p-ISSN 2797-9709
e-ISSN 2797-989X

Volume 2 No. 1 Januari 2022

KEAMANAN PESAN WHATSAPP MENGGUNAKAN KRIPTOGRAFI ALGORITMA GOVERNMENT STANDARD (GOST)

Lisda Juliana Pangaribuan[1*], Delima Sitanggang[2]
[1] AMIK Medan Business Polytechnic Medan, [2] Universitas Prima Indonesia
Email: [1]lisdajuliana@gmail.com, [2] delimasitanggang@unpri.ac.id
Email Penulis Korespondensi: lisdajuliana@gmail.com

ABSTRACT

Semakin banyaknya aplikasi media sosial membuat masyarakat lebih banyak melakukan komunikasi via chat. Hal ini membuat cryptanalis berlomba melakukan penyadapan, sebab itu sangat dibutuhkan keamanan yang baik dalam sebuah aplikasi media sosial. Salah satu metode pengamanan data adalah kriptografi Selain kerahasiaan kunci, algoritma untuk enkripsi dan dekripsi pesan juga sangat memegang peranan penting dalam keamanan pesan yang dikirim. Algoritma yang digunakan adalah algoritma Government Standard (GOST).Algoritma GOST merupakan algoritma simetris blok cipher 64 bit yang dikenal cukup aman karena memiliki 32 putaran dalam proses enkripsi dan dekripsi. Dalam penelitian ini pesan yang dikirim hanya dalam bentuk teks. Tujuan penelitian ini adalah meningkatkan keamanan pesan yang dikirim melalui Whatsapp dengan melakukan enkripsi dan dekripsi menggunakan algoritma GOST Hasil dari penelitian ini menunjukkan bahwa enkripsi pesan berupa teks dapat dikembalikan ke teks aslinya dengan algoritma GOST. Keamanan menggunakan Aloritma GOST sangat baik karena proses enkripsi dan dekripsi cukup panjang dan rumit karena melalui 32 putaran yang memproses 64 bit pesan.

Keyword : *kriptografi, algoritma GOST, S_BOX, Enkripsi, Dekripsi*

PENDAHULUAN

Penyadapan dan kloning yang sering terjadi pada saat mengirim atau meneruskan informasi melalui whatsapp membuat masyarakat sering menerima informasi hoax. Untuk itu dibutuhkan metode pengamanan pengiriman informasi yang baik supaya informasi yang dikirim hanya diterima oleh penerima yang pantas saja. Selain steganografi, metode yang sering digunakan untuk keamanan pesan adalah kriptografi [1]. Kriptografi merupakan seni mengenai metode pengiriman pesan secara rahasia supaya hanya penerima aslinya yang dapat menghapus dan membaca pesan dan memahaminya [2]. Berbagai algoritma kriptografi untuk keamanan informasi dikembangkan untuk melindungi informasi supaya tidak disadap, dirusak atau dicuri oleh pihak yang tidak berhak[3].

Selain kerahasiaan kunci, algoritma untuk enkripsi dan dekripsi pesan juga sangat penting dalam keamanan pesan yang dikirim. Walaupun semua pesan yang dikirim melalui whatsapp udah di enkripsi menggunakan metode END TO END, namun pesan masih juga dapat disadap sehingga dalam penelitian ini digunakan kriptografi dengan algoritma GOST untuk keamanan pesan melalui Whatsapp.

Algoritma GOST merupakan algoritma GOST dengan algoritma enkripsi sederhana yang memiliki jumlah proses sebanyak 32 round dan menggunakan 64 bit block cipher dengan 256 bit key. Metoda GOST juga menggunakan 8 buah S-Box yang permanen dan operasi XOR serta Rotate Left. Walaupun sederhana, namun kriptografi dengan algoritma GOST memiliki kerumitan dalam proses enkripsi dan dekripsi karena mengkombinasi beberapa perhitungan matematis.

Fungsi hash pada kriptografi algoritma GOST merupakan sebuah prosedur yang mengambil blok-blok data sebuah pesan dan mengembalikannya kembali ke pesan semula dengan string bit yang panjangnya telah ditentukan. Fungsi GOST hash menjadi dasar pembuatan GOST cipher block 64 bit dengan panjang kunci 256 bit atau 32 karakter yang dikembangkan oleh Federal Agency for Government Communication and Information dan oleh UniSofiet Scientific and Research Institute of Standardization menjadi fungsi hash

p-ISSN 2797-9709
e-ISSN 2797-989X

Volume 2 No. 1 Januari 2022

standar di Rusia. Pesan Sting bit yang memiliki blok-blok sebesar 256 bit diproses oleh fungsi GOST hash menjadi nilai hash 256 bit. Jika panjang pesan tidak mencapai kelipatan 256, maka pesan string bit akan di-padding seminimal mungkin panjang pesan menjadi klipatan 256 bit atau 32 karakter.[4]

ALGORITMA GOST

GOST adalah singkatan dari Government Standard merupakan suatu algoritma Blok chipper. Kriptografi Algoritma GOST merupakan kriptografi Modern, yang berorientasi bit karena penyandian modern menggunakan media komputer untuk mengolah pesan. Pesan pada sandi modern tidak selamanya berupa untaian karakter bisa juga berupa rangkaian bit seperti video atau berkas gambar.[5]

Komponen dari metoda GOST adalah :[6]

1. Key Store Unit (KSU) menyimpan 256 bit string dengan menggunakan 32 bit register (K0, K1, …, K7).
2. Dua buah 32 bit register (R1, R2).
3. 32 bit adder modulo 232 (CM1).
4. Bitwise Adder XOR (CM2).
5. Substitusion block (S) yaitu berupa 8 buah 64 bit S-Box.
6. Rotasi Left shift register (R) sebanyak 11 bit.

Algoritma Pembangkitan Kunci

Kunci internal pada algoritma GOST dibangkitkan dari kunci eksternal yang diberikan *user*. Pembangkitan kunci internal dilakukan dengan membagi kunci eksternal 256 bit (k1, k2, k3, k4, …, k256) menjadi 8 bagian dengan panjang masing-masing 32 bit dimana pembaginya K0, K1, …, K7. [7]

Algoritma Enkripsi

Proses Enkripsi pada algoritma GOST untuk satu putaran (iterasi), adalah: [5]

1) 64 bit plainteks dibagi menjadi 2 bagian 32 bit, yaitu Li dan Ri. Caranya : Input a1(0), a2(0), ….. ,a32(0) ; b1(0), b2(0), ….. ,b32(0)
   R0 = a32(0), a31(0), ….. ,a1(0)
   L0 = b32(0), b31(0), ….. ,b1(0)
2) (Ri + Ki) mod 232.
   Hasil penjumlahan modulo 232 berupa 32 bit.
3) Hasil dari penjumlahan modulo 232 dibagi menjadi 8 bagian, yang masing-masing bagian terdiri dari 4 bit. Setiap bagian dimasukkan ke dalam table S-box yang berbeda, 4 bit pertama menjadi input dari SBox 0, 4 bit kedua menjadi SBox 1 dan seterusnya.
   Hasil yang didapat dari substitusi ke S-Box kemudian digabungkan kembali menjadi 32 bit dan kemudian dilakukan RLS (Rotate Left Shift) pergeseran ke kiri sebanyak 11 bit.
5) Ri + 1 = RLS XOR Li
   6) Li + 1 = Ri sebelum dilakukan proses Langkah nomor 2 sampai 6 dilakukan sebanyak 32 kali (putaran). Pada langkah nomor 2 penggunaan kunci dijadwalkan penggunaannya sesuai dengan putarannya.
   Untuk putaran ke-31, langkah nomor 5 dan 6 sedikit berbeda. Langkah 5 dan 6 untuk putaran 31 adalah :
   R32 = R31 sebelum dilakukan proses
   L32 = L31 XOR R31
   Sehingga cipherteks yang dihasilkan adalah,
   L32 : b(32), b(31), ….. ,b(1)
   R32 : a(32), a(31), ….. ,a(1)

Algoritma Dekripsi

Proses dekripsi algoritma GOST hampir sama dengan proses enkripsi, bedanya penggunaan bit kunci pada setiap putarannya. Proses dekripsi akan melalui 32 putaran yang memproses 64 bit pesan (16 bilangan heksa karena 1 heksa = 4 bit) dan mengubahnya menjadi 64 bit cipher.[6]

S-Box yang digunakan pada metoda GOST dapat dilihat pada tabel1,

Tabel1. S-Box dari Metoda GOST

Tabel S-Box	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S-Box 0	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S-Box 1	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S-Box 2	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S-Box 3	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S-Box 4	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S-Box 5	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S-Box 6	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S-Box 7	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Tabel 2 Penjelasan Cara Kerja S-Box dari Metoda GOST

p-ISSN 2797-9709
e-ISSN 2797-989X

Volume 2 No. 1 Januari 2022

Posisi	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S-Box 1	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3

Misalkan data input ke S-Box adalah 5 maka dicari data pada posisi ke-5. Output yang dihasilkan adalah 8.

METODE PENELITIAN

Jenis penelitian ini adalah penelitian deskriptif menggunakan metode Pengolahan data untuk mengetahui hasil enkripsi dan dekripsi pesan pada whatsapp.Jenis data yang digunakan hanya pesan dalam bentuk teks atau karakter alfanumerik.

Tahapan Penelitian

Algoritma yang digunakan untuk enkripsi dan dekripsi pesan dalam penelitian ini adalah algoritma GOST. Untuk meningkatkan keamanan pesan dilakukan modifikasi terhadap kunci.

Langkah-langkah penyelesaian masalah
Penyelesaian masalah dalam penelitian ini menggunakan Metode GOST mengikuti langkah-langkah :

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

Proses Pembentukan Kunci

Langkah-langkahnya :

A. Input key berupa 256 bit key dengan perincian k1, k2, k3, k4, …, k256.
B. Kelompokkan dan masukkan key tersebut ke dalam 8 buah KSU :
   K0 = (k32, …, k1)
   K1 = (k64, …, k33)
   K2 = (k96, …, k65)
   K3 = (k128, …, k97)
   K4 = (k160, …, k129)
   K5 = (k192, …, k161)
   K6 = (k224, …, k193)
   K7 = (k256, …, k225)

Proses Enkripsi

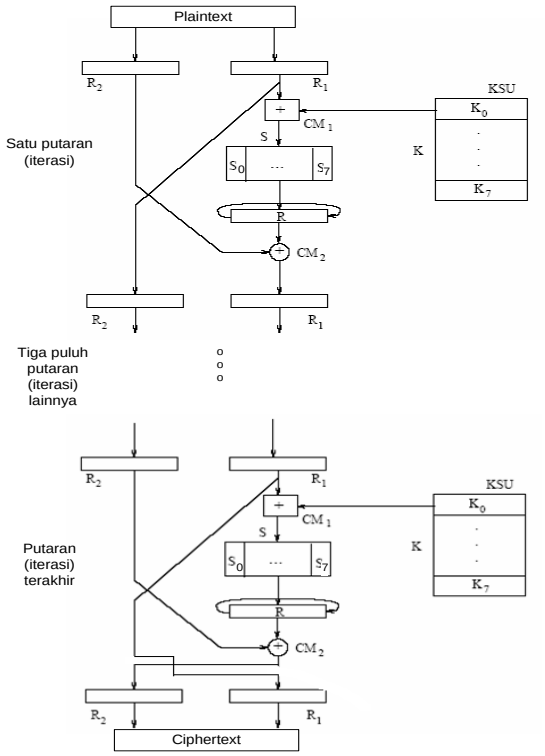
Proses enkripsi dari metoda GOST untuk satu putaran (iterasi) adalah :

1. 64 bit plaintext dibagi menjadi 2 buah bagian 32 bit, yaitu Li dan Ri.
   Caranya : Input a1(0), a2(0), …, a32(0), b1(0), …, b32(0)
   R0 = a32(0), a31(0), …, a1(0)
   L0 = b32(0), b31(0), …, b1(0)
2. (Ri + Ki) mod 232. Hasil dari penjumlahan modulo 232 berupa 32 bit.
   Hasil dari penjumlahan modulo 232 dibagi menjadi 8 bagian, dimanan masing-masing bagian terdiri dari 4 bit. Setiap bagian dimasukkan ke dalam tabel S-Box yang berbeda, 4 bit pertama menjadi input dari S-Box pertama, 4 bit kedua menjadi S-Box kedua, dan seterusnya.
3. Hasil yang didapat dari substitusi ke S-Box dan digabungkan kembali menjadi 32 bit dan kemudian dilakukan rotasi left shift sebanyak 11 bit.
4. Ri+1 = Ri (hasil dari rotate left shift) XOR Li.
5. Li+1 = Ri sebelum dilakukan proses.
6. ..Proses penjumlahan modulo 232, S-Box, Rotate Left Shift dilakukan sebanyak 32 kali (putaran) dengan penggunaan kunci pada masing-masing putaran berbeda-beda sesuai dengan aturan :
   Putaran 0 – 7 : K0, K1, K2, …, K7
   Putaran 8 – 15: K0, K1, K2, …, K7
   Putaran 16 – 23 : K0, K1, K2, …, K7
   Putaran 24 – 31 : K7, K6, K5, …, K0
   Untuk putaran ke-31, langkah 5 dan 6 agak sedikit berbeda. Langkah 5 dan 6 untuk putaran 31 adalah : R32 = R31 sebelum dilakukan proses
   L32 = L31 XOR R31
   Sehingga, ciphertext yang dihasilkan adalah
   L32 : b(32), b(31), …, b(1)
   R32 : a(32), a(31), …, a(1)
   T = a(1), …, a(32), b(1), …, b(32)

Proses enkripsi dari metoda GOST dapat digambarkan dalam bentuk skema seperti gambar 1.

p-ISSN 2797-9709
e-ISSN 2797-989X

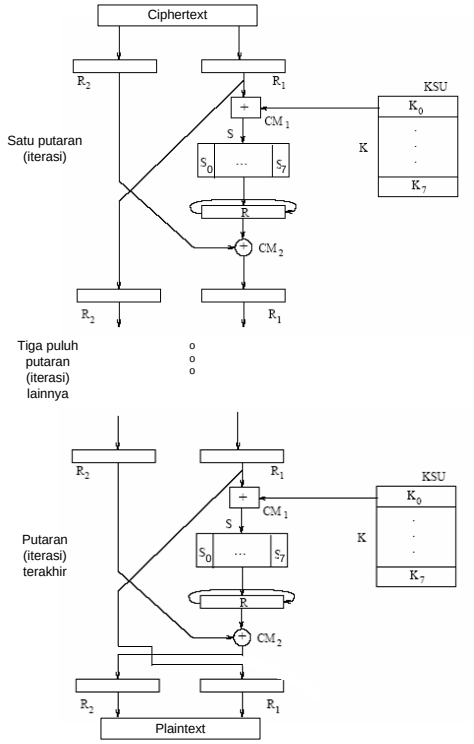
Volume 2 No. 1 Januari 2022



Gambar1. Skema Proses Enkripsi Algoritma GOST

Proses Dekripsi

Proses dekripsi dari metoda GOST dapat digambarkan dalam bentuk skema seperti Gambar 2



Gambar 2. Skema Proses Dekripsi Algoritma GOST

HASIL DAN PEMBAHASAN

Kriptografi yang digunakan untuk keamanan pesan whatsapp chating pada penelitian ini menggunakan algoritma GOST.Adapunproses yang dilakukan adalah 3 proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

A. Proses Pembentukan Kunci

Proses pembentukan kunci ini memerlukan input data key dengan panjang 256 bit atau 64 digit heksadesimal atau 32 buah karakter.

1. Input kunci (32 karakter).

KUNCI = ‘KriptoAlgoritmaGOST LISDAJULIANA'

2. Ubah kunci ke bentuk biner.

'K' = 75 = 01001011
'r' = 114 = 01110010
'i' = 105 = 01101001
'p' = 112 = 01110000
't' = 116 = 01110100
'o' = 111 = 01101111
'A' = 65 = 01000001
'l' = 108 = 01101100
'g' = 103 = 01100111
'o' = 111 = 01101111
'r' = 114 = 01110010
'i' = 105 = 01101001
't' = 116 = 01110100
'm' = 109 = 01101101
'a' = 97 = 01100001
'G' = 71 = 01000111
'O' = 79 = 01001111
'S' = 83 = 01010011
'T' = 84 = 01010100
' ' = 32 = 00100000
'L' = 76 = 01001100
'I' = 73 = 01001001
'S' = 83 = 01010011
'D' = 68 = 01000100
'A' = 65 = 01000001
'J' = 74 = 01001010
'U' = 85 = 01010101
'L' = 76 = 01001100
'I' = 73 = 01001001
'A' = 65 = 01000001
'N' = 78 = 01001110
'A' = 65 = 01000001

Hasil konversi kunci ke bentuk biner =
01001011011100100110100101110000011101000111001001101001011100000111010011011110100000010110110001100111011011101110010011010010111010000110110101100001010000111010011110101001101010100001000001001100010010010101001101000100010

000010100101001010101010011000100100101
00000101001110010000001

3. Kelompokkan hasil yang diperoleh pada K(0)-K(7)

K(0) = k(32), ... , k(1) =
00001110100101100100111011010010
K(1) = k(64), ... , k(33) =
00110110100000101111011000101110
K(2) = k(96), ... , k(65) =
10010110010011101111011011100110
K(3) = k(128), ... , k(97) =
11100010100001101011011000101110
K(4) = k(160), ... , k(129) =
00000100001010101100101011110010
K(5) = k(192), ... , k(161) =
00100010110010101001001000110010
K(6) = k(224), ... , k(193) =
00110010101010100101001010000010
K(7) = k(256), ... , k(225) =
10000010011100101000001010010010

4. Kunci K(0) - K(7) akan digunakan dalam proses enkripsi dan dekripsi.

PROSES ENKRIPSI -> Putaran 0 - 7 : K(0), K(1), K(2), ... , K(7)
Putaran 8 - 15 : K(0), K(1), K(2), ... , K(7)
Putaran 16 - 23 : K(0), K(1), K(2), ... , K(7)
Putaran 24 - 31 : K(7), K(6), K(5), ... , K(0)

PROSES DEKRIPSI -> Putaran 0 - 7 : K(0), K(1), K(2), ... , K(7)
Putaran 8 - 15 : K(7), K(6), K(5), ... , K(0)
Putaran 16 - 23 : K(7), K(6), K(5), ... , K(0)
Putaran 24 - 31 : K(7), K(6), K(5), ... , K(0)

B. Proses Enkripsi

Proses enkripsi dilakukan dengan input data plaintext 64 bit atau 16 digit heksadesimal atau 8 karakter dengan melalui 32 tahapan iterasi (putaran).

PROSES ENKRIPSI - PUTARAN 0
(1) PLAIN TEXT (8 karakter)= 'Pesan ku'
Konversi ke biner =
01010000011001010111001101100001011011
00010000001101011011010 1
L(0) = 10101110110101100000010001110110
R(0)= 10000110110011101010011000001010
(2) R(0) + K(0) mod 2^32 = 2506421468
= 10010101011001001111010011011100
(3) Pecah menjadi 8 kelompok dan masukkan ke SBox.
1001 = 9 -> SBOX(0) -> 11 = 1011
0101 = 5 -> SBOX(1) -> 13 = 1101
0110 = 6 -> SBOX(2) -> 4 = 0100
0100 = 4 -> SBOX(3) -> 0 = 0000
1111 = 15 -> SBOX(4) -> 2 = 0010
0100 = 4 -> SBOX(5) -> 7 = 0111
1101 = 13 -> SBOX(6) -> 8 = 1000
1100 = 12 -> SBOX(7) -> 6 = 0110
(4) Hasil digabungkan kembali dan lakukan Rotate Left Shift sebanyak 11 kali.
RLS(1)=01111010100000000100111100001101
RLS(2)=11110101000000001001111000011010
RLS(3)=11101010000000010011110000110101
RLS(4)=11010100000000100111100001101011
RLS(5)=10101000000001001111000011010111
RLS(6)=01010000000010011110000110101111
RLS(7)=10100000000100111100001101011110
RLS(8)=01000000001001111000011010111101
RLS(9)=10000000010011110000110101111010
RLS(10)=00000000100111100001101011110101
RLS(11)=00000001001111000011010111101010
(5) R(1) = R(0) XOR L(0)
R(1) = 10101111111010100011000110011100
(6) L(1) = R(0) sebelum proses.
L(1) = 10000110110011101010011000001010

PROSES ENKRIPSI - PUTARAN 1 diperoleh nilai
L(2)=10101111111010100011000110011100
PROSES ENKRIPSI - PUTARAN 2 diperoleh nilai L(3) = 10010101001001011011110011110000

PROSES ENKRIPSI - PUTARAN 3 diperoleh
L(4) = 10101000111011100110010100010101
PROSES ENKRIPSI - PUTARAN 4 diperoleh
L(5) = 11010011000011000011111111111001
PROSES ENKRIPSI - PUTARAN 5 diperoleh
L(6) = 01010011101011110001001011000011
PROSES ENKRIPSI - PUTARAN 6 diperoleh
L(7) = 11110111101100100000000010000000
PROSES ENKRIPSI - PUTARAN 7 diperoleh
L(8) = 10001100001010011111111000000010
PROSES ENKRIPSI - PUTARAN 8 diperoleh
L(9) = 00101101100101110010101010101111
PROSES ENKRIPSI - PUTARAN 9 diperoleh

L(10) = 00011000000110011000011100000010
PROSES ENKRIPSI - PUTARAN 10
L(11) = 11110001111101111010010000000000
PROSES ENKRIPSI - PUTARAN 11
L(12) = 01010101010110001101010000010111
PROSES ENKRIPSI - PUTARAN 12
L(13) = 11101011101101100001110101010000
PROSES ENKRIPSI - PUTARAN 13
L(15) = 10001010011010001100011101011111
PROSES ENKRIPSI - PUTARAN 15
L(16) = 11100110111100000100000011100111
PROSES ENKRIPSI - PUTARAN 16
L(17) = 11011010011011000101011101000101
PROSES ENKRIPSI - PUTARAN 17
L(18) = 00110010011111011110001001111101
PROSES ENKRIPSI - PUTARAN 18
L(20) = 01100100000001000101101110111000
PROSES ENKRIPSI - PUTARAN 20
L(21) = 00000101101010100111000110100000
PROSES ENKRIPSI - PUTARAN 21
L(22) = 00100100110010010011000110100000
PROSES ENKRIPSI - PUTARAN 22
L(23) = 10001101101011100001111101110111
PROSES ENKRIPSI - PUTARAN 23
L(24) = 11010000100101110010001001010001
PROSES ENKRIPSI - PUTARAN 24
L(25) = 00101001000101110001101100010101
PROSES ENKRIPSI - PUTARAN 25
L(26) = 11110101111100000000001011011110
PROSES ENKRIPSI - PUTARAN 26
L(27) = 10011110100001011001011110000010
PROSES ENKRIPSI - PUTARAN 27
L(28) = 01001110010000111010100100000111
PROSES ENKRIPSI - PUTARAN 28
L(29) = 10110010100000111000001110100000
PROSES ENKRIPSI - PUTARAN 29
L(30) = 11111110111100001110110011101101
PROSES ENKRIPSI - PUTARAN 30
L(31) = 00101111000001111000011001001110
PROSES ENKRIPSI - PUTARAN 31
Hasil dalam biner = 01000001001110100011101010100111110101110010100000101011101
1111 -> ubah ke ascii

Maka CIPHER TEXT = **A:=Së”ß**

C. Proses Dekripsi

Proses dekripsi dari algoritma GOST sama dengan proses enkripsi sebagai berikut :

PROSES DEKRIPSI - PUTARAN 0
(1) CIPHER TEXT = 'A:=Së”ß'
Konversi ke biner = 01000001001110100011101010100111110101110010100000101011101 1111
L(0) = 11111011101010000010100111010111
R(0) = 11001010101111000101110010000010
(2) R(0) + K(0) mod 2^32 = 3646073684
= 11011001010100101010101101010100
(3) Pecah menjadi 8 kelompok dan masukkan ke SBox.
1101 = 13 -> SBOX(0) -> 15 = 1111
1001 = 9 -> SBOX(1) -> 3 = 0011
0101 = 5 -> SBOX(2) -> 3 = 0011
0010 = 2 -> SBOX(3) -> 10 = 1010
1010 = 10 -> SBOX(4) -> 9 = 1001
1011 = 11 -> SBOX(5) -> 5 = 0101
0101 = 5 -> SBOX(6) -> 15 = 1111
0100 = 4 -> SBOX(7) -> 5 = 0101
(4) Hasil digabungkan kembali dan lakukan Rotate Left Shift sebanyak 11 kali.
LS(1)=11100110011101010010101111101011
LS(2)=11001100111010100101011111010111
LS(3)=10011001110101001010111110101111
LS(4)=00110011101010010101111101011111
LS(5)=01100111010100101011111010111110
LS(6)=11001110101001010111110101111100
LS(7)=10011101010010101111101011111001
LS(8)=00111010100101011111010111110011
LS(9)=01110101001010111110101111100110
RLS(10)=11101010010101111101011111001100
RLS(11)=11010100101011111010111110011001
(5) R(1) = R(0) XOR L(0)

p-ISSN 2797-9709
e-ISSN 2797-989X

Volume 2 No. 1 Januari 2022

R(1) = 00101111000001111000011001001110
(6) L(1) = R(0) sebelum proses.
L(1) = 11001010101111000101110010000010

PROSES DEKRIPSI - PUTARAN 1
L(2) = 00101111000001111000011001001110
PROSES DEKRIPSI - PUTARAN 2
L(3) = 11111110111100001110110011101101
PROSES DEKRIPSI - PUTARAN 3
L(4) = 10110010100000111000001110100000
PROSES DEKRIPSI - PUTARAN 4
L(5) = 01001110010000111010100100000111
PROSES DEKRIPSI - PUTARAN 5
L(6) = 10011110100001011001011110000010
PROSES DEKRIPSI - PUTARAN 6
L(14) = 01100011111100110010011101011111
PROSES DEKRIPSI - PUTARAN 14
L(15) = 00110010011111101111000100111110 1
PROSES DEKRIPSI - PUTARAN 15
L(16) = 11011010011011000101011101000101
PROSES DEKRIPSI - PUTARAN 16
L(17) = 11100110111100000100000011100111
PROSES DEKRIPSI - PUTARAN 17
L(18) = 10001010011010001100011101011111
PROSES DEKRIPSI - PUTARAN 18
L(19) = 11101000110000010011111011011011
PROSES DEKRIPSI - PUTARAN 19
L(20) = 11101011101101100001110101010000
PROSES DEKRIPSI - PUTARAN 20
L(21) = 01010101010110001101010000010111
PROSES DEKRIPSI - PUTARAN 21
L(22) = 11110001111101111010010000000000
PROSES DEKRIPSI - PUTARAN 22
L(23) = 00011000000110011000011100000010
PROSES DEKRIPSI - PUTARAN 23
L(24) = 00101101100101110010101010101111
PROSES DEKRIPSI - PUTARAN 24
L(25) = 10001100001010011111111000000010
PROSES DEKRIPSI - PUTARAN 25
L(26) = 11110111101100100000000010000000
PROSES DEKRIPSI - PUTARAN 26
L(27) = 01010011101011110001001011000011
PROSES DEKRIPSI - PUTARAN 27
L(28) = 11010011000011000011111111111001
PROSES DEKRIPSI - PUTARAN 28
L(29) = 10101000111011100110010100010101
PROSES DEKRIPSI - PUTARAN 29
L(30) = 10010101001001011011110011110000
L(7) = 11110101111100000000001011011110
PROSES DEKRIPSI - PUTARAN 7
L(8) = 00101001000101110001101100010101
PROSES DEKRIPSI - PUTARAN 7
L(8) = 00101001000101110001101100010101
PROSES DEKRIPSI - PUTARAN 8
L(9) = 11010000100101110010001001010001
PROSES DEKRIPSI - PUTARAN 9
L(10) = 10001101101011100001111101110111
PROSES DEKRIPSI - PUTARAN 10
L(11) = 00100100110010010011000110100000
PROSES DEKRIPSI - PUTARAN 11
L(12) = 00000101101010100111000110100000
PROSES DEKRIPSI - PUTARAN 12
L(13) = 01100100000001000101101110111000

PROSES DEKRIPSI - PUTARAN 13
PROSES DEKRIPSI - PUTARAN 30
L(31) = 10101111111010100011000110011100
PROSES DEKRIPSI - PUTARAN 31
L(32) = b(32), b(31), ... b(1)
R(32) = a(32), a(31), ... a(1)
Hasil = a(1), ... a(32), b(1), ... b(32)
Hasil dalam biner = 01010000011001010111001101100001011011100010000011010110111
0101 -> ubah ke ascii
maka PLAIN TEXT = **Pesan ku**

KESIMPULAN

Berdasarkan hasil dan pembahasan dalam penelitian ini maka dapat disimpulkan :

1. Proses pembentukan kunci pada metoda GOST cukup sederhana sedangkan proses enkripsi dan dekripsi cukup panjang dan rumit karena melalui 32 putaran yang memproses 64 bit pesan.
2. Hasil enkripsi pesan berupa teks dapat dikembalikan ke teks aslinya dengan algoritma GOST.
3. Panjang kunci yang di input adalah 32 karakter, jika kurang dari 32 maka sisanya akan dianggap sebagai Dummy atau spasi.

DAFTAR PUSTAKA

[1] L. J. Pangaribuan, “Kriptografi Hybrida Agloritma Hill Cipher Dan Rivest Shamir Adleman (RSA) Sebagai Pengembangan Kriptografi Kunci Simetris (Studi Kasus : Nilai Mahasiswa Amik Mbp),” *J. Teknol.*

Inf. DAN Komun., vol. 7, no. 1, pp. 11–26, 2018.

[2] L. J. Pangaribuan and F. H. Simbolon, "Kriptografi Hybrida Menggunakan Algoritma Hill Cipher Dan," in *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 2017, vol. I, pp. 6–16.

[3] C. Helena, P. Panjaitan, and L. J. Pangaribuan, "Analisis Pola Identifikasi Zero Knowledge Proof Dengan Algoritma Feige Fiat Shamir Menggunakan Blum Blum Shub," *MEANS (Media Inf. Anal. dan Sist.*, vol. 6, no. 1, pp. 7–12, 2021.

[4] A. S. Kodar Udoyono, "Jurnal Teknologi Informasi dan Komunikasi STMIK Subang, April 2018 ISSN: 2252-4517," *J. Teknol. Inf. dan Komun.*, no. April, pp. 49–66, 2018.

[5] T. Arianti and B. Nadeak, "Perancangan Aplikasi Pembelajaran Kriptografi Algoritma GOST dengan Menggunakan Metode Computer Based Instruction," vol. 01, no. 1, pp. 40–46, 2019.

[6] Hendri, "PENGAMANAN APLIKASI CHATTING MENGGUNAKAN METODE," *Maj. Ilm. INTI*, vol. 5, no. 1, pp. 14–19, 2017.

[7] Gustaria S ; Titin Fatimah, "IMPLEMENTASI KRIPTOGRAFI MENGGUNAKAN ALGORITMA GOST (GOSUDARSTEVVENYI STANDARD) UNTUK PENGIRIMAN E-MAIL PADA APLIKASI CIRUS-MAIL BERBASIS WEB," *Sist. Komput. dan Tek. Inform.*, no. 2721–4788.