Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

Penerapan Algoritma Gost Dan Mode Output Feedback Untuk Tingkat Keamanan Citra Digital

Sri Wahyuni, Surya Darma Nasution, Taronisokhi Zebua

Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma, Medan, Indonesia
Email: [1]swayuni2398@gmail.com

Abstrak−Citra digital yang bersifat pribadi dan rahasia sangat rentan terhadap penyadapan oleh pihak-pihak yang tidak bertanggung jawab. Terutama bila didistribusikan melalui jaringan internet seperti pada aplikasi berbasis chatting, whatsapp dan media e-mail. Citra yang dikirim terkadang merupakan citra yang bersifat rahasia dan harus dijaga keamananya. Demi menjaga keamanan citra digital dapat dilakukan dengan pemanfaatan teknik kriptografi. Teknik kriptografi dapat mengamankan citra digital dengan merubah nilai-nilai pixel dari citra digital sehingga menghasilkan nilai pixel yang berbeda dari citra asli yang akan diamankan. Adapun dalam penelitian ini algoritma yang digunakan adalah algoritma GOST dengan menambahkan teknik OFB untuk mengoptimalkan kunci GOST, serta hasil enkripsi dapat dikirim melalui aplikasi yang akan dirancang dengan model client-server. Pembuatan aplikasi perangkat lunak enkripsi dan dekripsi berbasis client-server, bertujuan untuk transfer data digital berupa plainimage yang sudah dienkripsi menggunakan algoritma GOST, serta dapat didistribusikan secara local menggunakan ip address yang terhubung dalam satu jaringan yang sama pada komputer pengirim dan penerima (client-server).

Kata Kunci: Kriptografi; Citra; GOST; Output Feedback (OFB); Client-Server

Abstract−Digital images that are private and confidential are very vulnerable to wiretapping by irresponsible parties. Especially if it is distributed over the internet such as chat-based applications, WhatsApp and e-mail media. The image that is sent is sometimes an image that is confidential and must be protected. In order to maintain the security of digital images, cryptographic techniques can be used. Cryptographic techniques can secure digital images by changing the pixel values of the digital image so that it produces a different pixel value from the original image to be secured. In this study, the algorithm used is the GOST algorithm by adding the OFB technique to optimize the GOST key, and the encryption results can be sent via an application that will be designed with a client-server model. Creating a client-server-based encryption and decryption software application, aimed at transferring digital data in the form of a plain image that has been encrypted using the GOST algorithm, and can be distributed locally using an IP address connected to the same network on the sending and receiving computers (client-server).

Keywords: Cryptography; Images; GOST; Output Feedback (OFB); Client-Server

1. PENDAHULUAN

Saat ini penggunaan media sosial atau yang sering dikenal dengan istilah sosmed dan semakin banyak digunakan sebagai alat dalam berkomunikasi. Citra digital merupakan salah satu objek yang sering didistribusikan melalui media komunikasi. Citra adalah suatu gambar bersifat visual yang dimiliki oleh seseorang mengenai pribadi, organisasi ataupun produk. Informasi dari sebuah citra digital yang didistribusikan dapat saja bersifat rahasia atau penting, misalnya foto KTP dan citra digital lainnya. Apabila citra digital yang didistribusikan tersebut tidak diamankan, maka dapat memudahkan pihak lain (penyerang) untuk menyadap, mencuri ataupun memanipulasi citra digital tersebut untuk tujuan yang merugikan pemilik.

Berdasarkan penelitian terdahulu, menyebutkan bahwa citra digital yang bersifat pribadi dan rahasia sangat rentan terhadap peyadapan oleh pihak-pihak yang tidak bertanggung jawab, terutama bila citra tersebut didistribusikan melalui jaringan internet seperti pada aplikasi pengiriman berbasis *chatting facebook*, *whatsapp* dan media *e-mail* [1]. Penelitian lain mengatakan bahwa citra yang dikirim melalui aplikasi sangat rawan terhadap penyerangan dan penyadapan sehingga perlu diamankan terlebih dahulu sebelum didistribusikan, serta penyimpanan yang dilakukan di dalam media *stroge* rawan terhadap pengaksesan oleh orang-orang yang tidak memiliki wewenang [2]. Oleh karena itu, maka sangat perlu dilakukan pengamanan dan penjagaan terhadap citra sebelum didistribusikan pada media komunikasi.

Salah satu solusi yang dapat dilakukan untuk menyelesaikan permasalahan di atas adalah mengoptimalkan pengamanan citra digital yang sifatnya pribadi atau rahasia khususnya yang didistribusikan melalui media komunikasi yang digunakan. Salah satu teknik yang dapat digunakan adalah teknik kriptografi. Teknik kriptografi merupakan sebuah teknik yang mempelajari enkripsi dimana data/informasi diacak menggunakan suatu kunci enkripsi, sehingga data menjadi sulit dibaca bila tidak memiliki kunci deskripsi [3]. Salah satu algoritma kriptografi yang dapat digunakan dalam mengamankan data adalah algoritma GOST

Algoritma *Gosudarstvenny Standart* (GOST) adalah algoritma yang dikembangkan oleh pemerintah Uni Soviet pada saat perang dingin. Algoritma ini merupakan algoritma enkripsi sederhana yang memiliki panjang kunci 256-bit atau 64 *byte* setara dengan 32 karakter, menggunakan operasi XOR dan *Left Circular Shif* 11 bit atau *Rotate Left Shif* (RLD 11 bit) dan juga menggunakan modulo 2^{32}. Cara kerja algoritma GOST adalah dengan mengkonversikan *plainteks* dan kunci ke dalam bilangan biner kemudian akan dilakukan proses enkripsi sebanyak 32 *round* (putaran) [4]. Kelemahan GOST yang diketahui adalah karena *key schedule* nya yang sederhana, sehingga ini menjadi titik lemah terhadap kriptanalisis [4].

Output Feedback (OFB) adalah mode operasi untuk *cipher block*, memiliki beberapa kesamaan dengan mode *ciphertext feedback* yang memungkinkan mengenkripsi ukuran blok yang berbeda, tetapi memiliki perbedaan utama bahwa *output* dari fungsi blok enkripsi adalah umpan balik (bukan *ciphertext)*. Mode operasi ini berkerja dengan

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

menggunakan skema umpan balik dengan mengaitkan *block plainteks* bersama-sama sedemikian sehingga *ciphertext* bergantung pada skema blok *plainteks* sebelumnya dan deskripsi dilakukan sebagai kebalikan proses enkripsi [5]. Mode *Output Feedback* (OFB) hampir sama dengan mode operasi CFB, namun dalam OFB tiap *bit* teks sandi tidak tergantung *bit* sebelumnya, sehingga mode operasi *Output Feedback* dapat digunakan sebagai sandi *stream* untuk meningkatkan algoritma keamanan data [6].

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kata kriptografi berasal dari dua kata yaitu, *kryptos* yang berarti rahasia dan *graphien* yang berarti tulisan. Jadi, kriptografi adalah sebuah teknik yang mempelajari enkripsi dimana data/informasi diacak menggunakan suatu kunci enkripsi, sehingga data menjadi sulit dibaca bila orang tersebut tidak memiliki kunci deskripsi. Atau dapat diartikan dengan mudah kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan [7].

Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi *ciphertex* . Enkripsi dan deskripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan P menyatakan *plaintext* dan C menyatakan *ciphertext*, maka fungsi engkripsi E memetakan P ke C.

Keamanan telah menjadi aspek yang sangat penting dari suatu sistem informasi yang umumnya hanya ditunjukan bagi segolongan tertentu, karena itu penting untuk melindungi system informasi tersebut demi mencegahnya jatuh kepada pihak-pihak lain yang tidak berkepentingan. Salah satu upaya pengaman sistem informasi yang dapat dilakukan adalah kriptografi dengan beberapa aspek keamanan informasi [8], yaitu :

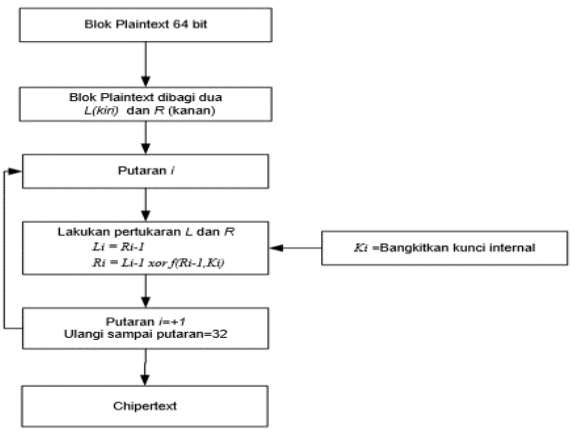
1. Kerahasiaan (*confidentiality*)
   Merupakan layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. Integritas Data (*integrity*)
   Merupakan layanan yang menjamin bahwa pesan masih utuh/asli atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*)
   Merupakan layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (*data origin authentication*).
4. Nir penyangkalan (*non repudiation*)
   Merupakan layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

2.2 Citra Digital

Citra adalah suatu gambar bersifat visual yang dimiliki oleh seseorang mengenai pribadi, organisasi ataupun produk. Citra digital juga dapat didefinisikan sebagai kumpulan piksel-piksel yang disusun dalam larik dua-dimensi, di mana piksel adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat [9]. Citra digital diperoleh dengan cara mengukur warna pada sebuah citra pada titik-titik pada citra dan merpresentasikannya ke dalam bentuk digital atau angka bilangan bulat.

2.3 Algoritma GOST

Algoritma *Gosudarstvenny Standart* (GOST) merupakan *block cipher* 64 bit dengan panjang kunci 256 bit. Cara kerja algoritma GOST adalah dengan mengkonversikan *plaintext* dan kunci ke dalam bilangan biner kemudian akan dilakukan proses enkripsi sebanyak 32 *round* (putaran) [4]. Secara sederhana cara kerja algoritma GOST dapat di lihat pada gambar 1:



Gambar 1. Skema Cara Kerja Algoritma GOST

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

Setiap putaran, blok R (kanan) tidak akan mengalami perubahan apapun karena hanya akan dipindah menjadi blok L pada putaran selanjutnya. Namun blok R akan digunakan bersamaan dengan *subkey* kunci internal untuk diolah pada fungsi f dan akan di XOR-kan dengan blok L (kiri).
Terdapat 3 proses utama dalam algoritma GOST [4], yaitu :

1. Proses Pembentukan Kunci
2. Proses Enkripsi
3. Proses Dekripsi

2.4 Metode OFB

Output Feedback (OFB) adalah mode operasi untuk *cipher block*, memiliki beberapa kesamaan dengan mode *ciphertext feedback* yang memungkinkan mengenkripsi ukuran blok yang berbeda, tetapi memiliki perbedaan utama bahwa *output* dari fungsi blok enkripsi adalah umpan balik (bukan *ciphertext).* Mode operasi ini berkerja dengan menggunakan skema umpan balik dengan mengaitkan *block plaintext* bersama-sama sedemikian sehingga *ciphertext* bergantung pada skema blok *plaintext* sebelumnya dan deskripsi dilakukan sebagai kebalikan proses enkripsi [10].

3. HASIL DAN PEMBAHASAN

Citra adalah suatu gambar bersifat visual yang dimiliki oleh seseorang mengenai pribadi, organisasi ataupun produk. Saat ini, citra digital dapat didistribusikan bebas memulai jaringan internet berbasis *chatting.* Jaringan internet yang bersifat publik tentu dapat disadap dan susupi *malware*, sehingga data citra yang dikirim bisa saja diduplikat dan dimanfaatkan, sehingga distribusi citra rahasia secara publik melalui pengiriman *online* tentu memiliki resiko keamanan. Terutama citra digital bersifat rahasia yang belum disandikan, maka dengan sangat mudah dapat dimengerti karena citra tersebut masih berupa data yang asli sehingga dapat merugikan salah satu pihak pengirim.

Oleh sebab itu, maka dibutuhkanya sebuah teknik pengamanan yang dapat memanipulasi citra rahasia menjadi citra yang tidak dapat dikenali visualya, serta merancang dan membangun aplikasi yang siap mengirimkan citra digital hasil enkripsi berbasis *client-server* dengan jaringan lokal. Hal ini bertujuan untuk mengurangi resiko keamanan dalam proses pengiriman secara publik dan global dengan saluran internet.

Berdasarkan rumusan masalah pada bab sebelumnya dan paparan di atas, masalah yang terjadi adalah bagaimana sebuah citra digital yang belum disandikan dapat diamankan dengan teknik kriptografi serta didistribusikan secara lokal dengan bantuan aplikasi berbasis *client-server*.

Dimana aplikasi ini nantinya dapat menyimpan hasil enkripsi berupa *cipherimage* dan kemudian citra *cipherimage* tersebut dapat didistribusikan secara local menggunakan *ip address* yang terhubung dalam satu jaringan yang sama pada komputer pengirim dan penerima *(client-server)*.

Teknik kriptografi memerlukan algoritma dalam melakukan pengamanan file citra digital. Adapun dalam penelitian ini algoritma yang digunakan adalah algoritma GOST dengan menambahkan teknik OFB untuk mengoptimalkan kunci GOST, serta hasil enkripsi dapat dikirim melalui aplikasi yang akan dirancang dengan model *client-server*.

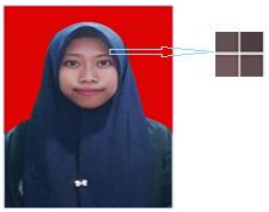
3.1 Penerapan Algoritma

Contoh kasus proses hitungan manual penerapan algoritma GOST dan mode operasi OFB untuk keamanan citra digital dengan *plainimage.* Dalam contoh kasus ini, diasumsikan bahwa terdapat komputer A yang berfungsi sebagi server (Pengirim) dan komputer B yang berfungsi sebagai *client* (Penerima). Komputer A akan mengirimkan sebuah citra digital rahasia yang akan dikirimkan ke komputer B. Adapun contoh citra digital rahasia yang akan diamankan dan dikirim kapada komputer B adalah sebagai berikut:

Gambar 2. *Plainimage* dengan resolusi 466 x 713

Berdasarkan pada gambar citra di atas, citra digital bewarna dengan ekstansi .jgp resolusi 466 x 713 dan *bitdepth* 24 *bit*. Adapun untuk keperluan hitungan manual maka diambil sampel citra berdasarkan citra asli dengan resolusi 2x2 pixel sebagai berikut:

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat



Gambar 3. *Plainimage* Sampel 4 *Pixel*

Berdasarkan pada gambar di atasm diperoleh nilai RGB dari 4 *pixel* sampel untuk perhitungan manual sebagai berikut:

Tabel 1. Nilai RBG *Pixel* Sampel

Pixel	Warna	Plain Desimal
1	R	160
	G	110
	B	79
2	R	110
	G	87
	B	104
3	R	80
	G	75
	B	150
4	R	97
	G	80
	B	120

Berdasarkan tabel 1 di atas, untuk mempermudah dalam hitungan manual algoritma GOST, maka penulis hanya mengambil nilai G pada *pixel* 3, sehingga nilai desimal dari *plainimage* adalah 160, 110, 79, 110, 87, 104, 80, 75. Sebelum melakukan proses enkripsi GOST terlebih dahulu melakukan pembangkitan kunci dengan metode OFB agar kunci GOST lebih optimal dalam mengamankan citra digital. Adapun *string* pembangkit kunci GOST berdasarkan mode OFB harus memiliki panjang karakter sebanyak 32 karakter (256 bit), sedangkan OFB juga memiliki kunci untuk mendapatkan karakter baru yang akan menjadi kunci asli GOST dengan panjang 8 karakter (64 bit). Berikut karakter *string* dan kunci OFB yang akan dieksekusi dengan mode OFB untuk mendapatkan kunci GOST:

String OFB : algoritma_gost_sri_wahyuni_2020_
Kunci OFB : 20202020

String OFB dan kunci OFB tersebut akan dikirimkan kepada penerima citra digital hasil enkripsi untuk membangkitan kunci asli GOST.

Langkah-langkah pemanfaatan mode operasi OFB pada algoritma GOST dalam mengamankan citra digital yaitu,:

1. Proses Pembentukan Kunci Berdasarkan OFB

a. Kelompokkan *String* OFB
String OFB = algoritma_gost_sri_wahyuni_2020_

Tabel 2. Konversi Kunci Awal

konversi kunci					
char	desimal	Biner	char	Desimal	Biner
a	97	0110 0001	r	114	0111 0010
l	108	0110 1100	i	105	0110 1001
g	103	0110 0111	_	95	0101 1111
o	111	0110 1111	w	119	0111 0111
r	114	0111 0010	a	97	0110 0001
i	105	0110 1001	h	104	0110 1000
t	116	0111 0100	y	121	0111 1001
m	109	0110 1101	u	117	0111 0101
a	97	0110 0001	n	110	0110 1110
_	95	0101 1111	i	105	0110 1001
g	103	0110 0111	_	95	0101 1111
o	111	0110 1111	2	50	0011 0010
s	115	0111 0011	0	48	0011 0000
t	116	0111 0100	2	50	0011 0010
_	95	0101 1111	0	48	0011 0000
s	115	0111 0011	_	95	101 1

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

b. Pengacakan Kunci Berdasarkan Mode Operasi OFB

Dalam algoritma GOST panjang kunci adalah 256 bit atau 32 karakter, untuk memudahkan proses enkripsi kunci menggunakan mode operasi OFB maka langkah yang dilakukan sebagai berikut :

1) Karena mode OFB beroperasi pada blok 64 pesan bit, maka bagi kunci menjadi 64 bit setiap kelompok dan gabungkan binernya, maka akan ada 4 kelompok :

Kelompok 1 : algoritm
Biner : 01100001011011000110011101101111011100100110100101110100011011 01
Kelompok 2 : a_gost_s
Biner : 01100001010111110110011101101111011100110111010001011111011100 11
Kelompok 3 : ri_wahyu
Biner : 01110010011010010101111101110111011000010110100001111001011101 01
Kelompok 4 : ni_2020_
Biner : 01101110011010010101111100110010001100000011001000110000010111 11

2) Kelompokkan kunci dengan antrian (*shift left register)* 8 *byte*:

Tabel 2. Binner yang Digunakan

Kunci awal	Posisi Bit	Biner yang diambil
P[1]	64...1	10110110 00101110 10010110 01001110
		11110110 11100110 00110110 10000110
P[2]	128..65	11001110 11111010 00101110 11001110
		11110110 11100110 11111010 10000110
P[3]	192...129	10101110 10011110 00010110 10000110
		11101110 11111010 10010110 01001110
P[4]	256...193	11111010 00001100 01001100 00001100
		01001100 11111010 10010110 01110110

3) Dalam pengoptimalan kunci menggunakan mode operasi OFB dibutuhkan kunci dengan panjang 64 bit.
Kunci (key) : 20202020
Desimal : 5048504850485048
Biner : 0011001000110000001100100011000000110010001100000011001000110000

4) Kemudian XOR -kan n-bit pertama dari *key* dengan *n-bit* terakhir *plainimage* (*Left-most byte).*
Persamaan nya : pi ⊕ ki = ci

P1 : 1011011000101110100101100100111011111011011100110001101101000011 0
Key : 0011001000110000001100100011000000110010001100000011001000110000
XOR
C1 : 0101001101011100010101010101111101000000010110010100011001011101
P2 : 1100111011111010001011101100111011110111011100110111110101000011 0
Key : 0011001000110000001100100011000000110010001100000011001000110000
XOR
C2 : 0101001101101111010101010101111101000001010001000110110101000011
P3 : 1010111010011110000101101000011011101110111110101001011001001110
Key : 0011001000110000001100100011000000110010001100000011001000110000
XOR
C3 : 0100000001011001011011010100011101010011010110000100101101000101
P4 : 1111101000001100010011000000110001001100111110101001011001110110
Key : 0011001000110000001100100011000000110010001100000011001000110000
XOR
C4 : 0101110001011001011011010000001000000010000000100000001001101111 1

5) Kemudian kelompokkan biner *cipher* menjadi 8 bit dan koversikan ke dalam *decimal* dan *char*.

Tabel 4. Konversi *Cipherkey :*

konversi kunci					
Char	Decimal	Biner	char	desimal	biner
S	83	01010011	@	64	01000000
\	92	01011100	Y	89	01011001
U	85	01010101	m	109	01101101
_	95	01011111	G	71	01000111
@	64	01000000	S	83	01010011
Y	89	01011001	X	88	01011000
F	70	01000110	K	75	01001011

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

]	93	01011101	E	69	01000101
S	83	01010011	\	92	01011100
O	111	01101111	Y	89	01011001
U	85	01010101	m	109	01101101
_	95	01011111		2	00000010
A	65	01000001		2	00000010
D	68	01000100		2	00000010
M	109	01101101		2	00000010
C	67	01000011	o	111	01101111

c. Pengelompokkan kembali biner-biner *cipherkey* untuk membentuk kunci GOST. Adapun langkah-langkah pengelompokkan *cipherkey* berdasarkan aturan algoritma GOST adalah sebagai berikut :
Kelompokan biner kunci menjadi 8 kelompok. Jumlah bit kunci setiap kelompok 32 bit dimulai dari :
K0 = (k_{32}, …, k_1)
K1 = (k_{64}, …, k_{33})
K2 = (k_{96}, …, k_{65})
K3 = (k128, …, k97)
K4 = (k160, …, k129)
K5 = (k192, …, k161)
K6 = (k224, …, k193)
K7 = (k256, …, k225)
Biner *cipherkey* seluruhnya:
01010011010111000101010101011111010000000101100101000110010111 01
01010011011011110101010101011111010000010100010001101101010000 11
01000000010110010110110101000111010100110101100001001011010001 01
01011100010110010110110100000010000000100000001000000010011011 11

Tabel 4. Pengelompokan *cipherkey :*

Kunci	Posisi Bit	Biner yang diambil
K[0]	32...1	11111010101010100011101011001010
K[1]	64...33	10111010011000101001101000000010
K[2]	96...65	11111010101010101111011011001010
K[3]	128...97	11000010101101100010001010000010
K[4]	160...129	11100010101101101001101000000010
K[5]	192...161	10100010110100100001101011001010
K[6]	224...193	01000000101101101001101000111010
K[7]	256...225	11110110010000000100000001000000

Ubah *cipherkey* menjadi desimal untuk memudahkan proses penjumlahan XOR

Tabel 6. *Cipherkey*

Kunci	Biner yang diambil	Desimal
[0]	11111010101010100011101011001010	4205460170
[1]	10111010011000101001101000000010	3127024130
[2]	11111010101010101111011011001010	4205508298
[3]	11000010101101100010001010000010	3266716290
[4]	11100010101101101001101000000010	3803617794
[5]	10100010110100100001101011001010	2731678410
[6]	01000000101101101001101000111010	1085708858
[7]	11110110010000000100000001000000	4131405888

d. Proses Enkripsi
Proses enkripsi pada perhitungan manual mengambil nilai sampel citra digital 2x2 pada gambar 3.10 dengan nilai yang tersaji dari hasil ekstraksi pada tabel 3.2 di atas. Berikut nilai-nilai *pixel* yang akan dilakukan enkripsi berdasarkan algoritma GOST dari *plainimage* tabel 3.2 yaitu 160, 110, 79, 110, 87, 104, 80, 75. Kunci yang digunakan dalam proses enkripsi adalah *cipherkey (*kunci hasil pengoptimalan menggunakan mode operasi *Output Feedback).*
Adapun proses enkripsi agoritma GOST di uraikan sebagai berikut:
a. Konversikan *plainimage* dan kunci ke biner.
Plainimage = 160, 110, 79, 110, 87, 104, 80, 75,

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

Tabel 7. Konversi *plainimage :*

Dec	160	110	79	110	87	104	80	75
Biner	10100000	011011 10	010011 11	011011 10	010101 11	011010 00	010100 00	010010 11

Gabungkan biner *plainimage :*
10100000011011100100111101101110010101110110100001010000010010 11
Kunci : S\U_@YF]SoU_ADmC@YmGSXKE\Ymo

b. Kelompokan *plainimage* menjadi 2 bagian yaitu R[0] dan L[0] dengan jumlah tiap bit kelompok adalah 32 bit. 32 bit bagian kiri menjadi R[0] dan mulai bit ke 33 sampai bit ke 64 (sebelah kanan) menjadi L[0]. Proses penulisan bitnya dilakukan secara terbalik.
R[0] = bit[32], bit[31].....bit[1]
L[0] = bit[64], bit[63].....bit[33]
Gabungan Biner *Plaintext*:
10100000011011100100111101101110010101110110100001010000010010 11
L(0) = 11010010000010100001011011101010
R(0) =01110110111100100111011000000101
Putaran 0→ i=0

1) L[0]= 11010010000010100001011011101010
R[0]= 01110110111100100111011000000101

2) (R[0]+K[0]) mod 2^{32}
R[0] = 1995601413
K[0] = 4205460170+
= 6201061583mod 2^{32}
= 1906094287
= 01110001100111001011000011001111

Tabel 7. Pengelompokan

Biner Kelompok	Dec Nilai Bit	SBOX	Hasil Permutasi dengan SBOX	Biner
0111	7	→S-Box(0)	14	1110
0001	1	→S-Box(1)	11	1011
1001	9	→S-Box(2)	15	1111
1100	12	→S-Box(3)	11	1011
1011	11	→S-Box(4)	14	1110
0000	0	→S-Box(5)	4	0100
1100	12	→S-Box(6)	6	0110
1111	15	→S-Box(7)	12	1100

3) Gabungkan biner- biner hasil pertukaran dari tabel SBOX dan lakukan *Rotate Left Shift* 11 bit.
Gabungan = 11101011111101111110010001101100
RLS[0] 11 bit = 11011111001000110110011101011111

4) R[1]=RLS[0] XOR L[0]
RLS[0] = 11011111001000110110011101011111
L[0] = 11010010000010100001011011101010
_____XOR
R[1] = 00001101001010010111000110110101
L[1] = R[0] sebelum diproses
L[1] = 01110110111100100111011000000101

5) Hasil Putaran -0 atau pada PUTARAN – i = 0 adalah :
L[1] = 01110110111100100111011000000101
R[1] = 00001101001010010111000110110101

Sampai tahap ini, maka L[1] dan R[1] telah didapatkan, maka tahap selanjutnya adalah proses enkripsi *round* 1-32. Adapun hasil keseluruhanya sebagai berikut:
10010001010101101101111010100101010010000111110111101101110 10111

Kelompokan bit hasil 8 bit per kelompok, kemudian masing – masing kelompok bit konversikan menjadi decimal, sebagai berikut :

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

Tabel 8. Hasil *Chiperimage* :

10010001	01010110	11011110	10100101	01001000	01111101	11101101	11010111
145	86	222	165	72	125	237	215

Sehingga perubahan nilai warna *pixel cipherimage* keseluruhan dalam hitungan manual dapat dilihat pada tabel di bawah ini :

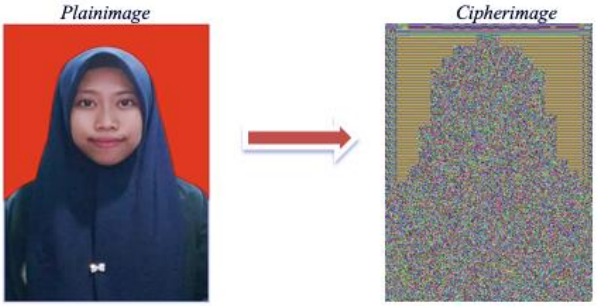
Tabel 9. Nilai *Pixel* Citra Hasil Enkripsi

Pixel	Warna	Plain Desimal	Cipher Desimal
1	R	160	145
	G	110	86
	B	79	222
2	R	110	165
	G	87	72
	B	104	125
3	R	80	237
	G	75	215
	B	150	150
4	R	97	97
	G	80	80
	B	120	120

Berdasarkan tabel 3.9 nilai desimal *plainimage* berbeda dengan nilai desimal *cipherimage*. Perbedaan ini akan membuat citra digital memiliki bentuk yang tidak dapat dimengerti manusia. Sedangkan pada *pixel* 3 nilai B dan *pixel* 4 nilai RGB tidak mengalami perubahan dikarenakan tidak diikutkan dalam proses enkripsi manual. Hasil perbandingan perubahan *plainimage* yang di enkripsi menggunakan algoritma GOST ke empat *pixel* adalah sebagai berikut :

Gambar 4. *pixel Cipherimage* sampel

Hasil perbandingan perubahan *plainimage* yang di enkripsi menggunakan algoritma GOST kesuluruhan adalah sebagai berikut :



Gambar 5. *Cipherimage* Keseluruhan

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan hasil proses enkripsi citra digital menggunakan algoritma GOST dan mode operasi *output feedback* memberikan *output cipherimage* yang memiliki tingkat keamanan yang baik, hal ini dapat dilihat dari perubahan warna-warna *pixel* citra yang tidak dapat dikenali maknanya. Penerapan mode operasi *output feedback* dapat menghasilkan kunci enkripsi dan dekripsi yang optimal pada algoritma GOST, hal ini dapat dilihat dari bentuk karakter kunci GOST yang lebih rumit ketika dilakukan pembangkitan menggunakan *output feedback.*

REFERENCES

[1] M. Winafil , S. Sinurat, Ta and T. Zebua, "IMPLEMENTASI ALGORITMA ADVANCED ENCRYPTION STANDARD DAN TRIPLE DATA ENCRYPTION STANDARD UNTUK MENGAMANKAN CITRA DIGITAL," KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), vol. 2, p. 450, 2018.

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 128-136
https://hostjournals.com/jimat

[2] J. Jenitra, A. U. Wibowo and R. P. Sari, "Implementasi Aplikasi Pengiriman File pada Protokol DTN Berbasis Web," Jurnal Aksara Komputer Terapan, vol. 3, 2014.
[3] S. Wahyuni, B. O. Sinaga, D. Almahera and I. Saputra, "Pengamanan File Docx Menerapkan Algoritma Gronsfeld," Seminar Nasional Teknologi Komputer & Sains (SAINTEKS, pp. 415-419, 2020.
[4] SYLVIANI, "Pengembangan Algoritma Kriptografi GOST (Gosudarstvenny Standart) untuk Peningkatan Keamanan dalam Penyandian Data," Universitas Telkom, Bandung, 2011.
[5] R. Munir and C. Lung, "STUDI DAN IMPLEMENTASI ADVANCED ENCRYPTION STANDART DENGAN EMPAT MODE OPERASI BLOCK CIPHER," Makalah Seminar Tugas Akhir Bandung : Departemen Teknik Informatika ITB, 2005.
[6] R. Sadikin, Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java, T. A. Prabawati, Ed., Yogyakarta, Jawa Tengah: C.V ANDI OFFSET (Penerbit Andi), 2012, p. 218.
[7] R. Munir, Kriptografi, Bandung: INFORMATIKA, 2006, p. 2.
[8] R. Munir, "Tujuan Kriptografi," in Kriptografi, Bandung, Informatika Bandung, 2006, p. 9.
[9] M. Amelia, 28 April 2014. [Online]. Available: https://meiilinuxer.wordpress.com/2014/04/28/operator-shift-right-dan-shift-left/.