman Coding Pada Steganografi Citra Grayscale," 2014. [Online]. Available: http://eprints.dinus.ac.id/13071/1/jurnal_13435.pdf. [Använd 08 Nov 2019].
- [12] K. Aisyatul, "Pengkutan Tingkat Ketahanan (Robustness) Metode LSB Terhadap Perubahan Kontras Pada Steganografi," Udinus Repo, Semarang,Indonesia, 2008.
- [13] S. H. L. &. Y. S. Shen, "A novel adaptive data hiding based on improved EMD and interpolation," *Multimed Tools Appl (2018)*, 02 June 2017.
- [14] C.-C.-C. Wen-Chung Kuo., "Binary power data hiding scheme," *AEU - International Journal of Electronics and Communications*, Vol. 69, No11, pp. 1574-1581, 2015.
- [15] T. Sharp, "An Implementation of Key-Based Digital Signal Steganography," i *International Workshop on Information Hiding*, Berlin,Heidelberg, 2001.