Jurnal Syntax Admiration Vol. 3 No. 2 Februari 2022
p-ISSN : 2722-7782 e-ISSN : 2722-5356 Sosial Teknik

PENGAMANAN DATA DENGAN KRIPTOGRAFI *HIBRIDA ALGORITMA HILL CIPHER DAN ALGORITMA LUC SERTA STEGANOGRAFI CHAOTIC LSB*

Novita Permata Dewi, David J.M Sembiring, Raheliya br. Ginting, Meiliyani Br Ginting
Institut Teknologi dan Bisnis Indonesia
Email: dewinovitapermata@gmail.com, davidjms366@gmail.com,
raheliyabrginting@gmail.com, meiliyani.ginting@gmail.com

INFO ARTIKEL
Diterima
25 Januari 2022
Direvisi
05 Februari 2022
Disetujui
15 Februari 2022

Kata Kunci:
kriptografi, steganografi; luc; hill cipher; chaotic LSB

ABSTRAK

Data yang bersifat rahasia biasanya memuat informasi penting, memiliki nilai yang tinggi dan tidak seharusnya diketahui oleh sembarang pihak, seperti data tentang bisnis, politik dan lainnya. Hal ini memicu munculnya usaha-usaha untuk mengetahui keberadaan dan isi informasi dari pihak yang tidak bertanggungjawab. Pada penelitian ini dilakukan pengamanan data dengan teknik kritografi hibrida dengan mengkombinasikan algoritma Hill Cipher dan LUC. Kunci Algoritma Hill Cipher akan diamankan oleh Algoritma LUC. Pesan asli (plaintext) dienkripsi dengan Hill Cipher. Sedangkan untuk pendistribusian kunci menggunakan algortima LUC dan disisipkan dalam media gambar dengan Chaotic LSB.

Keywords:
kriptografi; steganography; luc; hill cipher; chaotic LSB

ABSTRACT

Confidential data usually contains important information, has a high value and should not be known by just anyone, such as data about business, politics and others. This triggers the emergence of efforts to find out the existence and content of information from irresponsible parties. So, this research, data security was secured using a hybrid critography technique by combining the Hill Cipher and LUC algorithms. The Hill Cipher Algorithm key will be secured by the LUC Algorithm. The original message(plaintext) is encrypted with Hill Cipher. As for the distribution of keys using the LUC algorithm and inserted in the image media with Chaotic LSB.

Pendahuluan

Data yang bersifat rahasia biasanya memuat informasi penting, memiliki nilai yang tinggi dan tidak seharusnya diketahui oleh sembarang pihak, seperti data tentang bisnis, politik dan lainnya. Hal ini memicu munculnya usaha-usaha untuk mengetahui keberadaan dan isi informasi dari pihak yang tidak bertanggungjawab. Terlebih dengan

How to cite: Dewi, N, P., David J.M Sembiring, Raheliya br. Ginting & Meiliyani Br Ginting (2022) Pengamanan Data dengan Kriptografi Hibrida Algoritma Hill Cipher dan Algoritma Luc Serta Steganografi Chaotic Lsb, *Jurnal Syntax Admiration* 3(2). https://doi.org/10.46799/jsa.v3i2.389
E-ISSN: 2722-5356
Published by: Ridwan Institute

perkembangan teknologi sekarang, berbagai macam tindakan kejahatan komputer dapat dilakukan melalui teknologi seperti penyadapan, pencurian ataupun perusakan data.

Pencurian dan penyadapa dapat menyebabkan data dirusak, diubah, bahkan disebar secara luas untuk kepentingan pribadi yang merugikan orang lain. Oleh karena itu perlu adanya teknik untuk mengamankan data dari pengubahan maupun akses yang tidak sah. Teknik yang dapat digunakan untuk mengamankan data tersebut adalah kriptografi. Dengan kriptografi pesan asli akan diamankan dengan menggunakan teknik penyandian yang menghasilkan pesan yang tersandikan. Sehingga, pihak ketiga akan kesulitan menerjemahkan isi pesan. Walaupun demikian, hasil dari kriptografi adalah pesan acak yang tentu saja akan menimbulkan kecurigaan dari pihak ketiga, untuk mengoptimalkan pengamanan data, kriptografi dikombinasikan dengan steganografi.

Steganografi adalah teknik penyembunyian pesan kedalam suatu media, dan pada penelitian ini digunakan media citra (gambar). Tetapi, berdasarkan penelitian sebelumnya yang dilakukan oleh (Anggraini, 2014) pengamanan data yang dilakukan dengan mengkombinasikan kriptografi RSA dan steganografi Chaotic LSB, walaupun tidak secara kasat mata, tampak ada perbedaan antara citra asli dan hasil stego-image dalam penelitiannya, padahal karakter yang disembunyikan ke dalam gambar hanya sedikit. Sedangkan, data yang bersifat penting atau rahasia biasanya memuat karakter panjang, yang mungkin saja jika dokumen tersebut disisipkan ke dalam suatu media, maka hasil dari stego-image akan berbeda jauh dari citra asli.

Oleh karena itu, untuk pengoptimalan pengamanan data, teknik steganografi dikombinasikan dengan teknik kriptografi, khususnya kriptografi hibrida. Keunggulan dari kriptografi hibrida adalah dengan memanfaatkan keunggulan kecepatan proses data dengan kriptografi simetris dan pengamanan kunci dengan kriptografi asimetris (Ariyus, 2008).

Kriptografi simetris yang digunakan dalam penelitian ini adalah Hill Cipher. Berdasarkan penelitian sebelumnya yang dilakukan oleh (Laoli et al., 2020), disimpulkan bahwa algortima Hill Cipher merupakan salah satu kriptogarfi simetri yang sulit dipecahkan. Hal ini disebabkan karena Hill Cipher menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya, sehingga tidak mengganti setiap huruf yang sama pada plaintext dengan huruf lainnya yang sama pada ciphertext.

Berdasarkan penelitan yang dilakukan (Rahman, J.B., Muhsin., Nurhayati, 2018), Algoritma LUC salah satu kriptografi asimetris yang aman, sama halnya dengan algoritma RSA, dan proses enkripsi data juga cukup cepat karena tidak menggunakan bilangan berpangkat yang bernilai besar**.** Sebelum Algoritma LUC melakukan enkripsi, setiap karakter string dari *plaintext* yang dimasukkan akan dikonversi ke dalam bentuk bilangan. Kunci Hill Cipher yang digunakan merupakan matriks berukuran 3x3, maka kunci simetris tidak perlu melakukan proses konversi bilangan untuk dapat dienkripsi oleh algoritma LUC.

Pada penelitian ini digunakan kombinasi kriptografi hibrida dan steganografi. Pesan asli (*plaintext*) dienkripsi dengan Hill Cipher. Sedangkan untuk pendistribusian

kunci menggunakan algortima LUC dan disisipkan dalam media gambar dengan Chaotic LSB.

Berdasarkan uraian diatas maka penelitian ini berjudul "Pengamanan Data Dengan Kriptografi Hibrida Algoritma Hill Cipher dan Algoritma LUC serta Steganografi Chaotic LSB".

Metode Penelitian

1. Kriptografi

Kriptografi berasal dari bahasa Yunani yakni *crypto* yang artinya *secret* (rahasia),dan *graphia* yang artinya *writing* (tulisan). Menurut terminologinya, kriptogarfi adalah ilmu dan seni untuk menjaga pesan ketika pesan dikirim dari suatu tempat ke termpat lain (Ariyus, 2008).

Pesan asli akan mengalami proses penyandian menjadi pesan yang tersamarkan atau pesan acak yang dikenal dengan istilah *ciphertext.* Proses transformasi *plaintext* menjadi *ciphertext* disebut proses enkripsi atau *enciphering.* Sedangkan, proses sebaliknya, mengubah pesan yang tersamarkan menjadi pesan asli kembali disebut proses dekripsi atau *deciphering* (Mollin, 2007).

2. Algoritma Hill Cipher

Salah satu algoritma kriptografi simetris adalah Hill Cipher, yang dikembangkan oleh Lester S. Hill. Hill Cipher menggunakan *m* buah persamaan linear. Dimana *m* adalah *block* yang akan dienkripsikan (Stallings, 2005).

Hill Cipher menggunakan matriks persegi sebagai kunci yang digunakan untuk melakukan enkripsi dan dekripsi. Teknik kriptografi ini 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 tidak mengganti setiap abjad yang sama pada *plaintext* dengan abjad lainnya yang sama pada *ciphertext* karena menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya (Laoli et al., 2020).

Teknik kriptografi ini dipecahkan hanya dengan mengetahui berkasa *ciphertext* saja, tetapi jika mempunyai sebagian berkas *ciphertext* dan *plaintext* akan sangat mudah dipecahkan oleh kriptanalis. Semakin besar ukuran matriks kunci maka semakin sulit Hill Cipher dipecahkan.

3. Teknik Enkripsi Hill Cipher

Proses enkripsi pada *Hill Cipher* dilakukan per blok *plaintext*. Ukuran blok tersebut sama dengan ukuran matriks kunci. Pada penelitian yang dilakukan (Serdano, Akbar., Zarlis, Muhammad., Sawaluddin., Hartama, 2019) Sebelum membagi teks menjadi deretan blok-blok, *plaintext* terlebih dahulu dikonversi menjadi angka, masing-masing sehingga A=0, B=1, hingga Z=25.

Tetapi untuk penelitian ini, *plaintext* dikonversi sesuai tabel kode ASCII. Sama halnya dengan *ciphertext*, hasil *ciphertext* yang akan ditampilkan juga berdasarkan

table ASCII dengan rentang 32 – 127. Secara matematis, proses enkripsi pada *Hill Cipher* adalah:

$C = K \cdot P$

C = *Ciphertext*

K = Kunci

P = *Plaintext*

4. Implementasi Enkripsi Hill Cipher

Pesan yang digunakan adalah "Rahasia" dengan kunci (*K*)

$$K = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 5 & 0 \\ 2 & 0 & 6 \end{bmatrix}$$

Matriks yang menjadi kunci untuk Algoritma Hill Cipher harus merupakan matriks yang *invertible*, yaitu memiliki *multiplicative inverse* K^{-1}, sehingga : $K \cdot K^{-1} = 1$. Invers matriks kunci akan digunakan dalam proses dekripsi.

Implementasi Proses enkripsinya adalah :

a) Pesan atau *plaintext* diubah ke bentuk numerik, dalam penelitian ini numerik yang digunakan berdasarkan table.

ASCII						
R	A	H	A	s	I	a
82	97	104	97	115	105	97

b) Karena banyaknya abjad dalam plaintext yaitu 7 bukan kelipatan dari ukuran kolom matriks kunci yaitu 3 maka tambahkan sembarang abjad dalam plaintext sehingga *k* menjadi kelipatan m. Dalam Implementasi ini ditambahkan abjad X dan Q (huruf kapital).

c) Buatlah plaintext dalam bentuk blok dengan ukuran blok sama dengan ukuran kolom matriks kunci yaitu 3, sehingga plaintext menjadi:

$$P = \begin{bmatrix} R & a & h \\ a & s & i \\ a & X & Q \end{bmatrix}$$

d) Buat P transpose :

$$P = \begin{bmatrix} R & a & a \\ a & s & X \\ h & i & Q \end{bmatrix}$$

e) Korespondensikan hasil dengan numerik, sehingga diperoleh :

$$P = \begin{bmatrix} 82 & 97 & 97 \\ 97 & 115 & 88 \\ 104 & 105 & 81 \end{bmatrix}$$

f) Kalikan matriks kunci K dengan plaintext transpose dalam modulo 256 berikut:

$$C = K \,.\, P$$

$$C = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 5 & 0 \\ 2 & 0 & 6 \end{bmatrix} \begin{bmatrix} 82 & 97 & 97 \\ 97 & 115 & 88 \\ 104 & 105 & 81 \end{bmatrix}$$

$$C = \begin{bmatrix} 164 & 194 & 194 \\ 567 & 672 & 537 \\ 788 & 824 & 728 \end{bmatrix} (\text{mod } 256)$$

$$C = \begin{bmatrix} 37 & 67 & 67 \\ 59 & 37 & 29 \\ 26 & 62 & 93 \end{bmatrix}$$

g) Ubah hasil step f ke dalam string menggunakan tabel ASCII sehingga diperoleh *ciphertext.*

$$C = \begin{bmatrix} 37 & 67 & 67 \\ 59 & 37 & 61 \\ 58 & 62 & 93 \end{bmatrix}$$

$$C = \begin{bmatrix} \% & ; & : \\ C & \% & > \\ C & = &] \end{bmatrix}$$

h) Diperoleh *ciphertext* :

$$C = \begin{bmatrix} \% & ; & : \\ C & \% & > \\ C & = &] \end{bmatrix}$$

i) Ciphertext C = “% ; : C % > C =]”

5. Teknik Dekripsi Hill Cipher

Proses dekripsi pada *Hill Cipher* pada dasarnya sama dengan proses enkripsinya. Namun matriks kunci harus dibalik (invers) terlebih dahulu.

Secara matematis, proses persamaan deskripsi :

$$P = K^{-1} \,.\, C$$

6. Implementasi Dekripsi Hill Cipher

Secara Umum Tahap-Tahap Dekripsi Hill Cipher Adalah :

a) Matriks kunci : $K = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 5 & 0 \\ 2 & 0 & 6 \end{bmatrix}$

b) Mencari invers matriks K, dengan cara :

- Mencari adjoint matriks K, diperoleh :

- $Adj\, K = \begin{bmatrix} 30 & 0 & 0 \\ -6 & 12 & 0 \\ -10 & 0 & 10 \end{bmatrix}$

- Mencari determinan matrik *K :*

$|K| = (60 + 0 + 0) – (0 + 0 + 0)$

$|K| = 60$

$|K| = 60 \bmod 127$

$|K| = 60$

- Mencari inverse dari determinan matriks *K* dimodulus 256 (*x*) :

$$\frac{1}{|K|} \; mod\, 127 = x$$

$$(x\,.\,|\,K\,|)\; mod\, 127 \;= 1$$

$$(x.60)\; mod\, 127 = 1 \;, x = 36$$

- Mencari Invers matriks K :

$K^{-1} = (x\,.\,Adj\ K)\ mod\ 256$

$$K^{-1} = \begin{bmatrix} 30 & 0 & 0 \\ -6 & 12 & 0 \\ -10 & 0 & 10 \end{bmatrix} \quad mod\ 256$$

$$K^{-1} = \begin{bmatrix} 64 & 0 & 0 \\ 38 & 51 & 0 \\ 21 & 0 & 106 \end{bmatrix}$$

c) Mencari *plaintext* transpose

$P = K^{-1}\ C$

$$P = \begin{bmatrix} 64 & 0 & 0 \\ 38 & 51 & 0 \\ 21 & 0 & 106 \end{bmatrix} \begin{bmatrix} 37 & 67 & 67 \\ 59 & 37 & 61 \\ 58 & 62 & 93 \end{bmatrix}$$

$$P = \begin{bmatrix} 2368 & 4288 & 4288 \\ 4415 & 4433 & 4025 \\ 3533 & 7979 & 11265 \end{bmatrix} \quad mod256$$

$$P = \begin{bmatrix} 82 & 97 & 97 \\ 97 & 115 & 88 \\ 104 & 105 & 81 \end{bmatrix}$$

d) Dari step c diperoleh :

$$P = \begin{bmatrix} 82 & 97 & 97 \\ 97 & 115 & 88 \\ 104 & 105 & 81 \end{bmatrix}$$

e) Ubah hasil step d ke dalam abjad menggunakan koresponden abjad dengan numerik sehingga diperoleh *plaintext*:

$$P = \begin{pmatrix} R & a & h \\ a & s & i \\ a & X & Q \end{pmatrix}$$

f) Diperoleh *plaintext* : “Rahasia”

7. Algoritma LUC

Algoritma kriptografi LUC merupakan salah satu sistem kriptografi yang menerapkan algoritma kunci publik. Algoritma ini dikemukakan oleh Peter J. Smith dan Michael J.J. Lennon pada tahun 1993 di New Zealand. LUC memiliki iga tahap utama, yaitu Pembangkitan kunci, enkripsi dan dekripsi.

Operasi pada Algoritma LUC dilakukan dalam domain bilangan, oleh karena itu sebelum dilakukan enkripsi, teks terlebih dahulu di konversi kedalam bentuk bilangan. Hasil enkripsi berupa teks yang telah disandikan dalam bentuk bilangan. Kunci dekripsi pada algoritma LUC tergantung pada kunci enkripsi dan dipengaruhi

oleh bilangan prima p dan q, setiap satu kunci enkripsi mempunyai empat kemungkinan kunci dekripsi (Rahman, J.B., Muhsin., Nurhayati, 2018).

8. Pembangkit Kunci Algoritma LUC

Ada dua kunci yang digunakan dalam algoritma LUC, yaitu :

1. Algoritma Kunci Publik
   a) Pilih dua bilangan prima sebarang, misal p dan q dimana $p \neq q$ dan gcd(p,q) = 1
   b) Hitung nilai $N = p$ x q. Nilai N akan digunakan dalam menghitung modulo pada proses enkripsi dan dekripsi
   c) Hitung semua bilangan yang relatif prima terhadap ϕN = (p-1), (p+1), (q-1) dan (q+1)
   d) Pilih salah satu bilangan secara acak dari hasil yang didapatkan pada poin (c) sebagai kunci publik e dimana $e \in Z$, $e < n$-1 dan gcd(e,ϕN) = 1
2. Algoritma Kunci Privat
   a) Masukkan dua bilangan prima p dan q.
   b) Masukkan e yang dihitung pada tahap pembangkitan kunci publik.
   c) Hitung R(N) = lcm((p-1),(p+1),(q-1),(q+1))
   d) Hitung d sehingga mendapatkan hasil $e.d$ mod R(N) = 1

   Nilai(d,N) yang diperoleh merupakan kunci dekripsi (kunci privat) dari kunci enkripsi (e,N). Proses pembangkitan kunci dilakukan dengan rahasia terutama nilai bilangan prima p dan q, serta nilai $S(N)$ yang dipakai untuk dekripsi.

9. Implementasi Pembangkit Kunci Algoritma LUC

1) Kunci Publik
   a) $p = 7$; $q = 11$ dan gcd(p,q) = 1
   b) $N = p \times q$
   = 7 x 11
   N = 77
   c) ϕN = (p-1).(p+1).(q-1).(q+1)
   = (7-1).(7+1).(11-1).(11+1)
   = 6.8.10.12
   ϕN = 5760
   d) Relatif prima (p-1) = R.P 6 = {3, 5}
   e) Relatif prima (q-1) = R.P 10= { 3, 5, 7 }
   f) Relatif prima (p+1) = R.P 8 = {3, 5, 7}
   g) Relatif prima (q+1) = R.P 12 = {3, 5, 7, 11}

   Hasil perhitungan bilangan relatif prima diatas terdapat beberapa bilangan yang sama, yaitu {3, 5, 7, 11}. Maka pilih e = 7, GCD (7, 5760) = 1.
2) Kunci Privat
   a) $p = 7$; $q = 11$
   b) e = 7

c) R(N) = lcm ((p-1),(p+1),(q-1),(q+1))
= lcm(6,8,10,12)
R(N) = 120

d) Tabel 1 merupakan tabel perhitungan untuk mencari nilai d

Tabel 1
Proses Perhitungan Kunci Dekripsi (D)

D	7. d mod120 = 1
1	7 mod 120 = 7
2	14 mod 120 = 14
3	21 mod 120 = 21
103	721 mod 120 = 1

10. Teknik Enkripsi Algoritma LUC

Proses enkripsi adalah proses pengacakan data atau pesan, misalkan A akan bertukar informasi dengan B, pihak A dan B sama-sama melakukan pembangkitan kunci seperti yang telah dijelaskan pada sub bab sebelumnya, kemudian A dan B bertukar kunci publik (A menerima kunci publik dari B dan B menerima kunci publik dari A) dimana pertukaran kunci tersebut tidak bersifat rahasia. Dalam proses enkripsi dimisalkan B ingin mengirim data atau pesan kepada A, maka B terlebih dahulu harus mempunyai kunci publik (e) yang diberikan oleh A.

Selanjutnya proses enkripsi dapat dijelaskan sebagai berikut (Rahman, J.B., Muhsin., Nurhayati, 2018):

1) *Plaintext* (M) adalah isi pesan atau informasi yang akan disampaikan oleh B kepada A.
2) Nilai e dan N didapatkan dari kunci publik yang telah diberikan A kepada B.
3) *Plaintext* (M) yang akan disampaikan kepada A dipecah atau diatur menjadi blok-blok m_1, m_2, …m_i yang mempunyai satu karakter pada tiap blok.
4) Setiap blok yang telah didapatkan (m_i) di ubah dalam bentuk ASCII kemudian di enkripsi dengan persamaan V[e] = (mi . V [i – 1] – V [i – 2]) mod N. *Ciphertext*(c_i) = V[e] dimana V[0] = 2 dan V[1] = m_i
5) Setiap blok yang telah dienkripsi (c_i) digabungkan kembali sehingga menjadi sebuah *ciphertext* yang utuh (C).

11. Implementasi Enkripsi Algoritma LUC

1) *Plaintext* (M) = R
2) e = 7 dan N = 77
3) m_1 = R
4) ASCII dari m_1 = 82
   V[0] = 2 ; V[1] = 82
   V[2] = (82.V[1] – V[0]) mod 77
   = (82.82 – 82) mod 77
   = 6722 mod 77 = 23
   V[7] = (82.V[6] – V[5]) mod 77

= (82.9 – 61) mod 77
= 677 mod 77

V[7] = 61

Maka *ciphertext* dari "R"

adalah 61 = "=" atau $c_1 = 61$

12. Teknik Dekripsi Algoritma LUC

Proses dekripsi sebuah *ciphertext* hampir sama dengan proses enkripsi sebuah pesan, perbedaannya adalah persamaan yang dipakai adalah.

P[*e*] = (*ci* . P[*i* – 1] – P[*i* – 2]) *mod N*, serta kunci yang dipakai adalah kunci dekripsi (*d*,*N*) dimana kunci tersebut telah di ketahui pada proses pembangkitan kunci. Misalkan A telah menerima *ciphertex*t (*c*) dari B dengan menggunakan kunci publik yang telah diberikan kepada B, maka langkah-langkah dekripsi adalah sebagai berikut (Rahman, J.B., Muhsin., Nurhayati, 2018) :

1) *Ciphertext* (*c*) adalah isi pesan atau informasi yang telah dienkripsi oleh B dan diterima oleh A.
2) Nilai *N* didapatkan dari kunci privat yang telah dicari pada tahap pembangkitan kunci.
3) *Ciphertext* yang telah diterima dari B dipecah atau diatur menjadi blok-blok c_1,c_2, c_i yang mempunyai dua karakter pada tiap blok. Setiap blok yang telah didapatkan (c_i) di ubah dalam bentuk ASCII. Gunakan *d* dalam persamaan dekripsi P[d] = (ci . P[i – 1] – P[i – 2]) mod N, Plaintext (mi) = P[d] dimana P[0] = 2 dan P[1] = mi
4) Setiap blok yang telah didekripsi (*mi*) digabungkan kembali sehingga menjadi sebuah *plaintext* yang utuh (M).

13. Implementasi Dekripsi Algoritma LUC

1) *Ciphertext* (C) = "="
2) *d* = *103* dan *N* = 77
3) c_1 = "="
4) ASCII dari c_1 = 61

P[0] = 2
P[1] = 61
P[2] = (61.P[1] – P[0]) mod 77
= (61.61 – 2) mod 77
= 3719 mod 77
P[2] = 23 seterusnya
V[103] = (61.P[102] – P[101]) mod 77
= (61.9 – 5) mod 77
= 544 mod 77
V[103] = 82
Maka *plaintext* dari "=" adalah 82 = "R" atau $m_1 = 82$

14. Steganografi

Kata steganografi berasal dari bahasa Yunani *steganos*, yang artinya tersembunyi atau terselubung, dan *graphein* artinya menulis. Steganografi adalah seni dan ilmu menulis atau menyembunyikan pesan tersembunyi dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia (Jamaludin, 2015).

Steganografi membutuhkan dua properti, yakni media penampung dan pesan rahasia. Media penampung yang digunakan untuk menyembunyikan pesan.

15. Algoritma Chaotic LSB

Least Significant Bit (LSB) merupakan salah satu metode dalam steganografi. Cara kerja dari metode ini adalah melakukan penggantian bit terakhir dari rangkaian bit file cover. Pada Chaotic LSB penggantian bit dilakukan dengan cara random. Akan tetapi, bilangan random yang digunakan memiliki pola tertentu, sehingga meskipun orang yang tidak berkepentingan tahu bahwa di dalam file cover image tersebut terdapat file rahasia, orang tersebut akan kesulitan mengambil bit dari file rahasia tersebut karena tidak mengetahui secara pasti bit-bit mana yang telah diganti. Metode ini akan mengganti tiap bit terakhir dari tiap byte file cover image (Jamaludin, 2015).

Penentuan dari lokasi penyembunyian pesan ditentukan dengan cara sebagai berikut :

1. Untuk *cover-image* RGB

c (x , y) = [Rc , Gc , Bc] berukuran MxN, tentukan sebuah *random seed* dan bangkitkan *pseudorandom number* kemudian susun menjadi sebuah *pseudo-image* RGB

p (x , y) = [Rp ,Gp , Bp] berukuran MxN.

2. Hitung jarak antara *c (x , y)* dan *p (x , y)* dengan menggunakan rumus jarak dua vector :

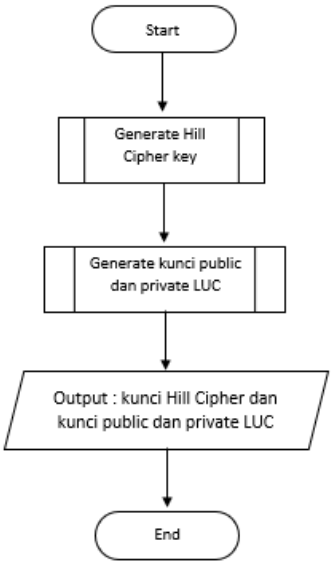
$$d(x,y) = \sqrt{(R_c - R_p)^2 + (G_c - G_p) + (B_c - B_p)^2}$$

3. Penyembunyian dimulai dari lokasi dengan jarak terkecil hingga jarak terbesar. Pengurutan jarak menggunakan algoritma pengurutan data.

Hasil dan Pembahasan

1. Flowchart Bangkit Kunci

Flowchart pada sistem ini dibagi menjadi 3 bagian yakni : bangkit kunci, enkripsi-embeding, dekripsi-ekstraksi. Pada Proses bangkit kunci, user terlebih dahulu membangkitkan dua bilangan prima untuk mendaptkan kunci publik dan kunci private LUC. Kemudian, membangkitkan bilangan acak sebagai matriks kunci Hill Cipher. Flowchart bangkit kunci dapat dilihat pada gambbar 2 berikut.

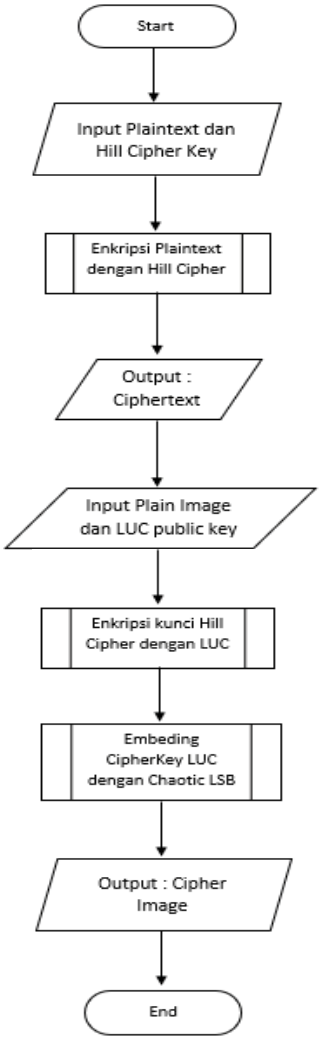


Gambar 1
Flowchart Bangkit Kunci

2. Flowchart Proses Enkripsi dan Embeding

Dapat dilihat pada gambar 3 yang merupakan *flowchart* dari proses enkripsi dan embeding, dimana pengguna meng-*input*-kan *plaintext*, *plain image* dan kunci hill cipher, kemudian sistem melakukan proses enkripsi *plaintext* dengan kunci yang diinputkan, kemudian sistem mengeluarkan hasil *output* pesan yang telah dienkripsi (*ciphertext*). Setelah itu pengguna meng-*input*-kan *public-key* LUC, kemudian sistem melakukan proses enkripsi kunci hill cipher dengan LUC, setelah itu sistem mengeluarkan hasil *output* kunci yang terenkripsi.

Selanjutnya adalah proses embeding yaitu dengan menyisipkan *cipherkey* kedalam bit-bit citra yang di-*input* sehingga menghasilkan *cipherimage.*



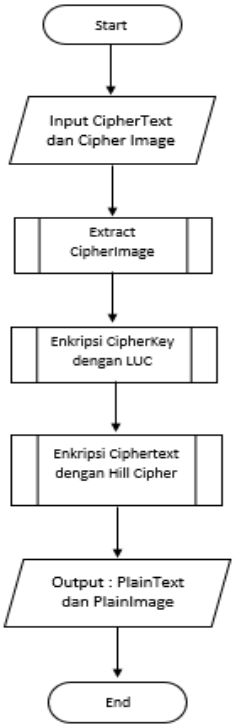
Gambar 2
Flowchart Proses Enkripsi dan Embeding

3. Flowchart Proses Dekripsi dan Ekstraksi

Pada Gambar 2 yang merupakan *flowchart* dari proses *extract* dan dekripsi, dimana pengguna mengi-*input*-kan *ciphertext* dan *cipherimage*. *Cipherimage* akan mengalami proses *extract* untuk mendapatkan *cipherkey*, kemudian sistem melakukan proses dekripsi *cipherkey* dengan LUC. Selanjutnya.

4. Tampilan Antarmuka Menu Utama (*Home)*

Tampilan yang muncul pertama kali pada saat sistem dijalankan adalah tampilan pada form Home. Pada form ini terdapat beberapa identitas penulis, seperti: judul penelitian, logo instansi, nama dan nim penulis, fakultas, program studi, nama universitas, dan tahun. Pada tampilan form Home terdapat 4 menu strip, yaitu Home, Bangkit Kunci, Enkripsi, dan Dekripsi. sistem mengeluarkan hasil *output plainkey*. *Plainkey* akan digunakan untuk proses dekripsi pesan. Setelah itu sistem akan melakukan proses dekripsi *ciphertext* dengan hill cipher. Akhirnya sistem mengeluarkan hasil *output plaintext* kembali, sehingga pesan mudah dibaca.



Gambar 3
Flowchart Proses Extract dan Dekripsi

Gambar 4
Tampilan Antarmuka Menu Utama

5. Tampilan Antarmuka dan Implementasi Bangkit Kunci

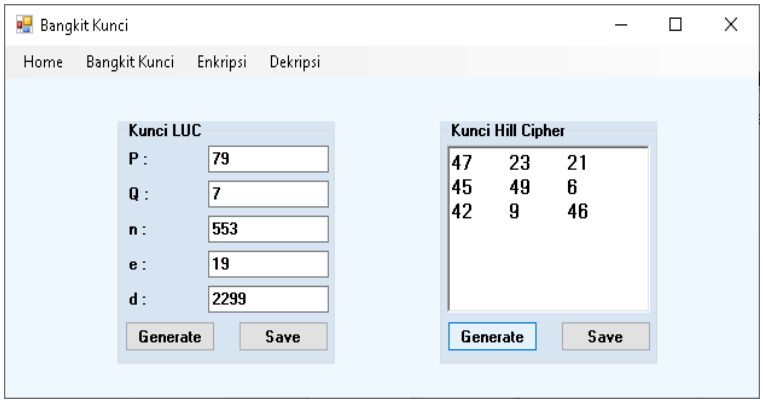
Langkah pertama dalam melakukan proses pembangkitan kunci ialah dengan menekan tombol Generate pada *Groupbox* Kunci LUC untuk mendapatkan bilangan prima acak P dan Q. Proses pembangkitan kunci menggunakan bilangan prima acak

yang telah didapatkan tersebut, proses ini akan mendapatkan pasangan kunci publik (e dan N) dan pasangan kunci privat (d dan N).

Langkah selanjutnya adalah dengan menekan tombol *Generate* pada *Groupbox Hill Cipher* untuk mulai proses pembangkitan kunci *Hill Cipher*. Hasil dari proses pembangkitan kunci dapat dilihat pada gambar 6.

Setelah mendapatkan kunci publik dan kunci private, *user* dapat menyimpan file tersebut dengan menekan tombol *save*. Ketika *user* telah memberi nama pada file kunci *.hk untuk kunci Hill Cipher, *private untuk kunci private LUC, dan *public untuk kunci publik, maka sistem akan menampilkan dialog bahwa kunci berhasil tersimpan,akhirnya kunci tersebut akan tersimpan di media penyimpanan yang dipilih oleh user.

Terdapat juga, tombol reset apabila *user* atau penerima pesan ingin mengulang kembali dalam proses pembangkitan kunci.



Gambar 5
Tampilan Antarmuka dan Implementasi Bangkit Kunci

6. Tampilan Antarmuka dan Implementasi Enkripsi-Embeding

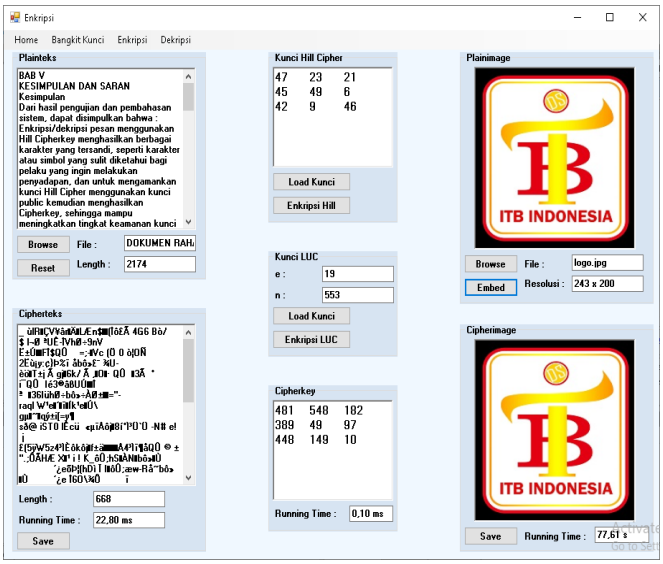
Langkah pertama yaitu melakukan proses enkripsi pesan, dimana *user* atau pengirim pesan terlebih dahulu harus menentukan dan memilih pesan mana yang akan dienkripsikan, dengan syarat pesan yang diinputkan harus ber-format *.txt, *.doc, *.docx. Untuk menginputkan pesan, maka *user* harus menekan tombol Browse. Setelah memilih pesan yang akan dienkripsi.

Langkah selanjutnya adalah *user* atau pengirim pesan harus menginputkan kunci Hill Cipher yang dilakukan dengan memilih kunci yang tadinya sudah disimpan dan kemudian akan muncul di kolom Input Hill Cipher yang disediakan.Setelah pengirim pesan telah memilih file kunci hill cipher, langkah selanjutnya yaitu dengan menekan tombol Enkripsi Hill. Kemudian, sistem akan memulai proses enkripsi pesan, menampilkan *ciphertext* hasil enkripsi, ukuran *ciphertext* serta menampilkan waktu lamanya proses enkripsi, sedangkan kunci yang diinputkan tersebut juga akan tampil di kolom bagian enkripsi kunci. Setelah didapatkan *Ciphertext*, *user* dapat menyimpan *Ciphertext* dengan menekan tombol

Save, kemudian sistem akan menampilkan jendela simpan file, untuk menyimpan file hasi enkripsi, yang akan tersimpan dengan ekstensi *enc.

Langkah ketiga yaitu melakukan proses enkripsi kunci, pada bagian ini *user* perlu menginputkan kunci Hill Cipher. Akan tetapi, karena kunci hiil cipher telah diinputkan sebelumnya dan sistem menampilkan kunci Hill Cipher di GroupBox Kunci Hill. Maka, *user* selanjutnya hanya tinggal menekan tombol Import Public Key untuk mengambil kunci publik, kemudian akan tampil jendela open file. Setelah file kunci *.public diambil , maka sistem akan menampilkan kunci publik (e) dan kunci publik (N). Setelah itu, langkah selanjutnya adalah menekan tombol enkripsi hill. Kemudian sistem akan memulai proses enkripsi kunci, menampilkan *Cipherkey* hasil enkripsi kunci dan waktu lamanya proses enkripsi.

Setelah didapatkan *cipherkey*, *user* dapat menyisipkan kedalam citra yang dipilih, akan tetapi *user* harus menginputkan citra yang dipilih terlebih dahulu kedalam sistem. Setelah memilih citra yang akan digunakan untuk melakukan proses penyisipan *cipherkey*. Setelah memilih citra, maka akan dilakukan proses embedding *cipherkey* kedalam bit-bit citra tersebut yang mana *user* akan menekan tombol embed dan setelah itu sistem akan menampilkan hasil embedding citra. Jika proses embedding cipherkey sudah dianggap selesai maka *user* dapat menyimpan hasil embed tersebut berupa citra yang dapat kemudian dikirim kepada penerima untuk melakukan extract cipherkey nantinya. Keseluruhan tampilan akhir dari proses enkripsi-embeding bisa dilihat pada gambar 7.



Gambar 6
Tampilan Sistem Setelah Selesai Proses
Embed Cipherkey Kedalam Suatu Citra

7. Tampilan Antarmuka dan Implementasi Dekripsi-Ekstraksi

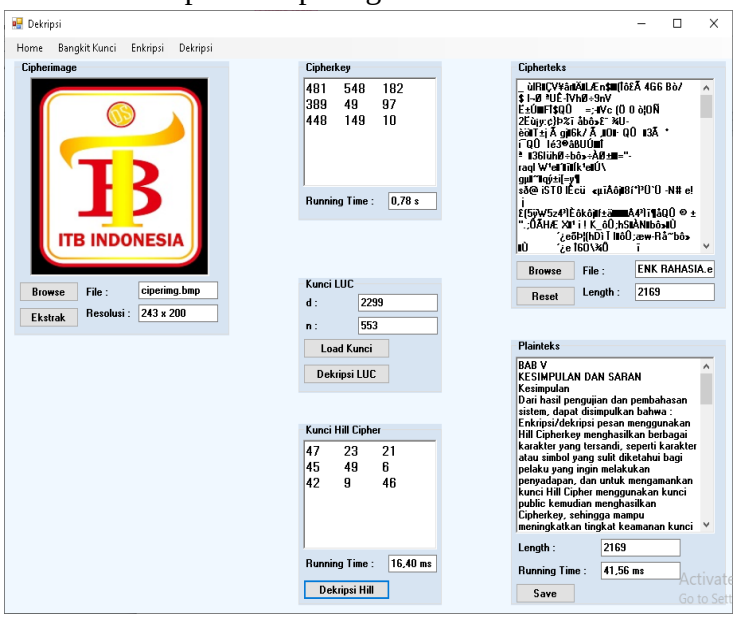
Langkah pertama yaitu melakukan proses pemilihan citra hasil embedding cipherkey yang dikirim sebelumnya kepada penerima. Untuk pemilihan citra tersebut dapat menekan tombol Browse pada groupbox Cipherimage dan akan muncul Pop-

Up windows untuk memilih citra tersebut. Setelah memilih cipherimage tersebut, sistem akan menampilkan gambar bserta keterangan seperti resolusi dan nama file nya.

Langkah kedua yaitu melakukan input kunci private yang dapat dilakukan dengan cara menekan tombol Load Kunci pada Groupbox Kunci LUC. Untuk tampilan harus menginputkan citra yang dipilih terlebih dahulu kedalam sistem.

Tahap selanjutnya adalah tahap extract cipherkey. *User* dapat melakukan proses extract dengan menekan tombol Ekstrak pada Groupbox Cipherimage untuk mendapatkan cipherkey yang telah di enkripsi oleh algoritma LUC. Langkah selanjutnya, yaitu melakukan proses dekripsi kunci, dimana *user* atau penerima pesan terlebih dahulu memiliki *cipherkey* yang akan didekripsikan. Karena cipherkey sudah didapatkan maka itu *user* hanya tinggal menekan tombol dekripsi LUC.

Langkah berikutnya yaitu melakukan proses dekripsi pesan, dimana *user* atau penerima pesan terlebih dahulu harus mengambil file *ciphertext* yang akan didekripsikan. Untuk menginputkan *ciphertext*, maka *user* harus menekan tombol Browse pada Groupbox Cipherteks, yang kemudian akan tampil jendela pilih file umtuk memilih file *enc. Setelah file *ciphertext* *.enc dipilih oleh *user* dan mengklik *open*, kemudian akan tampil nama alamat file, panjang teks, serta pesan yang telah diinputkan di kolom *ciphertext*. Maka, *user* selanjutnya hanya tinggal menekan tombol Dekripsi Hill pada Groupbox Kunci Hill Cipher untuk memulai proses dekripsi pesan, kemudian sistem akan memulai proses dekripsi pesan, menampilkan *Plaintext* hasil dekripsi pesan dan waktu lamanya proses dekripsi. Proeses Dekripsi-Ekstraksi selesai dan ditampilkan seperti gambar 7.



Gambar 7
Tampilan Sistem Setelah Selesai
Proses Dekripsi-Ekstraksi

8. Pengujian Sistem

Pengujian sistem merupakan tahap kedua setelah dilakukannya tahapan proses implementasi. Pengujian sistem dilakukan dengan tujuan untuk melihat dan mengetahui apakah sistem yang dibangun sudah berhasil dan sesuai dengan proses tahapan analisis dan perancangan dalam melakukan proses enkripsi-dekripsi pesan menggunakan algoritma *Hill Cipher*, proses enkripsi-dekripsi kunci menggunakan algoritma LUC dan proses *embed-extract* menggunakan Chaotic LSB, serta mengetahui waktu lamanya proses enkripsi-dekripsi, *embed-extract*.

9. Pengujian Enkripsi dan Embeding

Pengujian ini dilakukan untuk mengetahui pengaruh resolusi file citra yang diinputkan terhadap waktu lamanya proses enkripsi dan embeding pesan dengan menggunakan panjang pesan dan kunci yang sama dalam mengenkripsikan pesan menggunakan kunci *Hill Cipher*. Hasil dari beberapa pengujian enkripsi dan embeding dengan beragam variasi resolusi file citra dapat dilihat pada tabel 2.

Tabel 2
Pengujian Enkripsi Pesan dengan
Variasi Panjang karakter

No	Panjang Karakter sebelum enkripsi	Panjang Karakter setelah enkripsi	Running time
1	11	12	0,09 ms
2	52	54	0,13 ms
3	100	102	2,59 ms
4	565	578	3,14 ms
5	1000	957	3,43 ms

Dari table 2 dapat disimpulkan Panjang karakter sebelum dan sesudah melakukan enkripsi pesan adalah tidak sama dikarenakan penggunaan algoritma Hill cipher memakai konsep modulo 3 sehingga untuk melakukan kalkulasi enkripsi, sistem akan menambahkan (*padding*) karakter spasi. Itu sebabnya karakter setelah enkripsi (*cipher*) tidak sama Panjang dengan karakter asli (*plaintext*).

Dari segi waktu terdapat perbedaan yang signifikasi yaitu semakin banyak jumlah karakter yang ingin dienkripsi maka semakin besar dan lama waktu yang dibutuhkan untuk proses enkripsi.

Tabel 3
Pengujian Embed Cipherkey
dengan Variasi Resolusi Citra

Running Time Embed (ms)	Resolusi Citra			
	250 x 250	500 x 500	750 x 750	1000 x 1000
1	0,32 s	1,71 s	3,86 s	6,70 s
2	0,35 s	1,74 s	3,86 s	6,77 s
3	0,39 s	1,74 s	3,87 s	6,79 s
4	0,41 s	1,80 s	3,97 s	6,85 s
5	0,41 s	1,70 s	3,94 s	6,84 s

Tabel 3 menunjukkan waktu yang berbanding lurus dengan jumlah pixel atau resolusi yang digunakan. Semakin besar ukuran resolusi suatu citra maka akan

semakin besar waktu yang digunakan untuk proses embedding cipherkey kedalam bit-bit citra tersebut.

10. Pengujian Dekripsi dan Ekstraksi

Pengujian ini dilakukan untuk mengetahui pengaruh resolusi file cipherimage yang diinputkan terhadap waktu lamanya proses dekripsi pesan dengan menggunakan panjang pesan dan kunci yang sama dalam mendekripsikan pesan menggunakan kunci Hill Cipher. Hasil dari beberapa pengujian dekripsi dengan beragam variasi format file dapat dilihat pada tabel 4.

Tabel 4
Pengujian Extract Cipherkey dengan Variasi Resolusi Citra

Running Time Extract (ms)	Resolusi Citra			
	250 x 250	500 x 500	750 x 750	1000 x 1000
1	0,27 s	1,02 s	2,33 s	4,19 s
2	0,26 s	1,04 s	2,46 s	4,22 s
3	0,24 s	1,10 s	2,40 s	4,24 s
4	0,23 s	1,07 s	2,42 s	4,24 s
5	0,23 s	1,08 s	2,45 s	4,25 s

Pada tabel 4 yang menjelaskan extracting cipherkey pada resolusi citra menunjukkan bahwa resolusi yang besar akan memakan waktu lebih lama daripada resolusi yang lebih rendah karena akan memproses semua piksel. Akan tetapi proses extract relatif lebih cepat jika dibandingkan dengan proses embedding.

Tabel 5
Pengujian Enkripsi Pesan dengan Variasi Panjang Karakter

No	Panjang Karakter sebelum dekripsi	Panjang Karakter setelah dekripsi	Running Time
1	12	11	0,12 ms
2	54	52	0,13 ms
3	102	100	1,90 ms
4	578	565	2,4 ms
5	957	1000	3,0ms

Tabel 5 menunjukkan lama waktu dekripsi suatu pesan yang tersandikan. Untuk tabel diatas menunjukkan Panjang karakter yang berubah dikarenakan proses perhitungan yang mengharuskan modulo 3 sehingga panjang ciphertext tidak sama panjang dengan plaintext dapat kembali seperti semula. Dari segi waktu juga tidak bergitu beda jauh tetapi memiliki selisih dimana semakin banyak jumlah karakter yang didekripsi maka semakin besar juga waktu yang dibutuhkan untuk proses tersebut.

Kesimpulan

Dari hasil pengujian dan pembahasan sistem, dapat disimpulkan bahwa : 1) Enkripsi/dekripsi pesan menggunakan Hill Cipherkey menghasilkan berbagai karakter yang tersandi, seperti karakter atau simbol yang sulit diketahui bagi pelaku yang ingin

melakukan penyadapan, dan untuk mengamankan kunci Hill Cipher menggunakan kunci public kemudian menghasilkan Cipherkey, sehingga mampu meningkatkan tingkat keamanan kunci pesan. 2) Proses embedding atau penyisipan cipherkey kedalam bit-bit citra tidak menunjukkan perubahan warna yang signifikan bahkan terkesan tidak memiliki perubahan atau tidak mengalami proses embeding dikarenakan perubahan bit warna citra tersebut sangat kecil. 3) Perbandingan waktu dari Panjang karakter yang dienkripsi yaitu 10,50,100,500,1000 karakter mengalami waktu proses yang banding lurus yaitu semakin Panjang suatu pesan yang akan dienkripsi atau dekripsi, maka semakin besar atau lama waktu yang dibutuhkan. 4) Perbandingan waktu dari hasil embedding dan extract cipherkey kedalam citra berbanding lurus dengan jumlah resolusi yang digunakan yaitu 250x250, 500x500, 750x750, 1000x1000. Untuk resolusi yang lebih besar membutuhkan waktu proses embedding dan extract lebih lama. Namun terdapat kesimpulan hasil embed lebih lama daripada extract dikarenakan pada proses embedding sistem akan melakukan proses penyisipan kemudian menyusun ulang hingga membentuk citra baru yang memiliki bit yang berbeda yang dinamakan cipherimage.

BIBLIOGRAFI

Anggraini, S. (2014). *Implementasi Sistem Keamanan Data Menggunakan Algoritma RSA Dan Modified LSB*. Universitas Sumatera Utara. Google Scholar

Ariyus, D. (2008). *Pengantar Ilmu Kriptografi Teori, Analisis, dan implementasi*. Google Scholar

Jamaludin. (2015). Pengamanan Data Dengan Kombinasi Teknik Kriptografi Rabin Dan Teknik Steganografi Chaotic LSB. *Seminar Nasional Teknologi Informasi Dan Komunikasi*. Google Scholar

Laoli, D., Sinaga, B., & Sinaga, A. S. R. M. (2020). Penerapan Algoritma Hill Cipher Dan Least Significant Bit (LSB) Untuk Pengamanan Pesan Pada Citra Digital. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *4*(3), 138–148. Google Scholar

Mollin, R. A. (2007). *Cryptography: Theory and Practice*. JSTOR. Google Scholar

Munir, R. (2006). Pengantar Kriptografi. *ITB, Bandung*. Google Scholar

Oktafiansyah, N. M. D., Agus, F., & Maharani, S. (2016). Penerapan Kriptografi Dengan Algoritma Data Encryption Standart Pada Text Hasil Konversi Dari Citra. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi Vol*, *1*(1). Google Scholar

Rahman, J.B., Muhsin., Nurhayati, Y. (2018). Implementasi Algoritma LUC Untuk Pengamanan Pesan Berbasis Android. *Jurnal Nuansa Informatika*, *12*(1). Google Scholar

Rahman, J. B., & Nurhayati, Y. (2018). Impelementasi Algoritma Luc Untuk Pengamanan Pesan Berbasis Android. *Nuansa Informatika*, *12*(1). Google Scholar

Serdano, Akbar., Zarlis, Muhammad., Sawaluddin., Hartama, D. (2019). Pengamanan Pesan Menggunakan Algoritma Hill Cipher Dalam Keamanan Komputer. *Jurnal Mahajana Informasi*, *3*(2). Google Scholar

Stallings, W. (2005). Intruders. *Cryptography and Network Security, Fourth Edition, Prentice Hall*, 565–594. Google Scholar

Whitten, J. L., & Bentley, L. D. (2007). System analysis and design for the global enterprise. *Journal of Small Business Management*, *17*(1). Google Scholar