Kombinasi Algoritma OTP Cipher dan Algoritma BBS dalam Pengamanan File

Tomoyud S. Waruwu[1], Kristian Telaumbanua[2]
Universitas Sumatera Utara, Jl. Dr. T. Mansyur No. 9, Medan, Sumatera Utara
[1]Program Studi Magister Teknik Informatika, Universitas Sumatera Utara, Medan
[2]Jurusan Teknik Informatika, STMIK Mikroskil, Medan
[*1]tomoyud@gmail.com, [2]kristian@mikroskil.ac.id

Abstrak

Kriptografi adalah ilmu menyamarkan pesan sehingga hanya dikenal oleh penyedia dan penerima. Salah satu algoritma yang cukup aman dan sulit untuk memecahkan adalah satu-waktu algoritma One Time Pad (OTP) cipher. Namun, algoritma ini sangat tergantung pada keacakan kunci yang digunakan dan kunci hanya digunakan satu kali saja. Algoritma Blum Blum Shub (BBS) adalah sebuah algoritma untuk menghasilkan angka acak yang baik dan sulit untuk memprediksi. kombinasi dari kedua algoritma ini diharapkan untuk menciptakan keamanan data yang lebih baik dan terjamin.

Kata kunci*— kriptografi, algoritma OTP, algoritma BBS*

Abstract

Cryptography is the science of disguising messages so that only well known by the provider and the recipient. One algorithm that is sufficiently secure and difficult to solve is the one-time pad cipher algorithm. However, this algorithm is very dependent on the randomness of the key used and the key is only used one time only. blum blum shub algorithm is an algorithm to generate good random numbers and difficult to predict. the combination of these two algorithms are expected to create more and better data security and guaranteed.

Keywords*— kriptografi, algoritma OTP, algoritma BBS*

1. PENDAHULUAN

Semakin pesatnya kemajuan teknologi memberikan berbagai kemudahan bagi setiap pihak dalam melakukan pertukaran informasi. Namun, kemudahan ini juga membawa ancaman karena banyak pihak yang tidak berwenang yang berusaha untuk mengambil informasi tersebut untuk kepentingan pribadi atau organisasi.Untuk mengatasi ancaman ini, banyak pihak yang berusaha untuk menyandikan informasi yang mereka miliki sehingga pesan tersebut tidak memilik makna dan sulit untuk dipecahkan. Salah satu cara yang dapat diterapkan dalam menyandikan pesan atau informasi tersebut adalah melakukan kriptografi. Kriptografi merupakan metode untuk mengamankan data, baik berupa teks maupun gambar. Metode ini dilakukan dengan melakukan penyandian pesan kedalam bentuk yang tidak dipahami oleh orang lain maupun pihak ketiga. [1]

Secara umum ada dua tipe algoritma kriptografi berdasarkan kuncinya yaitu algoritma asimetris dan simetris. Algoritma asimetris terdiri atas 2 buah kunci yaitu kunci publik dan kunci privat. Kunci publik untuk melakukan enkripsi sedangkan kunci privat untuk melakukan dekripsi. Sedangkan algoritma simetris adalah algoritma yang memiliki kunci enkripsi dan dekripsi yang sama. Kriptografi kunci simetris memiliki berbagai macam metode algoritma, salah satunya adalah algoritma One Time Pad Cipher (OTP). [2]

Algoritma One Time Pad (OTP) Cipher merupakan algoritma berjenis symmetric key yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan deskripsi merupakan kunci yang sama.

Dalam proses enkripsi, algoritma ini menggunakan cara stream cipher dimana cipher tersebut berasal dari hasil XOR antara bit plaintext.

Untuk mendapatkan hasil yang enkripsi yang maksimal, Algoritma One Time Pad (OTP) Cipher membutuhkan sebuah kunci random yang hanya digunakan dalam satu kali siklus pengiriman pesan. Dalam mendapatkan sebuah kunci random, terdapat beberapa algoritma yang telah teruji. Salah satu algoritma tersebut adalah algoritma Blum Blum Shub (BBS). Tujuan dalam menggunakan algoritma Blum Blum Shub ini adalah agar kunci yang dihasilkan lebih sulit ditebak sehingga mempersulit kriptanalis dalam membaca pesan atau informasi tersebut.

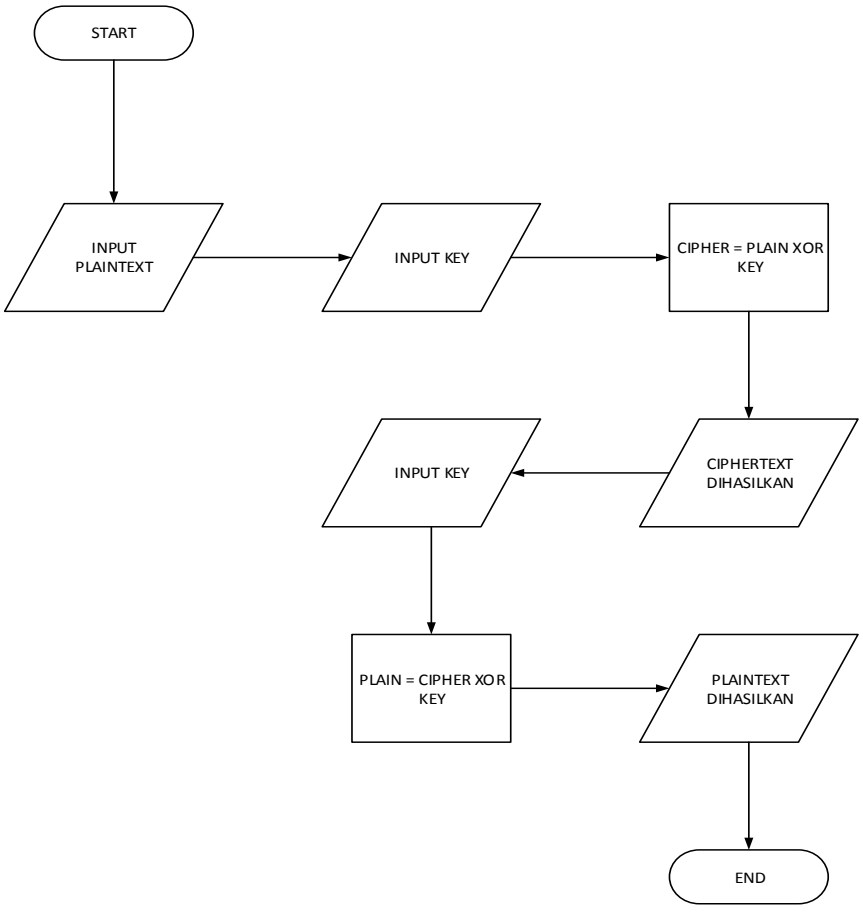
2. METODE PENELITIAN

2.1. Analisis Algoritma One Time Pad (OTP) Cipher

Secara teoritik, algoritma one time pad merupakan algoritma enkripsi yang sempurna (*perfect encryption*) asalkan menggunakan kunci yang benar acak dan tidak diproduksi ulang. Algoritma one time pad menggunakan kunci yang sama dalam proses enkripsi dan dekripsi serta memanfaatkan operasi xor pada proses enkripsi maupun dekripsi. Operasi xor menghasilkan nilai 0 apabila argumen sama (0 dengan 0 atau 1 dengan 1) dan menghasilkan 1 apabila argumen berbeda (0 dengan 1 atau 1 dengan 0).

Tabel 1 Tabel Operasi XOR

Bit Plaintext	Bit Kunci	Bit Hasil
0	0	0
1	0	1
0	1	1
1	1	0



Gambar 1 Flowchart Enkripsi dan Dekripsi Algoritma One Time Pad

Tahapan enkripsi dalam algoritma *one time pad* adalah:
1. Ubah plaintext kedalam bentuk biner
2. Bangkitkan kunci acak dan panjang kunci acak harus sama dengan plaintext yang akan dienkripsi.
3. Lakukan proses operasi xor terhadap plaintext dengan kunci yang telah dibangkitkan sebelumnya.
4. Ciphertext dihasilkan.

Sedangkan pada tahapan dekripsi algoritma *one time pad* adalah:
1. Ubah ciphertext kedalam bentuk biner.
2. Ambil kunci yang digunakan pada proses enkripsi.
3. Lakukan proses operasi xor terhadap plaintext dengan kunci yang telah dipilih.
4. Plaintext dihasilkan.

Ketangguhan algoritma *one time pad* tergantung pada keacakan kunci yang digunakan. Algoritma *one time pad* dapat digunakan untuk komunikasi yang rahasia dengan nilai yang tidak terlalu besar, namun pada skala yang besar, algoritma *one time pad* menjadi tidak praktik karena kendala pada manajemen kunci atau distribusi kunci tersebut.

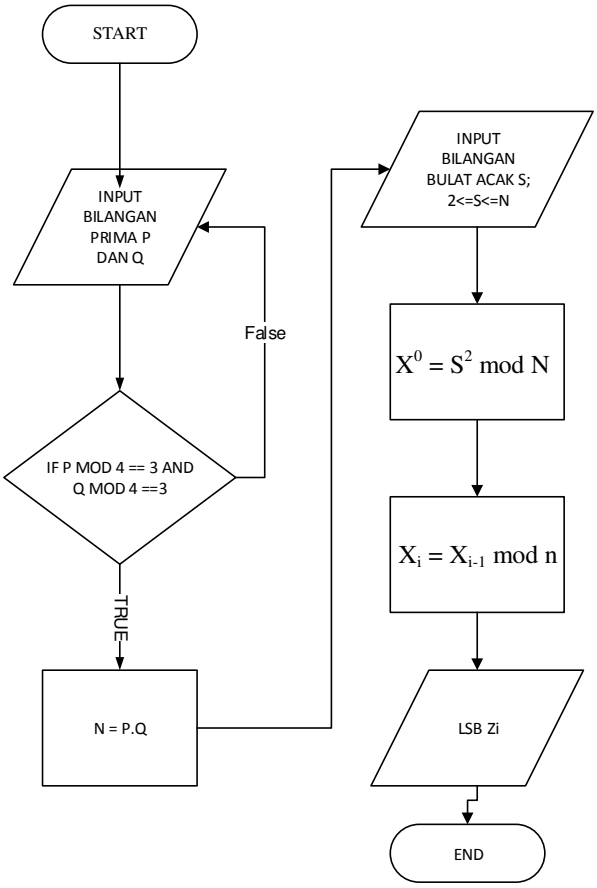
2.2. Analisis Algoritma Blum Blum Shub (BBS)

Dalam penelitian ini, CSPRNG yang digunakan adalah algoritma *blum blum shub*. CSPRNG ini dirancang berdasarkan teori bilangan. Algoritma ini terkenal karena kesederhanaannya dalam proses perhitungan namun tetap menghasilkan bilangan acak yang aman. Proses pembangkitan kunci dihasilkan melalui perhitungan :

$x_{n+1} = (x_n)^2 \bmod N$, dimana dalam hal ini, nilai N=p.q. Kedua nilai prima, *p* dan *q*, harus kongruen dengan 3 (mod 4) (hal ini akan menjamin tiap *quadratic residue* memiliki satu akar kuadrat yang juga merupakan *quadratic residue*) dan gcd(φ(*p*-1), φ(*q*-1)) juga harus kecil (hal ini membuat panjang siklus menjadi besar). Sebagai contoh, nilai p=947, q=523 sehingga nilai n=pq=495281, kemudian nilai s=164317, sehingga nilai $X_0=s^2 \bmod n=328055$, maka salah satu hasil yang bisa didapatkan antara lain:

Tabel 2 Pembangkit Kunci menggunakan Algoritma BBS

X_i	Bilangan Acak	LSB (Z_i)
1	474535	1
2	492608	0
3	210995	1
4	62059	1
5	14425	1
6	62605	1
7	227472	0
8	18871	1
9	7602	0



Gambar 2 Flowchart Algoritma Blum Blum Shub

2.3. Kriptografi

Kriptografi berasal dari bahasa Yunani dan terdiri atas dua kata, kryptos yang berarti tersembunyi dan graphein yang berarti menulis. Kriptografi pada awalnya didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Namun demikian, kriptografi berkembang sehingga tidak hanya terbatas pada menyandikan pesan, tetapi juga memberikan aspek keamanan lain. Oleh karena itu, definisi kriptografi diperbarui menjadi ilmu dan seni untuk meningkatkan aspek keamanan pesan.

Ada empat tujuan mendasar dari ilmu kriptografi yan juga merupakan aspek keamanan informasi yaitu:

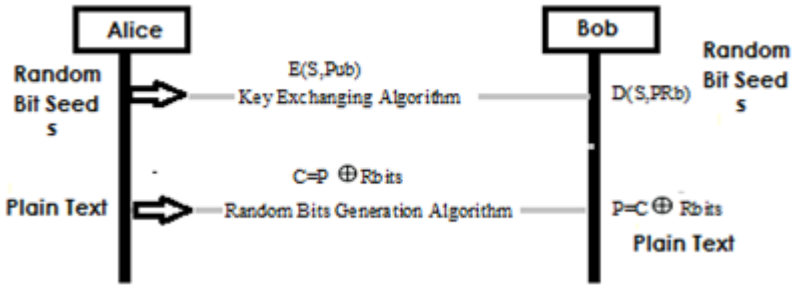
1. Kerahasiaan
   Layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas Data
   Berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data dari pihak-pihak yang tidak berwenang, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi
   Berhubungan dengan identitas/ pengenalan, baik secara kesatuan sistem atau informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri.
4. Non repudiasi
   Tidak ada penyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat. [1]

2.4. Algoritma One Time Pad (OTP) Cipher

Algoritma One Time Pad (OTP) cipher adalah sebuah metode yang menerapkan algoritma kunci simetris atau proses enkripsi dan dekripsi menggunakan kunci yang sama. Kerahasian kunci merupakan faktor utama dalam penentuan keamanan atau pesan yang dikirimkan. Algoritma vernam cipher diciptakan oleh Mayor J. Maugboune dan Gilbert Vernam pada tahun 1917.

Algoritma One Time Pad *cipher* adalah algoritma kriptografi yang sederhana dan mudah diimplementasikan karena hanya melakukan operasi XOR dan sudah dinyatakan oleh oleh para ahli kriptografi sebagai "*Perfect encryption Algorithm"* [3]. Sebagaimana yang telah dilakukan oleh Rinaldi Munir, [4] disimpulkan bahwa algoritma one time pad *cipher* tidak dapat dipecahkan (*unbreakable)* karena 2 alasan :

1. Barisan kunci acak yang ditambahkan ke pesan plainteks yang tidak acak menghasilkan *cipher*teks yang seluruhnya acak. *Cipher*teks ini tidak mempunyai hubungan statistik dengan plainteks.
2. Karena *cipher*teks tidak mengandung informasi apapun perihal plainteks, maka tidak mungkin ada cara untuk memecahkan *cipher*teks. Beberapa barisan kunci yang digunakan untuk mendeskripsikan mungkin menghasilkan plainteks yang mempunyai makna, sehingga kriptanalis tidak punya cara untuk menentukan plainteks mana yang benar.



Gambar 3 Enkripsi dan Dekripsi Algoritma One Time Pad (Penchalaiah et al., 2013)

Proses enkripsi maupun dekripsi, algoritma One Time Pad (OTP) cipher melakukan secara karakter per karakter atau sering juga disebut dengan metode stream cipher. Pada stream cipher, bila terjadi kesalahan selama transmisi maka kesalahan pada teks enkripsi penerima akan terjadi tepat di tempat kesalahan tersebut terjadi. Hasil enkripsi merupakan nilai yang didapatkan dari XOR antara plaintext dengan bit key. ada beberapa persyataran utama agar algoritma vernam cipher dapat digunakan secara maksimal :

1. Pemilihan kunci harus dilakukan secara acak agar tidak mudah ditebak
2. Jumlah karakter kunci harus sama panjang dengan karakter plain text.
3. Jika kunci tidak dapat diproduksi ulang maka algoritma dinyatakan aman.

2.5. Algoritma Blum Blum Shub

Pembangkitan kunci secara acak didalam kriptografi harus memiliki nilai yang tidak terduga dan sulit ditebak oleh musuh (Marton et al,. 2012). Bilangan acak yang cocok dengan kriptografi adalah *cryptographically secure pseudorandom number generator (CSPRNG)*. Adapun persyaratan dari CSPRNG adalah :

1. Secara statistik ia mempunyai sifat-sifat yang bagus (lulus uji keacakan statistik)
2. Tahan terhadap serangan (attack) yang serius. Serangan ini bertujuan untuk memprediksi bilangan acak yang dihasilkan. [5]

Algoritma Blum Blum Shub (*BBS*) adalah *CSPRNG* yang paling sederhana dan paling mangkus (secara kompleksitas teoritis). *BBS* dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum dan Michael Shub [6]. Algoritma *BBS* dapat dijelaskan secara singkat sebagai berikut:

1. Pilih dua buah bilangan prima rahasia p dan q, yang masing-masing kongruen 3 modulo 4 (dalam praktek bilangan prima yang digunakan cukup besar).
2. Kalikan keduanya menjadi $n = pq$. Bilangan n ini disebut **bilangan bulat Blum.**

3. Pilih bilangan bulat acak lain, *s*, sebagai umpan sedemikian sehingga: (i) $2 \leq s \leq n$ (ii) *s* dan *n* relatif prima kemudian hitung $x_0 = s^2 \bmod n$
4. Barisan bit acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan
   a. Hitung $x_i = x_{i-1}^{\,e} \bmod n$ degan $x_0 = s$.
   b. z_i = bit *LSB* (*Least Significant Bit*) dari x_i
5. Barisan bit acak yang dihasilkan adalah $z_1, z_2, z_3, \ldots$

Sebagai contoh, kita memilih p=11 dan q=19 sehingga n=pq = 209, selanjutnya kita pilih s=3. Kemudian untuk nilai e didapatkan dari rumus GCD(e,m)=1, oleh karena itu kita harus mendapatkan nilai m terlebih dahulu. m= (p-1)-(q-1) sehingga didapat m=(11-1)*(19-1), m=180. Kmudian untuk mencari nilai e, GCD(e,m), GCD(e,180)=1 sehingga nilai e=7. Setelah kita mendapatkan semua variabel, maka gunakan ketentuan yang ada untuk membangkitkan kunci tersebut dengan $X_{0=}s$, sehingga $X_{0=}3$.

$x_i = x_{i-1}^{\,e} \bmod n$ dengan x_0 demikian seterusnya.

Perhatikan bahwa nilai x_i yang mungkin hanya terletak antara 1 sampai 209 saja, sehingga pada suatu saat barisan tersebut akan berulang. Akibatnya, barisan bit yang dihasilkan pun juga akan berulang.

Jadi untuk nilai *n* yang kecil, maka *CSPRNG Blum Blum Shub* dapat dikatakan tidak aman, karena jika penyerang sudah mengetahui pola periodenya, maka tidak akan sulit untuk menebak bit yang dibangkitkan berikutnya. Sebenarnya untuk *n* besar sekalipun, jika pemilihan *s* tidak bagus, maka bisa terjadi periodenya kecil. Jika ini terjadi, penyerang juga bisa mengetahui barisan bit-bit yang dibangkitkan berikutnya. Bilangan acak yang diinginkan tidak harus 1 bit lsb, tetapi bisa juga j buah bit.

3. HASIL DAN PEMBAHASAN

3.1 Proses Enkripsi algoritma One Time Pad (OTP) dengan algoritma Blum Blum Shub (BBS)

Pada tahap ini, penulis akan mengimplementasikan penggunaan algoritma one time pad (OTP) cipher dan algoritma blum blum shub dalam pengamanan file. Plaintext yang akan dienkripsi adalah "Medan." Plaintext yang diberikan akan diubah kedalam bentuk desimal. Hasil pengubahan plaintext kedalam bentuk desimal adalah 77, 101, 100, 97, 110. Kemudian akan dibangkitkan kunci acak dengan nilai P = 526263477391, Q = 729324594683, N = P.Q = 383816897344657209312053, dan S = 377664220571268711194734, $X_0 = S^2 \bmod n$ = 177524508084670883192546 Kemudian dilakukan proses perhitungan dengan rumus $X_i = (X_{i-1}^2) \bmod N$, maka bilangan acak yang dihasilkan adalah 298813050508576857224146, 114464463378190093518026, 771355043559556997796691, 339247449786065473874696, 349727930184902188284497, 190373681903597488091751, 255759834524826056598806, 351106007173016248955579, 379673073177900629000279, 100117124357701197451810. Bilangan acak yang dihasilkan akan diambil 4 bit lsb kemudian digabungkan sehingga membentuk 8 bit lsb yang nantinya akan digunakan sebagai kunci pada proses enkripsi. Proses penggabungan 4 bit lsb menjadi 8 bit lsb untuk menjadi sebuah kunci, dapat dilihat pada tabel berikut :

Tabel 3 Kunci Enkripsi Algoritma One Time Pad menggunakan Algoritma Blum Blum Shub

No	Bilangan Acak	Kunci	Desimal
1	11111101000110101100000001101111100100010100000011010010000011011100111101<u>0010</u>	00101010	42
2	11000001111010010000110001000110001111011111011001000110011000000000100110<u>01010</u>		
3	10000010101011000011000101001011100100101001100000111011001011001010110<u>0011</u>	00111000	56
4	10001111101011010100100000111110101101001100111011100001101100110101011000<u>01000</u>		
5	10010100000111011001010000000010110110100101110010110111001111001001110100<u>10001</u>	00010111	23

No	Bilangan Acak	Kunci	Desimal
6	10100001010000001011011000111111001000010100111010001011001010101100100110111		
7	11011000101000110001001011011010011010101001100111000000110101001001010001110	01101011	107
8	10010100101100101111110101010100110010010000001100010100100010010110010101111011		
9	10100000110011000011110010110001000110100100101010110010010101010000000010110111	01110010	114
10	10101001100110101110000110101001100000001001000010100001010010101001000100010		

Setelah kunci terbentuk, maka dilakukan proses xor antara plaintext yang digunakan dengan kunci yang dihasilkan. Dari hasil percobaan didapatkan hasil :

$C[i] = P[i] \text{ xor } K[i] \bmod 256$
$C[] = P[1] \text{ xor } K[1] \bmod 256$
$C[1] = (77 \text{ xor } 42)\ 256$
$C[1] = 103 = g$

$C[i] = P[i] \text{ xor } K[i] \bmod 256$
$C[2] = P[2] \text{ xor } K[2] \bmod 256$
$C[2] = (101 \text{ xor } 56) \bmod 256$
$C[2] = 93 =]$

$C[i] = P[i] \text{ xor } K[i] \bmod 256$
$C[3] = P[3] \text{ xor } K[3] \bmod 256$
$C[3] = (100 \text{ xor } 23) \bmod 256$
$C[3] = 115 = s$

$C[i] = P[i] \text{ xor } K[i] \bmod 256$
$C[4] = P[4] \text{ xor } K[4] \bmod 256$
$C[4] = (97 \text{ xor } 107) \bmod 256$
$C[4] = 10 = <LF>$

$C[i] = P[i] \text{ xor } K[i] \bmod 256$
$C[5] = P[5] \text{ xor } K[5] \bmod 256$
$C[5] = (110 \text{ xor } 114) \bmod 256$
$C[5] = 28 = <FS>$

Sehingga ciphertext yang didapatkan pada percobaan pertama ini adalah g]s<LF><FS>

3.2 Proses Dekripsi algoritma One Time Pad (OTP) dengan algoritma Blum Blum Shub

Pada percobaan sebelumnya, ciphertext yang akan didekripsi adalah g]s<LF><FS> Ciphertext yang diberikan akan diubah kedalam bentuk desimal. Hasil pengubahan ciphertext kedalam bentuk desimal adalah 103, 93, 115, 10, 28. Kunci yang digunakan adalah 42, 56, 23, 107, 114. Kemudian dilakukan proses xor antara ciphertext yang digunakan dengan kunci yang dihasilkan. Dari hasil percobaan didapatkan hasil:

$P[i] = C[i] \text{ xor } K[i] \bmod 256$
$P[1] = C[1] \text{ xor } K[1] \bmod 256$
$P[1] = (103 \text{ xor } 42) \bmod 256$
$P[1] = 77 = M$

$P[i] = C[i] \text{ xor } K[i] \bmod 256$
$P[2] = C[2] \text{ xor } K[2] \bmod 256$
$P[2] = (93 \text{ xor } 56) \bmod 256$
$P[2] = 101 = e$

$P[i] = C[i] \text{ xor } K[i] \bmod 256$
$P[3] = C[3] \text{ xor } K[3] \bmod 256$
$P[3] = (115 \text{ xor } 23) \bmod 256$
$P[3] = 100 = d$

$P[i] = C[i] \text{ xor } K[i] \bmod 256$
$P[4] = C[4] \text{ xor } K[4] \bmod 256$
$P[4] = (10 \text{ xor } 107) \bmod 256$
$P[4] = 97 = a$

$P[i] = C[i] \text{ xor } K[i] \bmod 256$
$P[5] = C[5] \text{ xor } K[5] \bmod 256$
$P[5] = (28 \text{ xor } 114) \bmod 256$
$P[5] = 110 = n$

Sehingga plaintext yang didapatkan adalah “Medan”

4. KESIMPULAN

Dari penjelasan diatas, ada beberapa kesimpulan yang penulis dapatkan yaitu:

1. Algoritma One Time Pad Cipher (OTP) adalah algoritma yang sederhana karena menggunakan operasi XOR dalam proses enkripsi data.
2. Kunci yang digunakan hanya dipakai satu kali dan panjang kunci harus sama dengan plaintext
3. Kunci yang dibangkitkan harus merupakan kunci acak sehingga mempersulitkan kriptanalis dalam mengetahui pesan tersebut.
4. Algoritma blum blum shub adalah *cryptographically secure pseudorandom number generator* (CSPRNG) yang paling sederhana dan paling mangkus (secara kompleksitas teori)
5. Keamanan algoritma blum blum shub terletak pada sulitnya memfaktorkan n. Nilai n tidak perlu rahasia dan dapat diumumkan secara publik
6. Algoritma blum blum shub tidak dapat diprediksi dari arah kiri (*unpredictable to the left*) dan tidak dapat diprediksi dari arah kanan (*unpredictable to the right)* artinya jika diberikan barisan bit yang dihasilkan oleh algoritma blum blum shub, kriptanalis tidak dapat memprediksi barisan bit sebelumnya dan barisan bit sesudahnya.

5. SARAN

Saran Penulis terhadap penelitian ini adalah :

1. Untuk menjaga keamanan dari algoritma *blum blum shub* sebaiknya bilangan besar p dan q adalah bilangan prima dengan panjang lebih besar dari 2^{1024} dan menggunakan algoritma pencarian bilangan prima yang tepat sehingga waktu yang dibutuhkan untuk menghasilkan bilangan acak semakin efesien
2. Untuk penelitian lebih lanjut diharapkan proses enkripsi dan dekripsi dapat mencakup pada data image, suara maupun video

DAFTAR PUSTAKA

[1] Scheiner, B. 1996. *Applied Cryptography-2nd Edition: Protocols, Algorithms, and Source in C*. John Wiley & Sons, Inc: Seattle.

[2] Chehal, R. & Singh, K. 2012. Efficiency and security of data with symmetric encryption algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol 2, Issue 8, 472-475

[3] Pechalaiah, P. & Reddy, K.R. 2013. Efficient and Secure Encryption Schema on Random bit's (Rbits). *International Journal of Advanced Research in Computer,* Vol 3, Issue 10, 1026-1032

[4] Munir, Rinaldi. 2011. Algoritma enkripsi citra dengan pseudo One-Time Pad yang menggunakan sistem chaos. *Konferensi Nasional Informatika*. ISSN:2087-3328:12-16

[5] Marton, K., Suciu, A., Sacarea, C. & Cret, O., 2012. Generation and testing of random number for cryptographic applications. *Proceeding of Romanian Academy,* Vol 13, Issue 3, 368-377.

[6] Blum, L., Blum, M. & Shub, M., 1986. A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing*, Vol 15, Issue 2, 364-383

[7] Blum, M. & Micali, S. 1982. How to generate cryptographically strong sequence of pseudo random bits. *IEEE 23rd Symposium on the Foundations of Computer Science.* pp. 112-117

[8] Apoorva. & Kumar, Y. 2013. Comparative study of different symmetric key cryptography algorithms. *International Journal of Application or Innovation in Engineering and Management,* Vol 2, Issue 7, 204-206

[9] Elminaam, D.S.A., Kader, H.M.A. & Hadhoud, M.M., 2009. Performance evalution of symmetric encryption algorithms. *Communications of the IBIMA*, Vol 8,58-64

[10] Goel, S., Kaur, N. & Mandal, D. 2013. Analysis of channel coding technique using pseudorandom generator based public key cryptography. *International Journal of Engineering Science and Innovative Technology,* Vol 2, Issue 5, 512-519