TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Implementasi Metode GOST(Government Standard) dan LSB-1(Least Significant Bit) Untuk Mengamankan Teks

Mia Diana

Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: miadiana1402@gmail.com

Abstrak−Pengamanan data teks merupakan salah satu kegiatan yang dilakukan agar teks dari sebuah informasi yang dirahasiakan agar tidak dapat diketahui oleh orang lain kecuali orang-orang yang diberi hak untuk itu. Salah satu teknik yang dapat dilakukan untuk mengoptimalkan pengamanan data teks adalah mengkombinasikan teknik kriptografi dengan teknik steganografi. Teknik krriptografi berfungsi untuk menyandikan data teks yang dirahasiakan melalui proses enkripsi. Sedangkan teknik steganografi berfungsi untuk menyembunyikan (embedding) data teks tersandi tersebut kedalam sebuah media misalnya citra, video, audio. Penelitian ini menguraikan proses pengkombinasian tekni kriptografi dan teknik steganografi dalam mengamankan teks yang dikirim. Teknik kriptografi digunakan untuk melakukan penyandian pesan rahasia berdasarkan metode GOST (Government Standart) dan teknik steganografi digunakan untuk menyembunyian teks yang terenkripsi kedalam sebuah citra digital berdasarkan algoritma LSB-1(Least significant bit-1). Agar proses yang dilakukan lebih mudah maka dibangun sebuah aplikasi pengaman teks rahasia menggunakan bahasa pemograman visual basic 2008,

Kata Kunci: Kriptografi; Steganografi; GOST; LSB-1.

Abstact−Securing text data is one of the activities carried out so that the text of an information is kept confidential so that it cannot be known by others except those who are entitled to it. One technique that can be done to optimize the security of text data is to combine cryptographic techniques with the steganography technique. Cryptographic techniques function to encode text data that is kept secret through the encryption process. While the steganography technique serves to hide (embedding) encrypted text data into a media such as image, video, audio. This study describes the process of combining cryptographic techniques and steganography techniques in securing sent text. Cryptographic techniques are used to encode secret messages based on the GOST (Government Standard) method and the steganography technique is used to hide encrypted text into a digital image based on the LSB-1 (Least significant bit-1) algorithm. In order to make the process easier, a secret text security application was built using Visual Basic 2008 programming language.

Keywords: Cryptography; Steganography; GOST; LSB-1

1. PENDAHULUAN

Pemakaian teknologi komputer sebagai salah satu aplikasi dari teknologi informasi sudah menjadi suatu kebutuhan, karena banyak pekerjaan yang dapat diselesaikan dengan cepat akurat dan efesien. Dengan berkembangnya teknik telekomunikasi dan sistem pengolahan data yang berkaitan erat dengan komunikasi antar pengguna komsputer yang satu dengan yang lain yang berfungsi menyalurkan data sehingga masalah keamanan merupakan salah satu aspek penting.

Teks rahasia yang dapat diamankan dari pihak-pihak yang tidak berhak dengan menggunakan teknik kriptografi, yang mana teknik ini dapat merubah makna asli dari plainteks ke dalam bentuk cipherteks. Teknik kriptografi ini memiliki beberapa kelemahan salah satunya yaitu dapat menimbulkan kecurigaan pada pihak lain bahwa informasi tersebut bersifat rahasia, oleh karena itu pihak lain menimbulkan niat ingin mngetahui informasi yang sebenarnya [1]. Kriptografi mempelajari tentang ilmu informasi dan pengamanan data, dalam kriptografi banyak ditemukan metoda-metoda kriptografi.

Metode GOST (Government Standard) adalah salah satu metode kriptografi yang dapat digunakan untuk mengamankan teks. Metode ini memiliki cara sederhana untuk mengenkripsi teks dengan memakai jumlah proses sebanyak 32 round (putaran) dan memakai 64 bit block cipher dengan 256 bit key. Metode GOST juga menggunakan 8 buah S-Box yang berbeda-beda dan operasi XOR serta Left Circular Shift [2]. Steganografi mempunyai proses yang berbeda dengan kriptografi dimana teks rahasia yang ingin dikirimkan tidak di acak melainkan disembunyikan pada penampungnya. Hal ini sangat menguntungkan karena akan mengurangi keinginan seseorang untuk memeriksa informasi teks tersebut [3].

Steganografi teks “disamarkan” dalam bentuk yang relatif “aman” sehingga tidak terjadi kecurigaan. Steganografi dapat digunakan oleh berbagai jenis bentuk data, yaitu image, audio, dan video. Sedangkan pada kriptografi pesan disembunyikan dengan “diacak” sehingga pada kasus - kasus tertentu dapat dengan mudah mengundang kecurigaan. Salah satu metode yang digunakan dalam steganografi adalah Metode LSB (Least Significant Bit) Metode ini berkerja dengan cara mengganti bit terakhir dari masing-masing piksel dengan teks yang akan disisipkan. LSB mempunyai kelebihan yakni ukuran gambar tidak akan berubah [4].

Penelitian ini menguraikan metode GOST dan metode LSB-1 untuk mengoptimalkan pengamanan data teks dengan cara teks terlebih dahulu berdasarkan metode GOST kemudian menyembunyikan ke dalam citra berdasarkan metode LSB-1. Kombinasi metode GOST (Government Standart) dan LSB-1 (Least Significant Bit) mampu meningkatkan keamanan teks rahasia dari tindakan-tindakan orang lain yang tidak bertanggung jawab karena teks yang disembunyikan berdasarkan metode LSB-1 telah dienkripsi (disandikan) terlebih dahulu sehingga didapatkan pengamanan pesan yang belapis.

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan berupa teks yang dikirim oleh pengirim dapat di sampaikan kepada penerima dengan aman. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut kriptologi. Kriptologi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui pihak yang tidak berhak [6].
Terdapat empat elemen utama di dalam teknik kriptografi yang saling terkait satu sama lain [7], yaitu:

1. Plaintext
   Plaintext adalah pesan awal atau pesan asli yang dikirim pada proses komunikasi dan di jaga keamannannya. Plaintext inilah yang kemudian dienkripsi dan dideskripsi.
2. Ciphertext
   Ciphertext adalah pesan yang tersembunyi, yaitu pesan asli yang telah dienkripsi (disandikan) pada proses kriptografi. Ciphertext ini dapat diubah kembali kebentuk aslinya (Plaintext) memanfaatkan key yang telah disediakan.
3. Cryptography Key
   Cryptography Key adalah kunci yang digunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (key) yang sama maka proses enkripsi dan deskripsi tidaka dapat dilakukan dengan baik. Kunci (key) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya kriptografi.
4. Encryption Decryption Algorithm
   Encryption Decryption Algorithm adalah komponen terakhir yang juga sama pentingnya dalam proses kriptografi adalah algoritma yang digunakan untuk enkripsi dan dekripsi.

2.2 Metode Govermant Standard (GOST)

GOST adalah singkatan dari Gosudarstvennyi Standard atau Government Standard. Metode GOST adalah algoritma blok sandi yang dikembangkan oleh nasional Uni Soviet. Metode ini dikembangkan oleh Uni Soviet selama Perang Dingin untuk menyembunyikan data atau informasi yang bersifat rahasia pada saat komunikasi.

Algoritma ini adalah algoritma enkripsi sederhana yang memiliki beberapa proses sebanyak 32 round (putaran) dan menggunakan blok cipher 64-bit dengan kunci 256-bit [10]. Metode GOST juga menggunakan operasi S-Box 8 yang berbeda dan operasi XOR dan Circular Shift Left. Kelemahan GOST yang dikenal sampai sekarang adalah karena jadwal utamanya adalah bahwa dalam keadaan tertentu menjadi titik lemah dari metode pembacaan sandi sebagai cryptanalysis kunci terkait. Kelemahan ini dapat diatasi dengan meneruskan kunci ke fungsi hash yang kuat dalam kriptografi seperti SHA-1, kemudian hasilnya digunakan untuk menginput kunci hash inisialisasi. Keuntungan dari metode ini adalah kecepatan GOST cukup bagus, meskipun tidak secepat blowfish, lebih cepat daripada IDEA.

2.3 Metode LSB

Metode LSB merupakan metode steganografi yang paling sederhana dan paling mudah diimplementasikan [11]. Untuk menjelaskan metode ini kita menggunakan Citra digital sebagai coverteks.setiap pixel didalam citra berukuran 1 sampai 3 byte pada susunan bit dalam sebuah byte (I byte =8 bit), ada bit yang paling berarti (least significant bit atau LSB). Misalnya pada byte 11010010, bit 1 yang pertama (ditebalkan) adalah bit MSB dan bit 0 yang terakhir (ditebalkan) adalah bit LSB. Bit yang cocok untuk diganti dengan bit pesan adalah bit lsb, sebab modifikasi hanya mengubah nilai byte tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalnya byte tersebut di dalam gambar memberikan persebsi warna merah, maka perubahan satu bit LSB hanya mengubah persepsi warna merah tidak terlalu berarti. Mata manusia tidak dapat membedakan perubahan yang kecil ini.
Sebagai ilustrasi, misalnya segmen pixel citra sebelum penampakan bit bit watermark adalah
00110011 10100010 11100010 01101111
Misalnya pesan rahasia (yang telah dikoversi ke sistem biner) adalah 0111. Setiap bit dari watermark menggantikan pisisi LSB dari segmen pixel citra menjadi [11]:
00110010 10100011 11100011 01101111

2.4 Metode LSB-1

Least Significant Bit-1 (LSB-1), bekerja dengan teknik menukarkan bit citra penampung dimana posisi bit yang ditukar adalah bit ke 8-1 (bit ke-7). Contoh, asumsikan representasi biner pixel citra sebagai berikut [15] :
11110101 00010110 10101010
11000100 11111001 00000001
00000001 11110001 00011101
karakter T dalam biner = 01010100, maka akan dihasilkan citra hasil dengan urutan bit akhir sebagai berikut :
11110101 00010110 10101000
11000110 11111001 00000001

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

00000001 11110011 00011101

Kekurangan dari Least Significant Bit Insertion

Dapat diambil kesimpulan dari contoh 8 bit pixel, menggunakan Least Significant Bit Insertion dapat secara drastis merubah unsur pokok warna dari pixel. Ini dapat menunjukkan perbedaan yang nyata dari cover image menjadi stego image, sehingga tanda tersebut menunjukkan keadaan dari Steganography. Variasi warna kurang jelas dengan 24 bit image, bagaimanapun file tersebut sangatlah besar. Antara 8 bit dan 24 bit image mudah diserang dalam pemrosesan image, seperti cropping (kegagalan) dan compression (pemampatan).

Keuntungan dari LSB Insertion

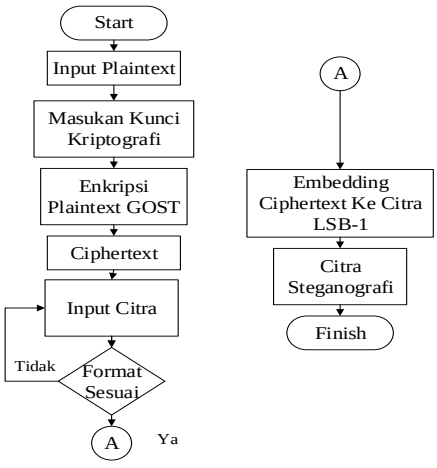
Keuntungan yang paling besar dari algoritma LSB ini adalah cepat dan mudah. Dan juga algoritma tersebut memiliki software Steganography yang mendukung dengan bekerja di antara unsur pokok warna LSB melalui manipulasi pallete (lukisan).

Untuk dapat menutup kelemahan ini maka untuk penelitian ini akan ditambahkan sistem pengamanan lagi yaitu dilakukan dengan yang namanya Kriptografi Kriptografi mengacak pesan sehingga tidak mudah untuk dimengerti, sedangkan Steganography menyembunyikan pesan sehingga tidak terlihat.

3. HASIL DAN PEMBAHASAN

Data teks yang bersifat rahasia rentan terhadap penyadapan dengan menyalurkan data rahasia ke internet. Menyalurkan data teks rahasia dengan bantuan media internet memiliki aspek penting yang harus diperhatikan oleh pengirim dan penerima yaitu aspek keamanan data teks rahasia tersebut. Apabila data rahasia jatuh kepihak yang tidak memiliki otoritas, maka dapat merugikan salah satu pihak pengirim. Masalah tersebut dapat diatasi dengan teknik kriptografi. Akan tetapi teknik kriptografi ini memiliki beberapa kelemahan salah satunya adalah dapat menimbulkan kecurigaan pada pihak lain tentang informasi yang dikandung didalamnya. Sehingga dibutuhkan sebuah teknik kombinasi untuk memaksimalkan data teks rahasia yang dikirim tanpa adanya kecurigaan dari pihak yang tidak bertanggung jawab.

Berdasarkan pembahasan rumusan masalah pada bab sebelumnya dan paparan di atas adalah bagaimana mengamankan data teks rahasia dengan teknik kriptografi tanpa adanya kecurigaan bahwa data tersebut bersifat rahasia, maka dalam skripsi ini akan menambah teknik steganografi. Teknik steganografi adalah sebuah teknik yang dapat menyembunyikan (embedding) teks rahasia kedalam sebuah objek sehingga tidak memberikan kecurigaan terhadap pihak lain. Objek tersebut dalam skripsi ini adalah sebuah gambar citra digital. Sebelum melakukan kombinasi teknik kriptografi data teks rahasia (plaintext) akan dienkripsi terlebih dahulu dengan algoritma GOST sehingga menghasilkan output berupa ciphertext. Kemudian ciphertext akan disembunyikan (embedding) kedalam objek citra digital berupa gambar menggunakan algoritma LSB-1. Adapun proses pengamanan pesan berdasarkan kombinasi teknik kriptografi dan steganorafi disajikan pada gambar di bawah ini:



Gambar 1. Diagram Proses Enkripsi dan Embedding Kriptografi Steganogarfi

Berdasarkan pada gambar di atas menyatakan bahwa proses yang dilakukan untuk menyisipkan pesan yang terenkripsi menggunakan teknik kriptografi dan steganografi adalah menyiapkan sebuah plaintext yang akan dienkripsi menggunakan algoritma GOST. Hasil enkripsi berupa ciphertext yang akan disisipkan kedalam objek. Objek tersebut berupa gambar citra digital yang inputkan oleh user. Citra digital yang akan digunakan sebagai objek penyembunyi (citra cover) adalah citra digital berformat bmp yang mampu menampung seluruh ciphertext yang akan disembunyikan. Proses selanjutnya adalah menyisipkan setiap biner dari ciphertext hasil enkrispi menggunakan algoritma steganografi LSB kedalam objek berupa gambar citra digital. Hasil dari penyisipan adalah citra stegano yang tidak mengalami perubahan bentuk jika dilihat dengan mata manusia.

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Contoh kasus proses hitungan manual dalam enkripsi dan embedding menggunakan algoritma GOST dan LSB-1. Proses pertama adalah menyiapkan plaintext untuk enkripsi menggunakan algoritma GOST. Kemudian menyiapkan sebuah gambar citra yang akan dijadikan objek penyembunyian teks hasil enkripsi menggunakan algoritma LSB-1.

1. Enkripsi Berdasarkan Algoritma GOST

Enkripsi adalah proses pertama sebelum teks di sembunyikan di objek citra digital. Proses enkripsi GOST dilakukan sebanyak 32 putaran (ronde) dan pembangkitan kunci. Pada hitungan manual ini plaintext dan kunci yang digunakan untuk enkripsi menggunakan algoritma GOST adalah sebagai berikut :

Plainteks = MIADIANA
Kunci : Algoritma_GOST_mia_diana_tanjung

a. Pembentukan Kunci
Kunci dirubah ke dalam bentuk biner dan dikelompokan menjadi 8 kelompok. Setiap kelompok kunci memiliki panjang 32 bit. Adapun proses pembangkitan kunci adalah mengkonversi setiap karakter kunci ke dalam bentuk desimal dan biner seperti tabel di bawah ini:

Tabel 1. Kunci Desimal dan Biner GOST

Char	Desimal	Biner	Char	Desimal	Biner
A	65	01000001	I	105	01101001
l	108	01101100	A	97	01100001
g	103	01100111	_	95	01011111
o	111	01101111	D	100	01100100
r	114	01110010	I	105	01101001
i	105	01101001	A	97	01100001
t	116	01110100	N	110	01101110
m	109	01101101	A	97	01100001
a	97	01100001	_	95	01011111
_	95	01011111	T	116	01110100
G	71	01000111	A	97	01100001
O	79	01001111	N	110	01101110
S	83	01010011	J	106	01101010
T	84	01010100	U	117	01110101
_	95	01011111	N	110	01101110
M	109	01101101	G	103	01100111

Kelompokkan biner kunci menjadi 8 kelompok jumlah bit kunci dalam sebuah kelompok adalah 32 bit. Adapun contoh proses pengelompokan kunci sebagai berikut :
Biner kunci keseluruhan:
01000001 01101100 01100111 01101111
01110010 01101001 01110100 01101101
01100001 01011111 01000111 01001111
01010011 01010100 01011111 01101101
01101001 01100001 01011111 01100100
01101001 01100001 01101110 01100001
01011111 01110100 01100001 01101110
01101010 01110101 01101110 01100111

Cara pengelompokkan adalah mengambil 32 bit pertama dengan menulis biner dengan cara terbalik dari kanan kekiri atau dari bit ke 32 hingga bit 1 untuk kelompok pertama. Demikian untuk kelompok bit seterusnya. Adapun proses pengelompokkan kunci 32 bit yang sudah di balik adalah sebagi berikut :

Tabel 2. Kelompok Kunci GOST

Kunci	Posisi Bit	Biner
K[0]	32,,,1	11110110111001100011011010000010
K[1]	64,,,33	10110110001011101001011001001110
K[2]	95,,,65	11110010111000101111101010000110
K[3]	128,,,96	10110110111110100010111011001010
K[4]	160,,,129	00100110111110101000011010010110
K[5]	192,,,161	10000110011101101000011010010110
K[6]	224,,,193	01110110100001100010111011111010
K[7]	256,,,225	11100110011101101010111001010110

a. Enkripsi Ronde 0

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Plainteks = MIADIANA
Konversi *plaintext* ke dalam bentuk decimal dan biner. Kemudian kelompokkan *plaintext* biner menjadi 2 bagian yaitu R[0] dan L[0] dengan jumlah kelompok 32 bit. 32 bit bagian kiri akan menjadi R[0] dan mulai bit 33 hingga 64 akan menjadi bagian L[0]. Proses penulisan bit dilakukan dengan cara terbalik.
R[0] = Bit [32], bit [31]….bit[1]
L[0] = Bit [64], bit [63]…bit[33]

Adapun biner plainteks dapat dilihat pada tabel sebagai berikut :

Tabel 3. Biner *Plaintext*

Char	M	I	A	D	I	A	N	A
Des	77	73	65	68	73	65	78	65
Biner	0100 1101	0100 1001	0100 0001	0100 0100	0100 1001	0100 0001	0100 1110	0100 0001

Gabungan Biner *Plaintext*:
01001101 01001001 01000001 01000100 (kiri)→R0
01001001 01000001 01001110 01000001 (kanan)→L0
Kemudian susun biner dengan susunan terbalik dari sebelah kanan kek kiri untuk R[0] dan L[0] sebagai berikut :
L[0]=10000010011100101000001010010010
R[0]=00100010100000101001001010110010

Proses enkripsi memiliki beberapa tahap. Tahap proses enripsi ronde ke 0 adalah sebagai berikut :

Tahap 1: L[0]=10000010011100101000001010010010
R[0]=00100010100000101001001010110010→rubah ke decimal

Tahap 2: (R[0] + K[0]) Mod 2^{23}
R[0] = 578.982.578
K[0] = 4142282370 +
Hasil = 4.721.264.948 mod $2^{23(4294967296)}$
Hasil mod = 426.297.652→konversikan ke biner
Biner = 0001 1001 0110 1000 1100 1001 0011 0100

Tahap 3: kelompokkan biner hasil di atas menjadi 4 bit

Tabel 4. S-Box round 0

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
0001	1	→S-Bok(1)	10	1010
1001	9	→S-Bok(2)	3	0011
0110	6	→S-Bok(3)	4	0100
1000	8	→S-Bok(4)	14	1110
1100	12	→S-Bok(5)	0	0000
1001	9	→S-Bok(6)	6	0110
0011	3	→S-Bok(7)	1	0001
0100	4	→S-Bok(8)	5	0101

Tahap 4: Hasil gabungan :10100011010011100000011000010101
Hasil RLS [0] 11 Bit : 01110000001100001010110100011010
Tahap 5: R[1] = RLS[0] XOR L[0]
RLS[0] = 01110000001100001010110100011010
L[0] = 10000010011100101000001010010010⊕
R[1] = 11110010010000100010111110001000
L[1] = R[0] Sebelum di proses
L[1] = 00100010100000101001001010110010
Tahap 6: Hasil Putaran -0 atau pada PUTARAN – I =0, adalah :
L[1] = 00100010100000101001001010110010
R[1] = 11110010010000100010111110001000

b. Enkripsi Ronde 1
Tahap 1: L[1] = 00100010100000101001001010110010
R[1] = 11110010010000100010111110001000→rubah ke decimal
Tahap 2: (R[1] + K[1]) Mod 2^{23}

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

R[1]= 4064423816
K[1]= 3056506446 +
hasil= 7120930262 mod 2^{23}
Hasil mod = 2825962966 → konversikan ke biner
Biner = 1010 1000 0111 0000 1100 0101 1101 0110

Tahap 3: Kelompokkan biner hasil di atas menjadi 4 bit

Tabel 5. S-Box round 1

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
1010	10	→S-Bok(1)	1	0001
1000	8	→S-Bok(2)	2	0010
0111	7	→S-Bok(3)	2	0010
0000	0	→S-Bok(4)	7	0111
1100	12	→S-Bok(5)	0	0000
0101	5	→S-Bok(6)	2	0010
1101	13	→S-Bok(7)	8	1000
0110	6	→S-Bok(8)	10	1010

Tahap 4: Hasil gabungan : 000100100011001110000001010001010
Hasil RLS [1] 11 Bit : 00111000000101000101000010010001
Tahap 5: R[2] =RLS[1] XOR L[1]
RLS[1] = 00111000000101000101000010010001
L[1] = 00100010100000101001001010110010⊕
R[2] = 00011010100101101100001000100011
L[2] = R[1] Sebelum di proses
L[2] = 11110010010000100010111110001000

Tahap 6: Hasil Putaran -1 atau pada PUTARAN – I =1, adalah :
L[2] = 11110010010000100010111110001000
R[2] = 00011010100101101100001000100011

c. Enkripsi Ronde 2

Tahap 1: L[2] =11110010010000100010111110001000
R[2] = 00011010100101101100001000100011→rubah ke decimal
Tahap 2 : (R[2] + K[2] Mod 2^{23}
R[2] = 446087715
K[2] = 4074961542 +
Hasil = 4521049257 mod $2^{23(4294967296)}$
Hasil mod = 226081961→konversikan ke biner
Biner = 0000 1101 0111 1001 1011 1100 1010 1001
Tahap 3: kelompokkan biner hasil di atas menjadi 4 bit

Tabel 6. S-Box round 2

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
0000	0	→S-Bok(1)	2	0100
1101	13	→S-Bok(2)	7	0111
0111	7	→S-Bok(3)	4	0010
1001	9	→S-Bok(4)	11	0100
1011	11	→S-Bok(5)	9	1110
1100	12	→S-Bok(6)	2	1001
1010	10	→S-Bok(7)	2	0111
1001	9	→S-Bok(8)	9	0010

Tahap 4 : Hasil gabungan : 0100 0111 0010 0100 1110 1001 0111 0010
Hasil RLS [3] 11 Bit : 00100111010010111001001000111001
Tahap 5 : R[3] = RLS[1] XOR L[1]
RLS[2] = 00100111010010111001001000111001
L[2] = 11110010010000100010111110001000⊕
R[3] = 11010101000010011011110110110001
L[3] = R[2] Sebelum di proses

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

L[3] = 00011010100101101100001000100011
Tahap 6:Hasil Putaran -2atau pada PUTARAN – I =2, adalah :
L[3] =00011010100101101100001000100011
R[3] =11010101000010011011110110110001

Proses berikutnya untuk ronde ke 3 hingga ronde ke 30 dilakukan dengan cara yang sama seperti proses sebelumnya. Sehingga berdasarkan hitungan manual keseluruhan ronde 3 hingga ronde ke 30 dapat dilihat pada tabel di bawah ini:

Tabel 7. Hasil Putaran 3 Sampai Dengan 30 GOST

RONDE	L[i] dan R[i]	Nilai Biner L[i] dan R[i]
RONDE 3	L[3]	00011010100101101100001000100011
	R[3]	11010101000010011011110110110001
RONDE 4	L[4]	11010101000010011011110110110001
	R[4]	01011000110001010100111010110011
RONDE 5	L[5]	01011000110001010100111010110011
	R[5]	01001100100110000010101011111010
RONDE 6	L[6]	01001100100110000010101011111010
	R[6]	11110111100110000100000101010001
RONDE 7	L[7]	11110111100110000100000101010001
	R[7]	11100100111100010000110111010000
RONDE 8	L[8]	11100100111100010000110111010000
	R[8]	10001000110100100001001011011011
RONDE 9	L[9]	10001000110100100001001011011011
	R[9]	00111110010001101101001010011011
RONDE 10	L[10]	00111110010001101101001010011011
	R[10]	11001110010010110000001101101010
RONDE 11	L[11]	11001110010010110000001101101010
	R[11]	10101111001000001101100101000011
RONDE 12	L[12]	10101111001000001101100101000011
	R[12]	10101101010101011101101100010110
RONDE 13	L[13]	10101101010101011101101100010110
	R[13]	00010001111111111110111011110010
RONDE 14	L[14]	00010001111111111110111011110010
	R[14]	11100001010001011001011010000111
RONDE 15	L[15]	11100001010001011001011010000111
	R[15]	01110001111011111001001010100001
RONDE 16	L[16]	01110001111011111001001010100001
	R[16]	10101011111000111011001010010101
RONDE 17	L[17]	10101011111000111011001010010101
	R[17]	01010100010110100011001000000010
RONDE 18	L[18]	01010100010110100011001000000010
	R[18]	11011011111111000011100011010010
RONDE 19	L[19]	11011011111111000011100011010010
	R[19]	01010110010110100010111100111110
RONDE 20	L[20]	01010110010110100010111100111110
	R[20]	01011100000000101111001011101011
RONDE 21	L[21]	01011100000000101111001011101011
	R[21]	11000010011010100101010000011011
RONDE 22	L[22]	11000010011010100101010000011011
	R[22]	11100101110000010000110001111111
RONDE 23	L[23]	11100101110000010000110001111111
	R[23]	00111010110001101100010010011111
RONDE 24	L[24]	00111010110001101100010010011111
	R[24]	01110001100101110011000010100001
RONDE 25	L[25]	01110001100101110011000010100001
	R[25]	10101011001110001110000010001101
RONDE 26	L[26]	10101011001110001110000010001101
	R[26]	11101010111001110001010001111010
RONDE 27	L[27]	11101010111001110001010001111010
	R[27]	00111110000101010110111111010100
RONDE 28	L[28]	00111110000101010110111111010100

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

RONDE	L[i] dan R[i]	Nilai Biner L[i] dan R[i]
	R[28]	01110011111011011000110000010000
RONDE 29	L[29]	01110011111011011000110000010000
	R[29]	11000001010100010111001100010000
RONDE 30	L[30]	11000001010100010111001100010000
	R[30]	11110101000010001101101000100110
RONDE 31	L[31]	11110101000010001101101000100110
	R[31]	00111101011101011101101110011110

d. Enkripsi Ronde 31

Tahap 1: L[31]=11110101000010001101101000100110

R[31]=00111101011101011101101110011110→rubah ke decimal

Tahap 2 :(R[31] + K[0] Mod 2^{23})

R[31] = 1031134110

K[0] = 4142282370+

Hasil = 5173416480 mod $2^{23(4294967296)}$

Hasil mod = 878449184→konversikan ke biner

Biner = 0011 0100 0101 1100 0001 0010 0010 0000

Tahap 3: Kelompokkan biner hasil di atas menjadi 4 bit

Tabel 8. S-Box round 31

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
0011	3	→S-Bok(1)	2	0010
0100	4	→S-Bok(2)	6	0110
0101	5	→S-Bok(3)	3	0011
1100	12	→S-Bok(4)	11	1011
0001	1	→S-Bok(5)	12	1100
0010	2	→S-Bok(6)	10	1010
0010	2	→S-Bok(7)	4	0100
0000	0	→S-Bok(8)	1	0001

Tahap 4 : Hasil gabungan : 0010 0110 0011 1011 1100 1010 0100 0001

Hasil RLS [31] 11 Bit: 11011110010100100000100100110001

Tahap 5 :R[32]=RLS[31] XOR L[31]

RLS[31] =11011110010100100000100100110001

L[31] =11110101000010001101101000100110⊕

L[32] = 00101011010110101101001100010111

R[32] = R[31] Sebelum di proses

R[32] = 00111101011101011101101110011110

Tahap 6 : Lakukan pembalikan penulisan biner R32 dan L32

R[32] = 01111001110110111010111010111100

L[32]= 11101000110010110101101011010100

Gabungkan kedua nilai biner dimulai dari nilai biner ke 1 R[32] hingga nilai biner ke 32 L[32]. Hasil penggabungan dapat dilihat sebagai di bawah ini :

01111001110110111010111010111100111010001100101101011010110101000

Berdasarkan pada nilai biner di atas, pisahkan nilai biner menjadi 8 kelompok, dimana setiap kelompok 8 bit. Adapun perubahan hasil enkripsi dari *plaintext* dapat dilihat pada tabel di bawah berikut :

Tabel 9. Biner *Ciphertext*

Biner	Nilai Desimal	Karakter
01111001	121	Y
11011011	219	Û
10101110	174	®
10111100	188	¼
11101000	232	È
11001011	203	Ë
01011010	90	Z
11010100	212	Ô

1. Penyisipan berdasarkan algoritma LSB-1

Pesan yang telah dienkripsikan akan disisipkan ke *citra cover* (media penyembunyi) dengan menggunakan metode

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

LSB-1 (*Least Significant Bit-1*). Untuk keperluan hitungan manual, penyisipan pesan dilakukan pada *citra* bewarna dengan resolusi 6 x 5 *pixel* dengan format jenis bitmap (bitmap). Adapun contoh kasus dalam penyisipan pesan teks ke *citra cover* sebagai berikut :

a. Citra Cover

Gambar 2. Citra Cover 6 x 5

Citra sampel yang digunakan kemudian diektraksi setiap nilai elemen *pixel*nya kedalam bentuk desimal. Proses ekstraksi mengambil nilai *Red Green Blue* (RGB) pada setiap *pixel*. Ektraksi nilai warna RGB menggunakan *software* matlab. Adapun nilai-nilai yang dihasilkan dari proses ektraksi citra sampel menggunakan *software* matlab dapat dilihat pada tabel berikut:

Tabel 10. Nilai RGB *Pixel* Sampel

Pixel	Warna	Citra Cover		Pixel	Warna	Citra Cover	
		Des	Biner			Des	Biner
	R	123	01111011		R	164	10100100
1	G	167	10100111	11	G	191	10111111
	B	152	10011000		B	182	10110110
	R	120	01111000		R	138	10001010
2	G	161	10100001	12	G	161	10100001
	B	147	10010011		B	153	10011001
	R	122	01111010		R	120	01111000
3	G	160	10100000	13	G	166	10100110
	B	147	10010011		B	153	10011001
	R	132	10000100		R	120	01111000
4	G	161	10100001	14	G	162	10100010
	B	54	00110110		B	150	10010110
	R	153	10011001		R	141	10001101
5	G	180	10110100	15	G	177	10110001
	B	171	10101011		B	167	10100111
	R	154	10011010		R	157	10011101
6	G	177	10110001	16	G	188	10111100
	B	169	10101001		B	180	10110100
	R	133	10000101		R	142	10001110
7	G	177	10110001	17	G	167	10100111
	B	162	10100010		B	161	10100001
	R	128	10000000		R	83	01010011
8	G	169	10101001	18	G	104	01101000
	B	155	10011011		B	99	01100011
	R	137	10001001		R	119	11000111
9	G	175	10101111	19	G	165	10100101
	B	162	10100010		B	152	10011000
	R	157	10011101		R	122	01111010
10	G	190	10111110	20	G	164	10100100
	B	179	10110011		B	152	10011000

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Pixel	Warna	Citra Cover		Pixel	Warna	Citra Cover	
		Des	Biner			Des	Biner
	R	149	10010101		R	150	10010110
21	G	185	10111001	26	G	120	01111000
	B	175	10101111		B	104	01101000
	R	147	10010011		R	161	10100001
22	G	179	10110011	27	G	185	10111001
	B	170	10101010		B	177	10110001
	R	119	11000111		R	99	01100011
23	G	144	10010000	28	G	132	10000100
	B	138	10001010		B	137	10001001
	R	0	00000000		R	155	10011011
24	G	17	00010001	29	G	150	10010110
	B	12	00001100		B	137	10001001
	R	149	10010101		R	122	01111010
25	G	152	10011000	30	G	141	10001101
	B	161	10100001		B	175	10101111

b. Kunci steganografi adalah kata **Budidarma** dan sebagai penanda akhir pesan adalah biner-biner dari karakter # Setiap karakter kunci dikonversi menjadi biner, kemudian masing-masing biner karakter di XOR-kan agar didapatkan kunci sebagai 8 bit.

Tabel 11. Nilai biner kunci steganografi

Karakter	Decimal	Biner
B	66	01000010
U	117	01110101
D	100	01100100
I	105	01101001
D	100	01100100
A	97	01100001
R	114	01110010
M	109	01101101
A	97	01100001

Masing –masing karakter di XOR-kan,sehingga :

B= 01000010
u= 01110101⊕
XOR= 00110111
d= 01100100 ⊕
XOR= 01010011
i= 01101001 ⊕
XOR= 00111010
d= 01100100⊕
XOR= 01011110

XOR = 01011110
a= 01100001⊕
XOR = 00111111
r = 01110010⊕
XOR = 01001101
m= 01101101⊕
XOR = 00100000
a=01100001⊕
XOR = 01000001

Berdasarkan hasil XOR di atas, maka kunci stegano yang digunakan adalah hasil XOR terakhir (XOR ke-8) yaitu **01000001.**
Biner karakter penanda akhir pesan(#) adalah **00100011.**
Kunci steganografi(hasil XOR ke -8) digabungkan dengan biner-biner data teks yang akan disembunyikan ke dalam citra cover serta biner penanda akhir pesan, sehingga biner keseluruhan dari data teks yang akan disembunyikan ke dalam citra cover adalah :
01000001 (kunci) 01111001 11011011 10101110 10111100
11101000 11001011 01011010 11010100 **00100011**
(biner penanda akhir pesan)

c. Proses Penukaran bit ke-7 (LSB-1)

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Proses selanjutnya adalah menukarkan bit ke-7 citra biner menggunakan metode LSB-1 dengan bit biner *ciphertext* hasil enkripsi menggunakan algoritma GOST sebelumnya. Proses penukuran 64 bit *ciphertext* dilakukan hingga bit 64 biner citra sampel sehingga menyisakan 8 bit yang tidak dilakukan penyisipan dan tidak mengalami perubahan nilai. Adapun proses penukaran bit citra biner dengan bit *ciphertext* dapat dilihat pada tabel di bawah ini:

Tabel 12. Proses Penyisipan Bit *Ciphertext*

Pixel	Warna	Citra Cover		Bit	Nilai warna Gambar Stegano		
		Des	Biner	Biner Cipher	Warna	Des	Biner
	R	123	01111011	0	R	121	01111001
1	G	167	10100111	1	G	167	10100111
	B	152	10011000	0	B	152	10011000
	R	120	01111000	0	R	120	01111000
2	G	161	10100001	0	G	161	10100001
	B	147	10010011	0	B	145	10010001
	R	122	01111010	0	R	120	01111000
3	G	160	10100000	1	G	162	10100010
	B	147	10010011	0	B	145	10010001
	R	132	10000100	1	R	134	10000110
4	G	161	10100001	1	G	163	10100011
	B	54	00110110	1	B	54	00110110
	R	153	10011001	1	R	155	10011011
5	G	180	10110100	0	G	180	10110100
	B	171	10101011	0	B	169	10101001
	R	154	10011010	1	R	154	10011010
6	G	177	10110001	1	G	179	10110011
	B	169	10101001	1	B	171	10101011
	R	133	10000101	0	R	133	10000101
7	G	177	10110001	1	G	179	10110011
	B	162	10100010	1	B	162	10100010
	R	128	10000000	0	R	128	10000000
8	G	169	10101001	1	G	171	10101011
	B	155	10011011	1	B	151	10011011
	R	137	10001001	1	R	139	10001011
9	G	175	10101111	0	G	173	10101101
	B	162	10100010	1	B	162	10100010
	R	157	10011101	0	R	157	10011101
10	G	190	10111110	1	G	190	10111110
	B	179	10110011	1	B	179	10110011
	R	164	10100100	1	R	166	10100110
11	G	191	10111111	0	G	189	10111101
	B	182	10110110	1	B	182	10110110
	R	138	10001010	0	R	140	10001000
12	G	161	10100001	1	G	163	10100011
	B	153	10011001	1	B	155	10011011
	R	120	01111000	1	R	122	01111010
13	G	166	10100110	1	G	166	10100110
	B	153	10011001	0	B	153	10011001
	R	120	01111000	0	R	120	01111000
14	G	162	10100010	1	G	162	10100010
	B	150	10010110	1	B	150	10010110
	R	141	10001101	1	R	143	10001111
15	G	177	10110001	0	G	177	10110001
	B	167	10100111	1	B	167	10100111
	R	157	10011101	0	R	157	10011101
16	G	188	10111100	0	G	192	10111100
	B	180	10110100	0	B	180	10110100
	R	142	10001110	1	R	142	10001110
17	G	167	10100111	1	G	167	10100111
	B	161	10100001	0	B	161	10100001

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Pixel	Warna	Citra Cover		Bit Biner Cipher	Nilai warna Gambar Stegano		
		Des	Biner		Warna	Des	Biner
	R	83	01010011	0	R	81	01010001
18	G	104	01101000	1	G	106	01101010
	B	99	01100011	0	B	97	01100001
	R	119	11000111	1	R	199	11000111
19	G	165	10100101	1	G	163	10100111
	B	152	10011000	0	B	152	10011000
	R	122	01111010	1	R	122	01111010
20	G	164	10100100	0	G	164	10100100
	B	152	10011000	1	B	154	10011010
	R	149	10010101	1	R	151	10010111
21	G	185	10111001	0	G	185	10111001
		175	10101111	1	B	175	10101111
	R	147	10010011	0	R	145	10010001
22	G	179	10110011	1	G	179	10110011
	B	170	10101010	1	B	170	10101010
	R	119	11000111	0	R	117	11000101
23	G	144	10010000	1	G	146	10010010
	B	138	10001010	0	B	136	10001000
	R	0	00000000	1	R	2	00000010
24	G	17	00010001	0	Gs	17	00001100
	B	12	00001100	0	B	12	00001100
	R	149	10010101	0	R	149	10010101
25	G	152	10011000	0	G	152	10011000
	B	161	10100001	1	B	163	10100011
	R	150	10010110	0	R	148	10010100
26	G	120	01111000	0	G	120	01111000
	B	104	01101000	0	B	104	01101000
	R	161	10100001	1	R	163	10100011
27	G	185	10111001	1	G	187	10111011
	B	177	10110001	-	B	177	10110001
	R	99	01100011	-	R	99	01100011
28	G	132	10000100	-	G	132	10000100
	B	137	10001001	-	B	137	10001001
	R	155	10011011	-	R	155	10011011
29	G	150	10010110	-	G	150	10010110
	B	137	10001001	-	B	137	10001001
	R	122	01111010	-	R	122	01111010
30	G	141	10001101	-	G	141	10001101
	B	175	10101111	-	B	175	10101111

Proses penyisipan bit berhenti pada nilai *bit* ke 180 yaitu pada *pixel* ke 22 nilai warna R (Red). Penyisipan bit berhenti pada nilai tersebut dikarenakan jumlah karakter dari *ciphertext* hanya 8 karakter, dimana 8 karakter sama dengan 64 bit (1 karakter 8 bit) sedangkan setiap *pixel* memiliki 3 nilai warna yaitu RGB, dimana setiap nilai memiliki jumlah 8 bit biner. Sehingga didapatkan nilai-nilai *pixel* dari citra *stegano* seperti tabel berikut:

Tabel 13. Nilai Citra Stegano

Pixel	Nilai warna Gambar Stegano			Pixel	Nilai warna Gambar Stegano		
	Warna	Des	Biner		Warna	Des	Biner
	R	121	01111001		R	139	10001011
1	G	167	10100111	9	G	173	10101101
	B	152	10011000		B	162	10100010
	R	120	01111000		R	157	10011101
2	G	161	10100001	10	G	190	10111110
	B	145	10010001		B	179	10110011
	R	120	01111000		R	166	10100110
3	G	162	10100010	11	G	189	10111101
	B	145	10010001		B	182	10110110
4	R	134	10000110	12	R	140	10001000

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Pixel	Nilai warna Gambar Stegano			Pixel	Nilai warna Gambar Stegano		
	Warna	Des	Biner		Warna	Des	Biner
	G	163	10100011		G	163	10100011
	B	54	00110110		B	155	10011011
	R	155	10011011		R	122	01111010
5	G	180	10110100	13	G	166	10100110
	B	169	10101001		B	153	10011001
	R	154	10011010		R	120	01111000
6	G	179	10110011	14	G	162	10100010
	B	171	10101011		B	150	10010110
	R	133	10000101		R	143	10001111
7	G	179	10110011	15	G	177	10110001
	B	162	10100010		B	167	10100111
	R	128	10000000		R	157	10011101
8	G	171	10101011	16	G	192	10111100
	B	151	10011011		B	180	10110100

Tabel Lanjutan 13 Nilai Citra Stegano

Pixel	Nilai warna Gambar Stegano			Pixel	Nilai warna Gambar Stegano		
	Warna	Des	Biner		Warna	Des	Biner
	R	142	10001110		R	117	11000101
17	G	167	10100111	23	G	146	10010010
	B	161	10100001		B	136	10001000
	R	81	01010001		R	2	00000010
18	G	106	01101010	24	G	17	00001100
	B	97	01100001		B	12	00001100
	R	199	11000111		R	149	10010101
19	G	163	10100111	25	G	152	10011000
	B	152	10011000		B	163	10100011
	R	122	01111010		R	148	10010100
20	G	164	10100100	26	G	120	01111000
	B	154	10011010		B	104	01101000
	R	142	10001110		R	163	10100011
21	G	167	10100111	27	G	187	10111011
	B	161	10100001		B	-	
	R	81	01010001		R	-	
22	G	106	01101010	28	G	-	
	B	97	01100001		B	-	

Bila dibandingan dengan nilai-nilai elemen warna masing-masing *pixel* pada citra *cover*, maka ada beberapa *pixel* yang mengalami perubahan nilai setelah disisipkan biner *ciphertext*. Perubahan nilai tersebut dikarenakan terjadinya perubahan bit ke-7 dari setiap elemen nilai *pixel* citra *cover*. Nilai elemen warna akan turun 2 bit apabila nilai ke-7 dari citra *cover* adalah 1 dan bit *ciphertext* yang disisipkan bernilai 0. Sedangkan nilai elemen warna *pixel cover* akan bertambah apabila nilai bit ke-7 dari citra *cover* adalah 0 dan bit *ciphertext* yang disisipkan bernilai 1. Namun penurunan dan penambahan nilai bit tidak begitu signifikan terhadap perubahan warna citra stegano bila dilihat secara *visual* dengan mata manusia.

Tabel 14. Perubahan Nilai Warna Pada Beberapa Pixel Citra Cover dan Citra Stegano

Pixel	Warna	Citra Cover	
		Des	Biner
1	R	123	01111011
	G	167	10100111
	B	152	10011000
2	R	120	01111000
	G	161	10100001
	B	147	10010011
3	R	122	01111010
	G	160	10100000
	B	147	10010011
4	R	132	10000100
	G	161	10100001
	B	54	00110110

Pixel	Nilai warna Gambar Stegano		
	Warna	Des	Biner
1	R	121	01111001
	G	167	10100111
	B	152	10011000
2	R	120	01111000
	G	161	10100001
	B	145	10010001
3	R	120	01111000
	G	162	10100010
	B	145	10010001
4	R	134	10000110
	G	163	10100011
	B	54	00110110

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Berdasarkan pada gambar di atas dapat diambil kesimpulan bahwa terjadi perbubahan beberapa elemen warna pada *pixel*.

d. Simpan kedalam media penyimpanan hasil penukaran bit bit tersebut menjadi citra yang baru tetapi telah menyembunyikan data teks di dalam (*stegano image*).

Gambar 3. Citra Stegano

Proses untuk mengungkap kembali data teks yang tersembunyi dari dalam citra *stegano* (ekstrasi/*extraction*), di awali dengan pengambilan nilai-nilai bit ke-7 dari masing-masing elemen warna setiap *pixel* citra *stegano.*
Biner-biner hasil pengambilan tersebut akan dikonversi menjadi karakter hingga membentuk rangkaian data teks. Adapun langkah-langkah yang dilakukan untuk ekstraksi adalah:

a. Siapkan citra stegano

Gambar 4. Citra Stegano yang akan Diekstraksi

Nilai-nilai elemen warna masing-masing pixel dari citra stegano yang telah dipilih dikoversikan menjadi biner.

b. Masukan kunci steganografi, kunci dikonversikan ke biner kemudian masing-masing karakter kunci di XOR-kan.
Kunci steganografi yang diimput adalah sama seperti kunci yang digunakan pada proses *embedding* yaitu **Budidarma**.
Kata **Budidarma** dikoversi ke biner, kemudian masing-masing karakternya di XOR, prosesnya sama seperti pada proses *embedding*. hasil XOE kunci adalah **01000001**.

2. Ekstraksi Citra Stegano
Proses untuk mengungkapkan kembali data teks yang tersembunyi dari dalam citra stegano (ekstraksi/*extraction*), di awali dengan pengambilan nilai nilai bit ke 7 dari masing masing elemen warna setian *pixel* citra stegano. Biner-biner hasil pengambilan tersebut akan dikonversi menjadi karakter hingga membentuk rangkaian data teks.
Adapun langkah-langkah yang dilakukan untuk ekstraksi citra stegano adalah sebagai berikut :

Tabel 15. Proses Ekstraksi Nilai Citra Stegano

Pixel	Nilai warna Gambar Stegano			Biner LSB-1 Yang Diambil	Keterangan
	Warna	Des	Biner		
1	R	121	01111001	0	Ini adalah kunci steganografi
	G	167	10100111	1	**01000001**
	B	152	10011000	0	
2	R	120	01111000	0	
	G	161	10100001	0	
	B	145	10010001	0	

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Pixel	Nilai warna Gambar Stegano			Biner LSB-1 Yang Diambil	Keterangan
	Warna	Des	Biner		
3	R	120	01111000	0	
	G	162	10100010	1	
	B	145	10010001	0	**01111001**
4	R	134	10000110	1	bila biner ini di konversikan
	G	163	10100011	1	menjadi karakter maka tidak
	B	54	00110110	1	berkolerasi dengan karakter
5	R	155	10011011	1	#(bukan akhir pesan)oleh karena
	G	180	10110100	0	itu proses pengambilan LSB-1
	B	169	10101001	0	dilanjutkan
6	R	154	10011010	1	
	G	179	10110011	1	**11011011**
	B	171	10101011	1	bila biner ini di konversikan
7	R	133	10000101	0	menjadi karakter maka tidak
	G	179	10110011	1	berkolerasi dengan karakter
	B	162	10100010	1	#(bukan akhir pesan)oleh karena
8	R	128	10000000	0	itu proses pengambilan LSB-1
	G	171	10101011	1	dilanjutkan
	B	151	10011011	1	

Proses pengambilan bit LSB-1 serta pengujian masing-masing kelompok bit untuk *pixel* ke-9 sampai dengan *pixel* ke-23 dilakukan dengan cara yang sama.

Tabel Lanjutan 15. Proses Ekstraksi Nilai Citra Stegano

Pixel	Nilai warna Gambar Stegano			Biner LSB-1 Yang Diambil	Keterangan
	Warna	Des	Biner		
	R	2	00000010	1	…..100
24	G	17	00001100	0	
	B	12	00001100	0	
	R	149	10010101	0	**00100011**
25	G	152	10011000	0	Kelompok biner ini bila
	B	163	10100011	1	dikonversikan menjadi
	R	148	10010100	0	karakter, maka
26	G	120	01111000	0	memiliki kesamaan
	B	104	01101000	0	dengan karakter
	R	163	10100011	1	penanda akhir pesan(#).
27	G	187	10111011	1	Proses pengambilan bit
	B	177	10110001		LSB-1
					DIBERHENTIKAN

Proses pengambilan nilai LSB-1 ini dilakukan dengan mengambil bit ke-7 dari masing-masing elemen warna *pixel* citra *stegano* yang di mulai dari posisi bit ke -7. Pengambilan bit berhenti pada nilai bit ke 80 yaitu pada *pixel* ke 27 nilai warna G (*Green*).
8 bit awal sebagai kunci stegano san 8 bit akhir sebagai karakter penanda tidak dijadikan sebagai bagian dari bit-bit data teks yang diungkap, sehingga
biner – biner data teks yang di ungkap seluruhnya adalah :
01111001 11011011 10101110 10111100 11101000 11001011 01011010 11010100

3. Dekripsi Berdasarkan Algoritma GOST
   Berdasarkan hasil ekstraksi dari citra stegano menggunakan algoritma LSB-1 dengan mengambil nilai biner ke-7 dari nilai *pixel* sampel di dapati 8 kelompok biner *ciphertext* sebagai berikut :
   Ciphertext : 01111001 11011011 10101110 10111100 11101000 11001011 01011010 11010100

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Proses dekripsi menggunakan algoritma GOST menggunakan kunci yang sama saat proses enkripsi.
Bagi 2 *ciphertext* dari kiri L0 dan R0
L[0] = 00101011010110101101001100010111
R[0] = 00111101011101011101101110011110

a. Putaran 0
Tahap 1 :L[0]= 00101011010110101101001100010111
R[0]= 00111101011101011101101110011110→rubah ke decimal
Tahap 2 :(R[0] + K[0]) Mod 2^{23}
R[0] = 1031134110
K[0] = 4142282370 +
Hasil = 5173416480 mod $2^{23(4294967296)}$
Hasil mod= 878449184→konversikan ke biner
Biner = 0011 0100 0101 1100 0001 0010 0010 0000
Tahap 3:kelompokkan biner hasil di atas menjadi 4 bit

Tabel 16. S-Box round 0

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
0011	3	→S-Bok(1)	2	0010
0100	4	→S-Bok(2)	6	0110
0101	5	→S-Bok(3)	3	0011
1100	12	→S-Bok(4)	11	1011
0001	1	→S-Bok(5)	12	1100
0010	2	→S-Bok(6)	10	1010
0010	2	→S-Bok(7)	4	0100
0000	0	→S-Bok(8)	1	0001

Tahap 4 :Hasil gabungan : 00100110001110111100101001000001
Hasil RLS [0] 11 Bit : 11011110010100100000100100110001
Tahap 5 :R[1] =RLS[0] XOR L[0]
RLS[0]=11011110010100100000100100110001
L[0] =00101011010110101101001100010111⊕
R[1] =11110101000010001101101000100110
L[1] = R[0] Sebelum di proses
L[1] = 00111101011101011101101110011110

Tahap 6 :Hasil Putaran -0 atau pada PUTARAN – I =0, adalah :
L[1]=00111101011101011101101110011110
R[1]=11110101000010001101101000100110

b. Putaran 1
Tahap 1 :L[1]= 00111101011101011101101110011110
R[1]= 11110101000010001101101000100110→rubah ke decimal
Tahap 2: (R[1] + K[1]) Mod 2^{23}
R[1]= 4110998054
K[1]= 3056506446 +
Hasil= 7167504500 mod $2^{23(4294967296)}$
Hasil mod= 2872537204→konversikan ke biner
Biner = 1010 1011 0011 0111 0111 0000 0111 0100
Tahap 3 :kelompokkan biner hasil di atas menjadi 4 bit

Tabel 17. S-Box round 1

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
1010	10	→S-Bok(1)	1	0001
1011	11	→S-Bok(2)	1	0001
0011	3	→S-Bok(3)	13	1101
0111	7	→S-Bok(4)	15	1111
0111	7	→S-Bok(5)	8	1000
0000	0	→S-Bok(6)	4	0100
0111	7	→S-Bok(7)	9	1001
0100	4	→S-Bok(8)	5	0101

Tahap 4 :Hasil gabungan : 00010001110111111000010010010101

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Hasil RLS [1] 11 Bit : 11111100001001001010100010001110
Tahap 5 :R[2]=RLS[1] XOR L[1]
RLS[1] =11111100001001001010100010001110
L[1] =00111101011101011101101110011110⊕
R[2] =11000001010100010111001100010000
L[2] =R[1] Sebelum di proses
L[2] =11110101000010001101101000100110
Tahap 6 :Hasil Putaran -0 atau pada PUTARAN – I =0, adalah :
L[2] =11110101000010001101101000100110
R[2] =11000001010100010111001100010000

c. Putaran 2
Tahap 1: L[2] =11110101000010001101101000100110
R[2] =11000001010100010111001100010000 →rubah ke decimal
Tahap 2 :(R[2] + K[2] Mod 2^{23}
R[2] = 3243340560
K[2] = 4074961542+
Hasil =7318302102 mod $2^{23(4294967296)}$
Hasil mod = 3023334806 →konversikan ke biner
Biner = 1011 0100 0011 0100 0110 1101 1001 0110
Tahap 3 : kelompokkan biner hasil di atas menjadi 4 bit

Tabel 18. S-Box round 2

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
1011	11	→S-Bok(1)	12	1100
0100	4	→S-Bok(2)	6	0110
0011	3	→S-Bok(3)	13	1101
0100	4	→S-Bok(4)	0	0000
0110	6	→S-Bok(5)	2	0010
1101	13	→S-Bok(6)	12	1100
1001	9	→S-Bok(7)	10	1010
0110	6	→S-Bok(8)	10	1010

Tahap 4 : Hasil gabungan : 11000110110 100000010110010101010
Hasil RLS [3] 11 Bit : 10000001011001010101011000110110
Tahap 5 :R[3] = RLS[1] XOR L[1]
RLS[2] = 10000001011001010101011000110110
L[2] = 11110101000010001101101000100110⊕
R[3] = 01110100011011011000110000010000
L[3] = R[2] Sebelum di proses
L[3] = 11000001010100010111001100010000
Tahap 6:Hasil Putaran -2atau pada PUTARAN – I =2, adalah :
L[3] = 11000001010100010111001100010000
R[3] = 01110100011011011000110000010000

Proses berikutnya untuk ronde ke 3 hingga ronde ke 30 dilakukan dengan cara yang sama seperti proses sebelumnya. Sehingga berdasarkan hitungan manual keseluruhan ronde 3 hingga ronde ke 30 dapat dilihat pada tabel di bawah ini :

Tabel 19. Hasil Dekripsi Ronde 3 Sampai Dengan 30

RONDE	L[i] dan R[i]	Nilai Biner L[i] dan R[i]
RONDE 3	L[3]	11000001010100010111001100010000
	R[3]	11110101000010001101101000100110
RONDE 4	L[4]	01110011111011011000110000010000
	R[4]	11000001010100010111001100010000
RONDE 5	L[5]	00111110000101010110111111010100
	R[5]	01110011111011011000110000010000
RONDE 6	L[6]	11101010111001110001010001111010
	R[6]	00111110000101010110111111010100
RONDE 7	L[7]	01110001100101110011000010100001
	R[7]	10101011001110001110000010001101
RONDE 8	L[8]	00111010110001101100010010011111

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

RONDE	L[i] dan R[i]	Nilai Biner L[i] dan R[i]
RONDE 9	R[8]	01110001100101110011000010100001
	L[9]	11100101110000010000110001111111
	R[9]	00111010110001101100010010011111
RONDE 10	L[10]	11000010011010100101010000011011
	R[10]	11100101110000010000110001111111
RONDE 11	L[11]	01011100000000010111001011101011
	R[11]	11000010011010100101010000011011
RONDE 12	L[12]	01010110010110100010111100111110
	R[12]	01011100000000010111001011101011
RONDE 13	L[13]	01010110010110100010111100111110
	R[13]	01011100000000010111001011101011
RONDE 14	L[14]	11011011111111000011100011010010
	R[14]	01010110010110100010111100111110
RONDE 15	L[15]	01010100010110100011001000000010
	R[15]	11011011111111000011100011010010
RONDE 16	L[16]	10101011111000111011001010010101
	R[16]	01010100010110100011001000000010
RONDE 17	L[17]	01110001111011111001001010100001
	R[17]	10101011111000111011001010010101
RONDE 18	L[18]	11100001010001011001011010000111
	R[18]	01110001111011111001001010100001
RONDE 19	L[19]	00010001111111111110111011110010
	R[19]	11100001010001011001011010000111
RONDE 20	L[20]	10101101010101011101101100010110
	R[20]	00010001111111111110111011110010
RONDE 21	L[21]	10101111001000001101100101000011
	R[21]	10101101010101011101101100010110
RONDE 22	L[22]	11001110010010110000001101101010
	R[22]	10101111001000001101100101000011
RONDE 23	L[23]	00111110010001101101001010011011
	R[23]	11001110010010110000001101101010
RONDE 24	L[24]	10001000110100100001001011011011
	R[24]	00111110010001101101001010011011
RONDE 25	L[25]	11100100111100010000110111010000
	R[25]	10001000110100100001001011011011
RONDE 26	L[26]	11110111100110000100000101010001
	R[26]	11100100111100010000110111010000
RONDE 27	L[27]	11101010111001110001010001111010
	R[27]	00111110000101010110111111010100
RONDE 28	L[28]	01011000110001010100111010110011
	R[28]	01001100100110000010101011111010
RONDE 29	L[29]	11010101000010011011110110110001
	R[29]	01011000110001010100111010110011
RONDE 30	L[30]	11000001010100010111001100010000
	R[30]	11110101000010001101101000100110
RONDE 31	L[31]	11110010010000100010111110001000
	R[31]	00100010100000101001001010110010

d. Putaran 31:
Tahap :L[31]=11110010010000100010111110001000
R[31]=00100010100000101001001010110010→rubah ke decimal
Tahap 2 :(R[31] + K[0] Mod 2^{23}
R[31] = 578982578
K[0] = 4142282370 +
Hasil = 4721264948 mod $2^{23(4294967296)}$
Hasil mod = 426297652 → konversikan ke biner
Biner = 0001 1001 0110 1000 1100 1001 0011 0100
Tahap 3 :kelompokkan biner hasil di atas menjadi 4 bit

TIN: Terapan Informatika Nusantara
Vol 1, No 7, Desember 2020, Hal 363-382
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Tabel 20. S-Box round 31

Biner Kelompok	Dec nilai bit	S-Box	Hasil permutasi dengan S-Box	Biner
0001	1	→S-Bok(1)	10	1010
1001	9	→S-Bok(2)	3	0011
0110	6	→S-Bok(3)	4	0100
1000	8	→S-Bok(4)	14	1110
1100	12	→S-Bok(5)	0	0000
1001	9	→S-Bok(6)	6	0110
0011	3	→S-Bok(7)	1	0001
0100	4	→S-Bok(8)	5	0101

Tahap 4 :Hasil gabungan :10100011010011100000011000010101
Hasil RLS [31] 11 Bit :01110000001100001010110100011010
Tahap 5 :R[32] = R[31] Sebelum di proses
R[32] = 00100010100000101001001010110010
Tahap 6 :R[32]=RLS[31] XOR L[31]
RLS[31]=01110000001100001010110100011010
L[31] =11110010010000100010111110001000⊕
L[32] =10000010011100101000001010010010
Tahap 7 :Lakukan pembalikan penulisan biner2 R32 dan L32
L[32] = 01001101010010010100000101000100
R[32] = 01001101010010010100000101000100
Gabungankan hasil dekripsi R[32] dan L[32] seperti di bawah ini :
01001101010010010100000101000100010011010100100101000001010001 00
Pisahkan bit menjadi 8 kelompok, dimana setiap kelompok memiliki 8 bit. Kemudian rubah kedalam bentuk decimal dan karakter menggunakan kode ASCII. Adapun hasil dekripsi dapat dilihat seperti tabel di bawah ini :

Tabel 21. Nilai Plaintext

Biner	Desimal	Karakter
01001101	77	M
01001001	73	I
01000001	65	A
01000100	68	D
010010s01	73	I
01000001	65	A
01001110	78	N
01000001	65	A

Berdasarkan tabel di atas di dapat *plaintext* hasil dekripsi menggunakan algoritma GOST yaitu MIADIANA.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan prosedur metode GOST dengan metode LSB-1 dilakukan secara berurut, dimana mode operasi metode GOST digunakan untuk menyandikan data teks yang akan disembunyikan, sedangkan metode LSB-1 digunakan untuk menyembunyikan biner-biner cipherteks kedalam citra digital. Hasil penerapan kedua metode ini adalah dalam bentuk citra yang secara penglihatan mata manusia tidak terdapat perbedaan yang signifikan dengan citra aslinya. Penyembunyian biner-biner pesan kedalam citra digital dilakukan dengan mengganti bit-bit *pixe*l citra *cover* dengan bit-bit data teks yang telah disandikan sebelumnya. Posisi bit citra yang digantikan adalah bit-bit pixel citra yang berada pada posisi ke-7 atau bit paling terkahir-1(least significant bit-1). Nilai elemen warna akan turun 2 bit apabila nilai ke-7 dari citra *cover* adalah 1 dan bit *ciphertext* yang disisipkan bernilai 0. Sedangkan nilai elemen warna *pixel cover* akan bertambah apabila nilai bit ke-7 dari citra *cover* adalah 0 dan bit *ciphertext* yang disisipkan bernilai 1.

REFERENCES

Zebua. T," Penerapan metode LSB-2 untuk Menyembunyikan Ciphertext pada citra digital", Pelita Informatika Budi Darma, vol. X, pp. 135-140, 2015

Hendri, "Pengamanan aplikasi chatting menggunakan metode kriptografi Government standard", Majalah Ilmiah INTI, Volume 12, pp. 295-300, 2017

Darmayanti, Harsa" Sistem Steganografi pada Citra Digital menggunakan Least Significant Bit", Prosiding Seminar Sains dan Teknologi FMIPA Unmul Samarinda, Volume 1, pp 51-56. 2016

Krisnawati"Metode Least Sugnificant Bit (LSB) dan end of file (EOF) untuk Menyisipkan teks ke dalam Citra Grayscale", Seminar Nasional Informatika 2008, pp.49-44. 2008

Doni Ariyus 2006, pengantar ilmu kriptografi, informatika, Bandung

Setyaningsih.E, 2015, Kriptografi &Implementasi menggunakan MATLAB, Andi publisher, Yogyakarta

A.A. Ibrahim, "Perancangan Pengamanan Data Menggunakan Algoritma AES (*Advanced Encryption Standard*)", Teknik Informatika STMIK Antar Bangsa, vol. III, pp.53-60, 2017

Takur, J, Kumar,N,"DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis" International Journal of Emerging Technology and Advanced Engineering, vol,1, Issue 2, December 2011

E.R. Agustina and A. Kurniati, "Pemanfaatan Kriptografi Dalam Mewujudkan Keamanan Informasi Pada *e-Voting* di Indonesia", Seminar Nasional Informatika, pp.22-28, 2009

Muhammad Iqbal1, Yudi Sahputra2, Andysah Putera Utama Siahaan3, "The Understanding of GOST Crytography Technique , International Journal of Engineering Trends and Technology (IJETT) – Volume 39 Number 3- September 2016

Munir, R 2006,"Kriptografi,Informatika",Bandung

Ludmila Babenko, Ekaterina Maro, "Algebraic Cryptanalysis of GOST Encryption Algorithm" Journal of Computer and Communications, 2014, 2, 10-17

Nosrati, Masoud, An introduction to steganography methods," World Applied Programming, vol 1, pp 191-195, August 2011.

Zebua.T, Pengamanan data teks dengan kombinasi Cipher Blok Chaining LSB-1". Seminar Nasional Inovasi dan Teknologi Informasi, pp 85-89,2015

Rosa, A.S, Shalahuddin, M, Modul Pembelajaran Rekayasa Perangkat Lunak, Bandung Modula. 2015