JURNAL ARMADA INFORMATIKA
p-ISSN: 2598-0416 https://jurnal.stmikmethodistbinjai.ac.id/ e-ISSN: 2615-689X

Algoritma Blowfish Untuk Pengaman Pesan Teks

Tomy Satria Alasi*[1], Jakaria Sembiring[2]
[1]STMIK Logika; Jl. Kol. Yos Sudarso No.374, Pulo Brayan, Medan
[2]Politeknik Unggul LP3M; Jalan Iskandar Muda No 3EF Medan
e-mail: *[1]tomysatriaalasi@live.com, [2]jakariasembiring@gmail.com

Abstrak

Penelitian untuk menjabarkan bagaimana proses pengamanan pesan. Karena berkomunikasi sama satu lain adalah sifat dasar manusia sejak di lahirkan, seiring perkembangan zaman media informasi dan menyimpan tulisan berbagai informasi dapat disimpan dalam bentuk pesan text. Dan informasi atau data yang asli bisa saja tidak sampai atau bias sampai tapi sudah diubah atau dirombak pada pihak-pihak yang tidak berhak atau merupakan penyadapan yang dilakukan oleh penyadap atau hacker.

Banyak metode yang digunakan untuk penyandian pesan yang disebut dengan Kriptografi, gunanya untuk merahasiakan dan menyandikan pesan yang dikirim dan pesan yang diterima. Salah satu diantaranya adalah algoritma kunci Simetris atau asimetris, salah satu metode enkripsi pesan adalah Blowfish. Blowfish adalah suatu algoritma penyandian pesan.

Kata kunci*—Pengamanan, Pesan Teks, Algoritma Blowfish*

Abstract

The research to describe how the process of securing messages. Because communicating with each other is a basic human nature since birth, along with the development of the times, information media and storing writings various information can be stored in the form of text messages. And the original information or data may not arrive or be biased until but has been changed or overhauled to unauthorized parties or is a wiretap carried out by eavesdroppers or hackers.

Many methods are used for encoding messages called Cryptography, which is used to keep secret and encode messages sent and messages received. One of them is a symmetric or asymmetric key algorithm, one of the message encryption methods is Blowfish. Blowfish is a message encoding algorithm.

Keywords— *Security, Text Messages, Blowfish Algorithms*

1. PENDAHULUAN

Hal yang sering kali ditakutkan oleh pihak yang saling ingin bertukar informasi satu sama lain adalah keamanan dan kerahasian sebuah informasi dalam komunikasi[1] menjadi hal yang sangat penting, karena data atau informasi yang sangat penting terkadang tidak sampai ketangan si penerima atau bisa saja sampai ke tangan si penerima[2], tapi informasi atau data tersebut sudah tidak asli atau sudah dirubah tanpa ada pengetahuan dari si pengirim dan penerima[3].
informasi atau data yang asli sudah disadap dan dirubah isi dari data tersebut, seharusnya data

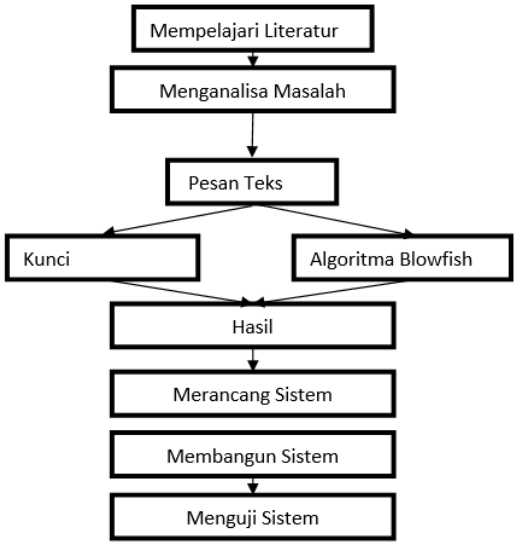
yang dikirim dan diterima berupa data yang asli, tapi sebaliknya data yang dikirim dan diterima berupa data yang sudah dirubah oleh si penyadap atau hacker sehingga data atau informasi tersebut sudah tidak asli lagi.

Maka suatu data atau informasi perlu ada keamanan dan kerahasian agar tidak ada seorang pun bisa menyadap pesan atau data yang dikirim dan bisa diterima oleh penerima pesan atau informasi yang asli tanpa ada perubahan. Selain itu kebanyakan dari pihak yang saling bertukar informasi menggunakan beberapa macam metode untuk menjaga keamanan dan kerahasian data atau informasi yang dikirimkan[4].

Antara lain yang menggunakan sebuah metode penyadian pesan yang disebut dengan Kriptografi *(Cryptography)*[5], digunakan untuk merahasiakan dan menyandikan pesan yang dikirim dan pesan yang diterima. salah satu diantaranya *algoritma* kunci *simetris* ataupun *asimetris* (pembagian berdasarkan kunci). Salah satu metode *enkripsi* pesan adalah *Blowfish*. *Blowfish* adalah suatu algorima penyandian pesan yang diciptakan oleh seorang *Cryptanalyst* bernama Bruce Schneier Presiden perusahaan *Counterpane Internet Security*[6].

2. METHODS

Berikut ini adalah kerangka metode penyelesaian masalah. Untuk pengamanan pesan teks menggunakan algoritma blowfish.



Gambar 1 Metode Penelitian

3. RESULTS AND DISCUSSION

Untuk melakukan enkripsi, proses diawali dengan Input-bit teks terang sebanyak 64-bit. Kemudian 64-bit dibagi menjadi dua bagian yaitu disisi kiri (xL) sepanjang 32 bit dan disisi kanan (xR) sepanjang 32 bit. Setiap bagian teks terang dioperasikan sendiri-sendiri dengan melakukan enam belas (16) iterasi. Kemudian di-XORkan xL(kiri) dengan Pi, kemudian hasil dari Xor antara xL dan Pi akan menjadi X(1). Sisi sebelah kiri akan mengalami proses yang sama dengan sisi sebelah kanan, Kemudian di-XORkan F(xL) dengan xR kemudian hasil dari Xor antara F(xL) dan xR akan menjadi P18, kemudian xL,xR di swap (tukar) agar mendapat hasil yang terakhir.

Langkah terakhir Swap (xL,xR), kemudian xR di Xor dengan P17, dan xL juga di Xor dengan P18, lalu hasil dari peng-Xor xL dan xR digabung kembali untuk mendapatkan *Cipherteks.*

Tabel 1 S-Box

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

P1= NE= 0100111001000101
P2= NI = 0100111001001001
P3= SU= 0101001101010101
P4= RI = 0101001001001001
P5= YA= 0101100101000001
P6=NI = 0100111001001001
P7=DC = 0100010001000011
P8=CD = 0100001101000100
P9=EF = 0100010101000110
P10=FE= 0100011001000101
P11=GH=0100011101001000
P12=HG=0100100001000111
P13=TG= 0101010001000111
P14=ED= 0100010101000100
P15=MI= 0100110101001001
P16=OP= 0100111101010000
Kunci
K1=ABCD=
01000001010000100100001101000100
Langkah II
P1 Xor 32 bit
00000000000000000100111001000101
01000001010000100100001101000100

Xor
01000001010000100000110100000001
K2=
01000001010000100000110100000001
P2 Xor K2
00000000000000000100111001001001
01000001010000100000110100000001

-- **Xor**
01000001010000100100001101001000
K3=
01000001010000100100001101001000
P3 Xor K3
00000000000000000101001101010101
01000001010000100100001101001000

Xor
01000001010000100001000000011101
K4=
01000001010000100001000000011101
P4 Xor K4
00000000000000000101001001001001
01000001010000100001000000011101

Xor
01000001010000100100001001010100
P5= YA= 0101100101000001
P6=NI = 0100111001001001
P7=DC = 0100010001000011
P8=CD = 0100001101000100
P9=EF = 0100010101000110
K5=
01000001010000100100001001010100
P5 Xor K5
00000000000000000101100101000001
01000001010000100100001001010100

Xor
01000001010000100001101100010101
K6=
01000001010000100001101100010101
P6 Xor K6
00000000000000000100111001001001
01000001010000100001101100010101

Xor
01000001010000100101010101011100
K7=
01000001010000100101010101011100
P7 Xor K7
00000000000000000100010001000011
01000001010000100101010101011100

Xor
01000001010000100001000100011111
K8=
01000001010000100001000100011111
P8 Xor K8
00000000000000000100001101000100
01000001010000100001000100011111

Xor

01000001010000100101001001011011
K9=
01000001010000100101001001011011
P9 Xor K9
00000000000000000100010101000110
01000001010000100101001001011011

Xor
01000001010000100001011100011101
P10=FE= 0100011001000101
P11=GH=0100011101001000
P12=HG=0100100001000111
P13=TG= 0101010001000111
P14=ED= 0100010101000100
K10=
01000001010000100001011100011101
P10 Xor K10
00000000000000000100011001000101
01000001010000100001011100011101

Xor
01000001010000100101000101011000
K11=
01000001010000100101000101011000
P11 Xor P11
00000000000000000100011101001000
01000001010000100101000101011000

Xor
01000001010000100001011000010000

K12=
01000001010000100001011000010000
P12 Xor K12
00000000000000000100100001000111
01000001010000100001011000010000

Xor
01000001010000100101111001010111
K13=
01000001010000100101111001010111
P13 Xor K13
00000000000000000101010001000111
01000001010000100101111001010111

Xor
01000001010000100000101000010000
K14=
01000001010000100000101000010000
P14 Xor K14
00000000000000000100010101000100
01000001010000100000101000010000

Xor
01000001010000100100111101010100
Langkah III
P15=MI= 0100110101001001
P16=OP= 0100111101010000
K15=
01000001010000100100111101010100
P15 Xor K15
00000000000000000100110101001001
01000001010000100100111101010100

Xor
01000001010000100000001000011101

K16=
01000001010000100000001000011101
P16 Xor K16
00000000000000000100111101010000
01000001010000100000001000011101

Xor
01000001010000100100110101001101
Langkah IV ganti P1 dan P2 dengan keluaran langkah III
P17=
10000010100001000001101000000001
P18=01000001010000100100001101001000
Langkah V enkripsikan langkah ke II dengan keluaran III
P1=01000001010000100000110100000001
P2=01000001010000100100001101001000
P3=01000001010000100001000000011101
P4=01000001010000100100001001010100
P5=01000001010000100001101100010101
P6=01000001010000100101010101011100
P7=01000001010000100001000100011111
P8=01000001010000100101001001011011
P9=01000001010000100001011100011101
P10=01000001010000100101001010110000
P11=01000001010000100001011000010000
P12=01000001010000100101111001010111
P13=01000001010000100000101000010000
P14=01000001010000100100111101010100

P15=0100000101000010000000100001110
1
P16=0100000101000010010011010100110
1
P17=0100000101000010000011010000000
1
P18=
0100000101000010010000110100100
0
Kunci (kunci yang baru)= opqr =
0111000001110001011100100111001
1

P1 Xor K1
0100000101000010000011010000000
1
0111000001110001011100100111001
1

Xor
0011000100110011011111110111001
0
K2=0011000100110011011111110111001
0
P2 Xor K2
0100000101000010010000110100100
0
0011000100110011011111110111001
0

-Xor
0111000001110001001111000011101
0
K3=0111000001110001001111000011101
0
P3 Xor K3
0100000101000010000100000001110
1
0111000001110001001111000011101
0

--- Xor
0011000100110011001011000010011
1
K4=0011000100110011001011000010011
1
P4 Xor K4
0100000101000010010000100101010
0
0011000100110011001011000010011
1

--- Xor
0111000001110001011011100111001
1
K5=0111000001110001011011100111001
1
P5 Xor K5
0100000101000010000110110001010
1
0111000001110001011011100111001
1

--- Xor
0011000100110011011101010110011
0
K6=0011000100110011011101010110011
0
P6 Xor K6
0100000101000010010101010101110
0
0011000100110011011101010110011
0

--- Xor
0111000001110001001000000011101
0
K7=0111000001110001001000000011101
0
P7 Xor K7
0100000101000010000100010001111
1
0111000001110001001000000011101
0

--- Xor
0011000100110011001100010010010
1
K8=0011000100110011001100010010010
1
P8 Xor K8
0100000101000010010100100101101
1
0011000100110011001100010010010
1

--- Xor
0111000001110001011000110111111
0
K9=0111000001110001011000110111111
0
P9 Xor K9
0100000101000010000101110001110
1
0111000001110001011000110111111
0

--- Xor
0011000100110011011101000110001
1
K10=0011000100110011011101000110001
11
P10 Xor K10
0100000101000010010100010101100
0
0011000100110011011101000110001
1

--- Xor
0111000001110001001001010011101
1

K11=
0111000001110001001001010011101
1
P11 Xor K11
0100000101000010000101100001000
0
0111000001110001001001010011101
1

--- Xor
0011000100110011001100110010101
1
K12=0011000100110011001100110010101
11
P12 Xor K12
0100000101000010010111100101011
1

00110001001100110011001100101011

--- Xor
01110000011100010110110101111100
K13=01110000011100010110110101111100
P13 Xor K13
01000001010000100000101000010000
01110000011100010110110101111100

--- Xor
00110001001100110110011101101100
K14=00110001001100110110011101101100
P14 Xor K14
01000001010000100100111101010100
00110001001100110110011101101100

--- Xor
01110000011100010010100000111000
K15=01110000011100010010100000111000
P15 Xor K15
01000001010000100000001000011101
01110000011100010010100000111000

--- Xor
00110001001100110010101000100101

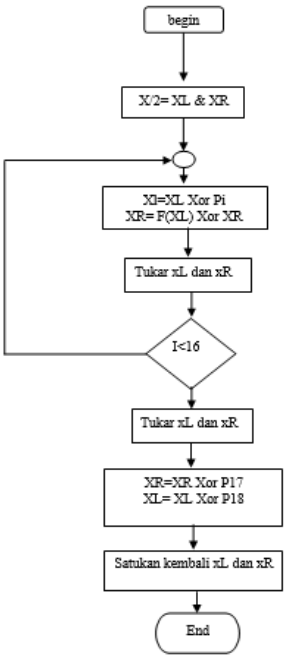
K16=00110001001100110010101000100101
P16 Xor K16
01000001010000100100110101001101
00110001001100110010101000100101

--- Xor
01110000011100010110011101101000
K17=01110000011100010110011101101000
P17 Xor K17
01000001010000100000110100000001
01110000011100010110011101101000

--- Xor
00110001001100110110101001101001
K18=00110001001100110110101001101001
P18 Xor K18
01000001010000100100001101001000
00110001001100110110101001101001

--- Xor
01110000011100010010100100100001

Prosedur ini digunakan untuk melakukan proses *Blowfish*. Rincian prosesnya dipaparkan oleh algoritma berikut :

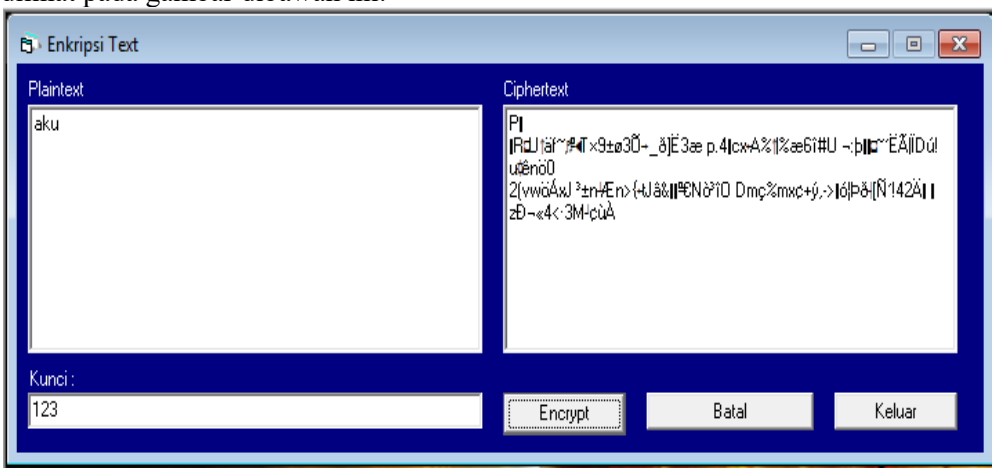


Gambar 2 Algoritma Blowfish

1. *Begin*
2. *Input* blok *plaintext* 64 bit

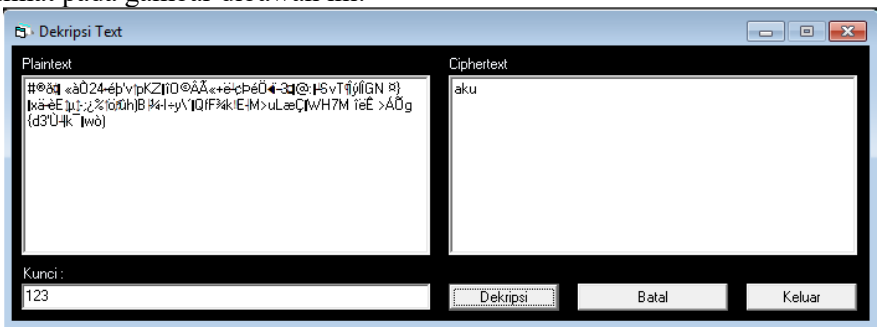
3. Partisi blok *plaintext* menjadi 2 sub blok 32 bit, XL=32 bit dan XR=32 bit
4. *Input* blok kunci enkripsi 32 bit
5. Partisi blok kunci enkripsi menjadi 4 sub blok kunci k0=8 bit, k1=8 bit, k2=8 bit, k3=8 bit.
6. I=1
7. Tahap I enkripsi XL= XL Xor Pi dan XR= F(XL) Xor XR
8. Tahap II enkripsi XR= XR Xor P16 dan XL= XL Xor P17
9. I=I-1
10. Periksa apakah I <=16
11. Jika tidak kembali ke langkah (7)
12. Jika ya, gabung *ciphertext*
13. *End*

Pada tampilan *form Enkripsi* adalah untuk memasukkan teks yang ingin di enkripsi dapat dilihat pada gambar dibawah ini:



Gambar 3 Tampilan *Form Input* Enkripsi

Pada tampilan *form Dekripsi* adalah untuk memasukkan teks yang ingin di dekripsi dapat dilihat pada gambar dibawah ini:



Gambar 4 Tampilan *Form Input* Dekripsi

4. KESIMPULAN

Cara kerja Algoritma *Blowfish* dengan memanfaatkan teknik pemanipulasi bit dan teknik pemutaran ulang dan bergiliran. Cara menerapkan Aplikasi pengaman pesan teks dengan Algoritma *Blowfish*, dengan cara pesan tersebut dienkripsi dan didekripsi lalu diubah ke kode ASCII.

REFERENCES

[1] T. S. Alasi, “Penerapan Algoritma Algoritma Boyer Moore untuk Penyaringan Pesan dan Algoritma Hill Cipher dalam Keamanan Pesan Teks Berbasis Web Chat,” *KAKIFIKOM Kumpul. Artik. Karya Ilm. Fak. Ilmu Komput*, vol. 1, no. 2, pp. 73–79, 2019.

[2] T. S. Alasi, “Implementasi Kriptografi Dengan Algoritma Ceasar Cipher Untuk Keamanan Data Microsoft Office Word Dan Excel,” *J. Inf. Komput. Log.*, vol. 1, no. 2, 2019.

[3] M. Natsir, “Pengembangan Prototype Sistem Kriptografi Untuk Enkripsi Dan Dekripsi Data Office Menggunakan Metode Blowfish Dengan Bahasa Pemrograman Java,” *J. Format*, vol. 6, no. 1, pp. 87–105, 2017.

[4] H. P. Martosedono and P. Purwanto, “IMPLEMENTASI ALGORITMA BLOWFISH DAN ALGORITMA RC4 PADA APLIKASI KEAMANAN EMAIL,” *SKANIKA*, vol. 1, no. 2, pp. 545–550, 2018.

[5] D. Andriani, “Perancangan Aplikasi Penyandian Teks Dengan Menggunakan Algoritma Chiper Block Chaining,” *J. Tek. Inform. UNIKA St. Thomas*, pp. 14–23, 2017.

[6] D. Dharma, “Studi Perbanding Penggunaan Algoritma Hash SHA 256 dengan Simetrik dan Asimetrik Ciphers dalam Perancangan Secure SWF Rich Internet Application (RIA),” 2017.