Implementasi Keamanan Data Dengan Menerapkan Algoritma Rabbit Stream Cipher Untuk Penyandian Data Teks

Mei Diana Utami
Teknik Informatika STMIK Budi Darma Medan Jl. Sisingamangaraja No. 338 - Medan
e-mail : meidiana.utami@yahoo.com

Abstrak

Perkembangan ilmu pengetahuan dan teknologi yang demikian pesat telah memberikan banyak kemudahan bagi manusia dalam melakukan segala kegiatannya, termasuk dalam melakukan pertukaran informasi. Akan tetapi, terdapat pihak-pihak tertentu dengan berbagai kepentingan berusaha mengungkap pertukaran informasi yang dilakukan oleh pihak lainnya. Tentu hal ini menimbulkan suatu kerugian apabila informasi yang dipertukarkan merupakan informasi yang bersifat rahasia, misalnya informasi mengenai account pribadi pada suatu bank.

Rabbit merupakan salah satu algoritma cipher aliran yang diperkenalkan pada tahun 2003. Algoritma Rabbit menggunakan 128 bit kunci rahasia dan 64 bit Initialization Vector (IV) sebagai masukan untuk membangkitkan blok keluaran yang terdiri dari 128 bit acak semu (pseudo-random,, yang merupakan kombinasi dari bit-bit pada status internal, untuk setiap iterasi.

Proses enkripsi/dekripsi dilakukan dengan meng XOR-kan blok acak semu tersebut dengan plainteks/cipherteks. Ukuran dari status internal adalah 513 bit dibagi menjadi 8 variabel status dengan panjang 32 bit, 8 counter dengan panjang 32 bit, dan 1 bit carry untuk counter. Kedelapan variabel status diupdate dengan 8 buah fungsi non-linear.

***Kata Kunci** : Kriptografi, Rabbit Stream, Enkripsi, Dekripsi*

Abstract

The rapid development of science and technology has provided many conveniences for humans in carrying out all their activities, including in exchanging information. However, there are certain parties with various interests trying to uncover information exchanges carried out by other parties. Of course this causes a loss if the information exchanged is confidential information, for example information about a personal account at a bank.

Rabbit is one of the flow cipher algorithms introduced in 2003. Rabbit's algorithm uses 128 bits of secret keys and 64 bit Initialization Vector (IV) as input to generate output blocks consisting of pseudo-random 128 bits, which is a combination from bits in internal status, for each iteration.

The process of encryption / decryption is done by XOR the pseudo random block with plaintext / ciphertext. The size of the internal status is 513 bits divided into 8 status variables with a length of 32 bits, 8 counters with a length of 32 bits, and 1 bit carry for the counter. The eight status variables are updated with 8 non-linear functions.

***Keywords :** Kriptographi, Rabbit Stream, Algoritma Kriptografi*

1. PENDAHULUAN

Perkembangan teknologi digital serta internet saat ini telah memberi kemudahan untuk melakukan akses serta mendistribusikan berbagai informasi dalam format digital. Perkembangan teknologi digital serta internet ini dapat digunakan secara "negatif" seperti pencurian atau perusakan data pada dokumen digital. Hal ini mengakibatkan perlunya suatu sistem pengamanan data dalam sebuah dokumen digital sehingga menjamin kemanan dari dokumen digital tersebut, baik dalam bentuk teks, gambar, suara maupun video.

Teks ataupun tulisan merupakan dokumen yang paling banyak dibuat dibandingkan dengan dokumen gambar ataupun lainnya. Setiap harinya banyak dokumen yang berupa teks atau tulisan yang dihasilkan. Seiring dengan pertumbuhan dari dokumen elektronik ini timbul setidaknya dua masalah yaitu, pertama dari banyak dokumen yang dihasilkan beberapa diantaranya merupakan dokumen yang sifatnya rahasia dan pribadi. Hal ini tidak mengkhawatirkan apabila dokumen itu digunakan hanya untuk keperluan pribadi saja (tanpa perlu orang lain untuk mengetahui) karena dapat memanfaatkan fungsi *password* pada dokumen dan hanya diri kita yang mengetahui *password* tersebut, akan tetapi hal yang perlu diperhatikan adalah ketika dokumen yang bersifat rahasia ini diperlukan atau digunakan oleh banyak pihak sehingga *password* yang digunakan untuk mengamankan dokumen ini kini harus diketahui oleh beberapa orang. Oleh karena itu diperlukan suatu teknik pengamanan data yang lebih kompleks, misalnya dengan teknologi kriptografi.

Salah satu teknik pengamanan data yang sering dilakukan pada sebuah media digital adalah kriptografi. Kriptografi adalah sebuah teknik pengamanan data, di mana data yang akan diamankan diacak (*encrypt*) isinya sehingga tidak dapat dimengerti oleh pihak lain. Data yang telah di*encrypt* ini hanya dapat dibuka dan disusun kembali (*decrypt*) oleh aplikasi khusus yang ditentukan oleh *user* yang melakukan enkripsi, sehingga kemanan data dokumen digital tersebut terjamin, khususnya data teks. Kriptografi muncul di dasari atas berkomunikasi dan saling bertukar informasi/data secara jarak jauh.komunikasi dan pertukaran data antar wilayah dan negara ataupun benua bukan lagi menjadi suatu kendala yang berarti.

Penggunaan kriptografi dalam mengamankan dokumen digital di nilai sangat efektif, karena nilai *biner* data teks dalam dokumen digital tersebut berubah dan menghasilkan sebuah data teks yang berbeda dengan aslinya. Bentuk perubahan nilai biner ini tergantung dari metode kriptografi yang digunakan dalam mengenkripsi dokumen digital tersebut. Banyak teknik kriptografi yang telah dipergunakan untuk menjaga keamanan data saat ini, contohnya: LOKI, GOST, Blowfish, Vigenere, MD2, MD4, RSA dan lain sebagainya. Masing-masing teknik kritografi tersebut memiliki kelemahan dan kelebihan. Salah satu metode kriptografi yang efektif untuk mengamankan sebuah dokumen digital adalah metode *Rabbit Stream Cipher*. *Rabbit* merupakan salah satu algoritma cipher aliran yang diperkenalkan pada tahun 2003.

Algoritma Rabbit menggunakan 128 bit kunci rahasia dan 64 bit Initialization Vector (IV) sebagai masukan untuk membangkitkan blok keluaran yang terdiri dari 128 bit acak semu (pseudo-random) yang merupakan kombinasi dari bit-bit pada status internal, untuk setiap iterasi. Proses enkripsi/dekripsi dilakukan dengan meng- XOR-kan blok acak semu tersebut dengan plainteks/cipherteks. Ukuran dari status internal adalah 513 bit dibagi menjadi 8 variabel status dengan panjang 32 bit, 8 counter dengan panjang 32 bit, dan 1 bit carry untuk counter. Kedelapan variabel status diupdate dengan 8 buah fungsi non-linear.

2. METODOLOGI PENELITIAN

2.1. Kriptografi

Pengertian kriptografi modern tidak saja berurusan dengan penyembunyian pesan, namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [5].

Berikut ini adalah rangkuman beberapa mekanisme yang berkembang pada kriptografi modern :

a. Fungsi *Hash*.
   Fungsi *hash* adalah fungsi yang melakukan pemetaan pesan dengan dengan panjang sembarang ke seluruh teks khusus yang disebut *message digest* dengan panjang tetap. Fungsi *hash* umumnya dipakai sebagai nilai uji (*check value*) pada mekanisme keutuhan data.
b. Penyandian dengan kunci simetrik (*symmetric key encipherment*).
   Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi berniali sama. Penyandian dengan kunci simetrik disebut dengan penyandian kunci rahasia atau *secret key encripherment*,
c. Penyandian dengan kunci asimetrik (*Asymetric key encipherment*).
   Penyandian dengan kunci asimetrik atau penyandian dengan kunci publik (*public key*) adalah penyandian kunci enkripsi dan dekripsi berbeda nilai. Kunci enkripsi yang juga disebut dengan kunci publik (*public key*) bersifat terbuka. Sedangkan kunci dekripsi yang juga disebut kunci privat (*private key*) bersifat tertutup/rahasia [5].

2.2. Algoritma Kriptografi

Kriptografi merupakan bentuk algoritma untuk mengacak pesan dan mengembalikan acakan pesan tersebut, dimana pembuatan algoritma kriptografi tersebut mempergunakan perhitungan dan formula-formula matematika. Algoritma kriptografi adalah fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Untuk mengenkrip sebuah pesan *plaintext*, terapkan algoritma enkripsi ke pesan *plaintext* tersebut. Untuk mendekrip sebuah pesan *ciphertext*, terapkan algoritma dekripsi ke pesan *ciphertext* tersebut. Kerahasiaan berasal dari adanya algoritma yang kuat dan dipublikasikan dengan kunci yang panjang.

Syarat-syarat algoritma kriptografi yang baik antara lain,

a. Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan.
b. Algoritmanya memiliki ruang kunci (keyspace) yang besar.
c. Menghasilkan ciphertext yang terlihat acak dalam seluruh tes statistik yang dilakukan terhadapnya.
d. Mampu menahan seluruh serangan yang telah dikenal sebelumnya.

2.3. Algoritma Rabbit Stream Cipher

Algoritma Rabbit di publikasikan pertama kali pada Fast Software Encryption Workshop 2003. Algoritma ini merupakan algoritma stream cipher yang menggunakan kunci (secret key) dengan panjang 128- bit dan 64-bit IV sebagai input. Hasil kombinasi internal state-nya menghasilkan output berupa rangkaian bit semi acak (pseudorandom bit) dengan panjang 128-bit per blok. Proses enkripsi/dekripsi dilakukan dengan men-XOR-kan rangkaian bit semi acak tersebut dengan plaintext/ciphertext. Internal state-nya berjumlah 513 bit dibagi ke dalam delapan 32-bit variabel state, delapan 32-bit counter dan satu counter carry bit.

Kedelapan variabel state di-update oleh fungsi non-linear masing-masing pasangan variabel state tersebut. Algoritma Rabbit didesain agar efisien dalam implementasi pada software serta mengimbangi ukuran kunci sepanjang 128 bit untuk mengenkripsi sampai dengan 264 blok plaintext. Ini berarti bahwa untuk melakukan attack tanpa mengetahui rangkaian kunci, maka pihak penyerang harus menentukan sampai dengan 264 blok output ciphertext atau melakukan exhaustive key search sebanyak 2128 kombinasi kunci [3].

Algoritma di inisialisasi dengan memperluas kunci berukuran 128 bit menjadi 8 variabel status dan 8 variabel *counter* sehingga di peroleh korespondensi satu-ke-satu antara kunci dan variabel status awal, $x_{j,0}$ dan variabel *ounter* awal, $c_{j,0}$. Kunci awal, $k^{[127''0]}$, dibagi menjadi 8 sub-kunci, yaitu:

$k_0 = K^{[15,,0]}$
$\boldsymbol{k}_1 = K^{[31,,16]}$
$\boldsymbol{k}_2 = K^{[47,,32]}$
$k_3 = K^{[63,,48]}$
$k_4 = K^{[79,,64]}$
$k_5 = K^{[95,,80}$
$k_6 = K^{[111,,96]}$
$k_7 = K^{[127,,112]}$

Variabel status dan variabel *counter* di inisialisasi dari sub-kunci dengan aturan sebagai berikut:

$$x_{j,0} = \begin{cases} k_{(j+1 \bmod e\,\theta)} \circ k_j & jika\ j\ genap \\ k_{(j+5 \bmod e\,\theta)} \circ k_{(j+4 \bmod e\,\theta)} & jika\ j\ ganjil \end{cases} \quad (1)$$

Dan

$$c_{j,0} = \begin{cases} k_{(j+4 \bmod e\,\theta)} \circ k_{(j+5 \bmod e\,\theta)} & jika\ j\ genap \\ k_{(} \circ k_{(j+1 \bmod e\,\theta)} & jika\ j\ ganjil \end{cases} \quad (2)$$

Algoritma Rabbit untuk setiap saat akan hanya mengenkripsi plaintext sepanjang 16 karakter. Jika plaintext memiliki ukuran lebih panjang dari 16 karakter maka plaintext akan dibagi menjadi beberapa blok string dengan panjang masing-masing adalah 16 karakter. Untuk semua *j*, untuk mencegah kunci di ketahui dengan di lakukan pembalikan pada variabel-variabel *counter* [7].

3. HASIL DAN PEMBAHASAN

Proses pembentukan kunci ini memerlukan *input data key* dengan panjang 128 bit (16 buah karakter). Misalkan *key* : '1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16', maka proses pembentukan kunci dari *key* di atas menggunakan algoritma Rabbit Stream Cipher dapat diuraikan dengan langkah-langkah sebagai berikut:

3.1. Skema Key Setup

Tentukan panjang kunci awal sebanyak 128 bit.
Kunci (*key*) = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
kemudian ubah kunci ke bentuk biner :
'1' = 0 0 0 0 0 0 0 1
'2' = 0 0 0 0 0 0 1 0
'3' = 0 0 0 0 0 0 1 1
'4' = 0 0 0 0 0 1 0 0
'5' = 0 0 0 0 0 1 0 1
'6' = 0 0 0 0 0 1 1 0
'7' = 0 0 0 0 0 1 1 1
'8' = 0 0 0 0 1 0 0 0
'9' = 0 0 0 0 1 0 0 1
'10' = 0 0 0 0 1 0 1 0

'11' = 0 0 0 0 1 0 1 1
'12' = 0 0 0 0 1 1 0 0
'13' = 0 0 0 0 1 1 0 1
'14' = 0 0 0 0 1 1 1 0
'15' = 0 0 0 0 1 1 1 1
'16' = 0 0 0 1 0 0 0 0

Hasil konversi kunci ke bentuk biner :

K = 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0
0 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0

Bagi kunci awal menjadi 8 sub kunci.

$K_0 = K^{[15..0]}$ = 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
$K_1 = K^{[31..16]}$= 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0
$K_2 = K^{[47..32]}$= 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0
$K_3 = K^{[63..48]}$= 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0
$K_4 = K^{[79..64]}$= 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0
$K_5 = K^{[95..80]}$= 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0
$K_6 = K^{[111..96]}$= 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0
$K_7 = K^{[127..112]}$= 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 → *Bit Carry*

Selanjutnya dilakukan inisialisasi untuk memperluas kunci menjadi 8 variabel status dan 8 variabel counter sehingga diperoleh korespondensi satu-ke-satu antara kunci dan variabel status awal, $x_{j,0}$, dan variabel *counter* awal, $c_{j,0}$. Variabel status dan variabel *counter* di inisialisasi dari subkunci dengan aturan sebagai berikut :

$$x_{j,0} = \begin{cases} k_{(j+1 \bmod 8)} \lozenge k_j & Jika\ j\ genap \\ k_{(j+5 \bmod 8)} \lozenge k_{(j+4 \bmod 8)} & Jika\ j\ ganjil \end{cases}$$

dan

$$c_{j,0} = \begin{cases} k_{(j+4 \bmod 8)} \lozenge k_{(j+5 \bmod 8)} & Jika\ j\ genap \\ k_j \lozenge k_{(j+1 \bmod 8)} & Jika\ j\ ganjil \end{cases}$$

Maka dapat dihasilkan :

a. Variabel Status ($x_{j,4}$)

$x_{0,4} = k_1 \lozenge k_0$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
$x_{1,4} = k_6 \lozenge k_5$ = 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0
$x_{2,4} = k_3 \lozenge k_2$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0
$x_{3,4} = k_0 \lozenge k_7$ = 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0
$x_{4,4} = k_5 \lozenge k_4$ = 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0
$x_{5,4} = k_2 \lozenge k_1$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0
$x_{6,4} = k_7 \lozenge k_6$ = 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0
$x_{7,4} = k_4 \lozenge k_3$ = 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0

b. Variabel Counter ($c_{j,4}$)

$c_{0,4} = k_4 \lozenge k_5$ = 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0

$c_{1,4} = k_1 \lozenge k_2$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$c_{2,4} = k_6 \lozenge k_7$ = 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0

$c_{3,4} = k_3 \lozenge k_4$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0

$c_{4,4} = k_0 \lozenge k_1$ = 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{5,4} = k_5 \lozenge k_6$ = 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0

$c_{6,4} = k_2 \lozenge k_3$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0

$c_{7,4} = k_7 \lozenge k_0$ = 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

Selanjutnya lakukan modifikasi status counter dengan aturan sebagai berikut:

$c_{j,4} = c_{j,4} \oplus x_{(j+4 \bmod 8)}$

$c_{0,4} = c_{0,4} \oplus x_{(0+4 \bmod 8)} = c_{0,4} \oplus x_{4,4}$

$c_{0,4}$ = 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0

$x_{4,4}$ = 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0

$c_{0,4}$ = 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0

$c_{1,4} = c_{1,4} \oplus x_{(1+4 \bmod 8)} = c_{1,4} \oplus x_{5,4}$

$c_{1,4}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$x_{5,4}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{1,4}$ = 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0

$c_{2,4} = c_{2,4} \oplus x_{(2+4 \bmod 8)} = c_{2,4} \oplus x_{6,4}$

$c_{2,4}$ = 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0

$x_{6,4}$ = 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0

$c_{2,4}$ = 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0

$c_{3,4} = c_{3,4} \oplus x_{(3+4 \bmod 8)} = c_{3,4} \oplus x_{7,4}$

$c_{3,4}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0

$x_{7,4}$ = 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0

$c_{3,4}$ = 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0

$c_{4,4} = c_{4,4} \oplus x_{(4+4 \bmod 8)} = c_{4,4} \oplus x_{0,4}$

$c_{4,4}$ = 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$x_{0,4}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{4,4}$ = 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0

$c_{5,4} = c_{5,4} \oplus x_{(5+4 \bmod 8)} = c_{5,4} \oplus x_{1,4}$

$c_{5,4}$ = 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0

$x_{1,4}$ = 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0

$c_{5,4}$ = 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0

$c_{6,4} = c_{6,4} \oplus x_{(6+4 \bmod 8)} = c_{6,4} \oplus x_{2,4}$

$c_{6,4}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0

$x_{2,4}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$c_{6,4}$ = 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0

$c_{7,4} = c_{7,4} \oplus x_{(7+4 \bmod 8)} = c_{7,4} \oplus x_{3,4}$

$c_{7,4}$ = 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$x_{3,4}$ = 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0

$c_{7,4}$ = 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0

Untuk semua *j*, demi mencegah kunci diketahui maka dilakukan pembalikan pada variabel-variabel *counter*.

$c_{0,4}$ = 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0
$c_{0,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0
$c_{1,4}$ = 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0
$c_{1,4}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0
$c_{2,4}$ = 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0
$c_{2,4}$ = 0 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0
$c_{3,4}$ = 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0
$c_{3,4}$ = 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0
$c_{4,4}$ = 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0
$c_{4,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0
$c_{5,4}$ = 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0
$c_{5,4}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0
$c_{6,4}$ = 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0
$c_{6,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0
$c_{7,4}$ = 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0
$c_{7,4}$ = 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 ↓

Pada Skema IV Setup ini dilakukan untuk menetapkan nilai *Initialization Vector* (IV) dengan panjang 64 bit.

IV = 1, 2, 3, 4, 5, 6, 7, 8
IV = 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0
$IV^{[31..0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
$IV^{[63..32]}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

Selanjutnya lakukan modifikasi terhadap variabel *counter* dengan aturan sebagai berikut :
i=1

$c_{0,1} = c_{0,4} \oplus IV^{31...0}$

$c_{0,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0
$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
$c_{0,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0

$c_{1,4} = c_{1,4} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{1,4}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0
$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0
$c_{1,1}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0

$c_{2,1} = c_{2,4} \oplus IV^{63..32}$

$c_{2,4}$ = 0 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0
$IV^{[63..32]}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0
$c_{2,1}$ = 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0

$c_{3,1} = c_{3,4} \oplus (IV^{47...32} \lozenge IV^{15...0})$

$c_{3,4}$ = 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0
$IV^{47..32} \lozenge IV^{15...1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
$c_{3,1}$ = 0 1 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0

$c_{4,1} = c_{4,4} \oplus IV^{31...0}$

$c_{4,4}$ = 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0
$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{4,1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0

$c_{5,1} = c_{5,4} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{5,4}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0

$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{5,1}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0

$c_{6,1} = c_{6,4} \oplus IV^{31...0}$

$c_{6,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{6,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 0

$c_{7,1} = c_{7,4} \oplus (IV^{47...32} \lozenge IV^{15...0})$

$c_{7,4}$ = 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0

$IV^{47..32} \lozenge IV^{15...0}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{7,1}$ = 0 1 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0

i=2

$c_{0,2} = c_{0,1} \oplus IV^{31...0}$

$c_{0,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{0,2}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0

$c_{1,2} = c_{1,1} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{1,1}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0

$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{1,2}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0

$c_{2,2} = c_{2,1} \oplus IV^{63..32}$

$c_{2,1}$ = 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0

$IV^{[63..32]}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$c_{2,2}$ = 0 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0

$c_{3,2} = c_{3,1} \oplus (IV^{47...32} \lozenge IV^{15...0})$

$c_{3,1}$ = 0 1 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0

$IV^{47..32} \lozenge IV^{15...1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{3,2}$ = 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0

$c_{4,2} = c_{4,1} \oplus IV^{31...0}$

$c_{4,1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{4,2}$ = 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0

$c_{5,2} = c_{5,1} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{5,1}$ = 0 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1

$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{5,2}$ = 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 1

$c_{6,2} = c_{6,1} \oplus IV^{31...0}$

$c_{6,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{6,1}$ = 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0

$c_{7,2} = c_{7,1} \oplus (IV^{47...32} \lozenge IV^{15...0})$

$c_{7,1}$ = 0 1 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0

$IV^{47...32} \lozenge IV^{15...0}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{7,2}$ = 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0

i=3

$c_{0,3} = c_{0,2} \oplus IV^{31...0}$

$c_{0,2}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{0,3}$ = 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0

$c_{1,3} = c_{1,2} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{1,2}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0

$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{1,3}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0

$c_{2,3} = c_{2,2} \oplus IV^{63..32}$

$c_{2,2}$ = 0 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0

$IV^{[63..32]}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$c_{2,3}$ = 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0

$c_{3,3} = c_{3,2} \oplus (IV^{47...32} \lozenge IV^{15...0})$

$c_{3,2}$ = 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0

$IV^{47...32} \lozenge IV^{15...1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{3,3}$ = 0 1 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0

$c_{4,3} = c_{4,2} \oplus IV^{31...0}$

$c_{4,2}$ = 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{4,3}$ = 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0

$c_{5,3} = c_{5,2} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{5,2}$ = 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 1

$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{5,3}$ = 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1

$c_{6,3} = c_{6,2} \oplus IV^{31...0}$

$c_{6,2}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{6,3}$ = 0 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 0

$c_{7,3} = c_{7,2} \oplus (IV^{47...32} \lozenge IV^{15...0})$

$c_{7,2}$ = 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0

$IV^{47...32} \lozenge IV^{15...0}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{7,3}$ = 0 1 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0

i=4

$c_{0,4} = c_{0,3} \oplus IV^{31...0}$

$c_{0,3}$ = 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{0,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0

$c_{1,4} = c_{1,3} \oplus (IV^{63...48} \lozenge IV^{31...16})$

$c_{1,3}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0

$IV^{63...48} \lozenge IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{1,4}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0

$c_{2,4} = c_{2,3} \oplus IV^{63..32}$

$c_{2,3}$ = 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0

$IV^{[63..32]}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$c_{2,4}$ = 0 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0

$c_{3,4} = c_{3,3} \oplus (IV^{47...32} \Diamond IV^{15...0})$

$c_{3,3}$ = 0 1 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0

$IV^{47..32} \Diamond IV^{15...1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{3,4}$ = 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0

$c_{4,4} = c_{4,3} \oplus IV^{31...0}$

$c_{4,3}$ = 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{4,4}$ = 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0

$c_{5,4} = c_{5,3} \oplus (IV^{63...48} \Diamond IV^{31...16})$

$c_{5,3}$ = 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1

$IV^{63...48} \Diamond IV^{31...16}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$c_{5,4}$ = 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 1

$c_{6,4} = c_{6,3} \oplus IV^{31...0}$

$c_{6,3}$ = 0 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 0

$IV^{[31...0]}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

$c_{6,4}$ = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0

$c_{7,4} = c_{7,3} \oplus (IV^{47...32} \Diamond IV^{15...0})$

$c_{7,3}$ = 0 1 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0

$IV^{47...32} \Diamond IV^{15...0}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{7,4}$ = 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0

Sebelum menghitung fungsi *Next State* maka terlebih dahulu dahulu *System Counter* di diperbaharui. Sedangkan dinamika *system counter* dapat di definisikan sebagai berikut :

$$c_{0,i+1} = \begin{cases} c_{0,i} + a_0 + \phi_{7,i} \bmod 2^{32} & jika\ j = 0 \\ c_{j,i} + a_j + \phi_{j-1,i+1} \bmod 2^{32} & jika\ j > 0 \end{cases}$$

Dengan bit carry $\phi_{j,i+1}$ ditentukan dengan :

$$\phi_{j,i+1} = \begin{cases} 1 & jika\ c_{0,i} + a_0 + \phi_{7,i} \geq 2^{32} \wedge\ j = 0 \\ 1 & jika\ c_{0,i} + a_0 + \phi_{j-1,i+1} \geq 2^{32} \wedge\ j > 0 \\ 0 & Jikatidakkeduanya \end{cases}$$

Jika :

a_0 = 2468 = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0

a_1 = 2745 = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1

a_2 = 1257 = 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1

a_3 = 3254 = 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0

a_4 = 8163 = 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1

a_5 = 6587 = 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1

a_6 = 4365 = 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1

a_7 = 3658 = 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0

i=1; j=0

$c_{0,1+1} = c_{0,2}$

$c_{0,2} = c_{0,1} + a_0 + \phi_{7,1} \bmod 2^{32}$

$c_{0,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0

a_0 = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0

$\phi_{7,1}$ = 0 +

$c_{0,2}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 1 0 mod 2^{32}

$c_{0,2}$ = 549602446 mod 2^{32} = 549602446

$c_{0,2}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 1 0

$\phi_{j,i+1} = \phi_{0,2}$ = 0 di mana 549602446 < 2^{32}, j=0

i=1; j=1

$c_{1,1+1} = c_{1,2}$

$c_{1,2} = c_{1,1} + a_1 + \phi_{0,2} \bmod 2^{32}$

$c_{1,2}$ = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0

a_1 = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1

$\phi_{7,1}$ = 0 +

$c_{1,2}$ = 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 mod 2^{32}

$c_{1,2}$ = **67568647** mod 2^{32} = **67568647**

$c_{1,2}$ = 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1

$\phi_{j,i+1} = \phi_{1,2}$ = 0 di mana **67568647**< 2^{32}, j>0

i=1; j=2

$c_{2,1+1} = c_{2,2}$

$c_{2,2} = c_{2,1} + a_2 + \phi_{1,2} \bmod 2^{32}$

$c_{2,1}$ = 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0

a_2 = 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1

$\phi_{1,2}$ = 0 +

$c_{2,2}$ = 0 0 0 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 mod 2^{32}

$c_{2,2}$ = **469526457** mod 2^{32} = = **469526457**

$c_{2,2}$ = 0 0 0 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1

$\phi_{j,i+1} = \phi_{2,2}$ = 0 di mana **469526457** < 2^{32}, j>0

i=1; j=3

$c_{3,1+1} = c_{3,2}$

$c_{3,2} = c_{3,1} + a3 + \phi_{2,2} \bmod 2^{32}$

$c_{3,1}$ = 0 1 1 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0

a_3 = 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0

$\phi_{2,2}$ = 0 +

$c_{3,2}$ = 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 mod 2^{32}

$c_{3,2}$ = : **1772112772** mod 2^{32} = **1772112772**

$c_{3,2}$ = 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0

$\phi_{j,i+1} = \phi_{3,2}$ = 0 di mana **1772112772** < 2^{32}, j>0

i=1; j=4

$c_{4,1+1} = c_{4,2}$

$c_{4,2} = c_{4,1} + a_4 + \phi_{3,2} \bmod 2^{32}$

$c_{4,1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0

a_4 = 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1

$\phi_{3,2}$ = 0 +

$c_{4,2}$ = 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1 1 mod 2^{32}

$c_{4,2}$ = : **1755383523** mod 2^{32} = **1755383523**

$c_{4,2}$ = 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0

$\phi_{j,i+1} = \phi_{4,2}$ = 0 di mana **1755383523**< 2^{32}, j>0

i=1; j=5

$c_{5,1+1} = c_{5,2}$

$c_{5,2} = c_{5,1} + a_5 + \phi_{4,2} \bmod 2^{32}$

$c_{5,1}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0

a_5 = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1

$\phi_{4,2}$ = 0 +

$c_{5,2}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1 mod 2^{32}

$c_{5,2}$ = : **283455173** mod 2^{32} = **283455173**

$c_{5,2}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1

$\phi_{j,i+1} = \phi_{5,2}$ = 0 di mana **283455173** < 2^{32}, j>0

i=1; j=6

$c_{5,1+1} = c_{6,2}$

$c_{6,2} = c_{6,1} + a_6 + \phi_{5,2} \bmod 2^{32}$

$c_{6,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 0

a_6 = 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1

$\phi_{5,2}$ = 0 +

$c_{6,2}$ = 0 0 1 0 0 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 1 1 mod 2^{32}

$c_{6,2}$ = : **650986635** mod 2^{32} = **650986635**

$c_{6,2}$ = 0 0 1 0 0 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 1 1

$\phi_{j,i+1} = \phi_{6,2}$ = 0 di mana **650986635** < 2^{32}, j>0

i=1; j=7

$c_{7,1+1} = c_{7,2}$

$c_{7,2} = c_{7,1} + a_7 + \phi_{6,2} \bmod 2^{32}$

$c_{7,1}$ = 0 1 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0

a_7 = 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0

$\phi_{6,2}$ = 0 +

$c_{7,2}$ = 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 1 1 0 0 1 1 0 1 0 mod 2^{32}

$c_{7,2}$ = : **1840532378** mod 2^{32} = **1840532378**

$c_{7,2}$ = 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 1 1 0 0 1 1 0 1 0

$\phi_{j,i+1} = \phi_{7,2}$ = 0 di mana **1840532378** < 2^{32}, j>0

Fungsi utama dari algoritma Rabbit terletak pada persamaan berikut:

$$x_{j,i+1} = \begin{cases} g_{j,i} + (g_{j-1 \bmod 8,i} \lll 16) + (g_{j-2 \bmod 8,i} \lll 16) & jika\ j\ genap \\ g_{j,i} + (g_{j-1 \bmod 8,i} \lll 8) + (g_{j-2 \bmod 8,i}) & jika\ j\ ganjil \end{cases}$$

Dengan

$$g_{j,i} = \left(\left(x_{j,i} + c_{j,i} \right)^2 \oplus \left(\left(x_{j,i} + c_{j,i} \right)^2 \gg 32 \right) \right) mod\ 2^{32}$$

i=1, j=0

$g_{0,1} = \left(\left(x_{0,1} + c_{0,1}\right)^2 \oplus \left(\left(x_{0,1} + c_{0,1}\right)^2 >> 32\right)\right) \bmod 2^{23}$

$x_{0,1}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$c_{0,1}$ = 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0 +

$x_{0,1} + c_{0,1}$ =0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0

$\left(x_{0,1} + c_{0,1}\right)^2$ = $(33948166)^2$ = 1152477974763560 = 01000100101100010110101100010110

$\left(\left(x_{0,1} + c_{0,1}\right)^2 >> 32\right)$ = 1125466772230 = 01000011000101010100001010010100

$\left(\left(x_{0,1} + c_{0,1}\right)^2 \oplus \left(\left(x_{0,1} + c_{0,1}\right)^2 >> 32\right)\right)$ = 00000111101001000010100110000010

$\left(\left(x_{0,1} + c_{0,1}\right)^2 \oplus \left(\left(x_{0,1} + c_{0,1}\right)^2 >> 32\right)\right)$ = 128199042

$g_{0,1} = \left(\left(x_{0,1} + c_{0,1}\right)^2 \oplus \left(\left(x_{0,1} + c_{0,1}\right)^2 >> 32\right)\right) \bmod 2^{23}$

$= 128199042 \bmod 2^{23}$

$= 128199042$

= 00000111101001000010100110000010

i=1, j=1

$g_{1,1} = \left(\left(x_{1,1} + c_{1,1}\right)^2 \oplus \left(\left(x_{1,1} + c_{1,1}\right)^2 >> 32\right)\right) \bmod 2^{23}$

$x_{1,1}$ = 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0

$c_{1,1}$ = 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0 +

$x_{1,1} + c_{1,1}$ =0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0

$\left(x_{1,1} + c_{1,1}\right)^2$ = $(1716524566)^2$

= 2946456585681490000= 01000100101100010110101100010110

$\left(\left(x_{1,1} + c_{1,1}\right)^2 >> 32\right)$ = 2877399009454580 = 10101011100000011001111111100001

$\left(\left(x_{1,1} + c_{1,1}\right)^2 \oplus \left(\left(x_{1,1} + c_{1,1}\right)^2 >> 32\right)\right)$ = 11101111001100011111010011110111

$\left(\left(x_{1,1} + c_{1,1}\right)^2 \oplus \left(\left(x_{1,1} + c_{1,1}\right)^2 >> 32\right)\right)$ = 4013028599

$g_{1,1} = \left(\left(x_{1,1} + c_{1,1}\right)^2 \oplus \left(\left(x_{1,1} + c_{1,1}\right)^2 >> 32\right)\right) \bmod 2^{23}$

$= 4013028599 \bmod 2^{23}$

$= 4013028599$

= 11101111001100011111010011110111

Setelah selesai perhitungan di atas, maka diperoleh nilai $x_{i,j}$ adalah sebagai berikut :

$x_{0,1}$ = 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

$x_{1,1}$ = 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0

$x_{2,1}$ = 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0

$x_{3,1}$ = 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0

$x_{4,1}$ = 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0

$x_{5,1}$ = 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0

$x_{6,1}$ = 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0

$x_{7,1}$ = 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0

Setelah setiap iterasi, 4 rangkaian bit (*word*) acak semu sepanjang 32 bit dibangkitkan dengan aturan sebagai berikut:

$$s_{j,i}^{[15..0]} = x_{2j,i}^{[15..0]} \oplus x_{2j+3\ mod\ 8,i}^{[31..16]}$$

$$s_{j,i}^{[31..16]} = x_{2j,i}^{[31..16]} \oplus x_{2j+3\ mod\ 8,i}^{[15..0]}$$

i=1, j=0

$S_{0,1}^{[15..0]} = x_{0,1}^{[15..0]} \oplus x_{3,1}^{[31.16]}$

$=0010000011000000 \oplus 0000100011110000$

$= 0010100000110000$

$S_{0,1}^{[31..16]} = x_{0,1}^{[31..16]} \oplus x_{3,1}^{[15..0]}$

$=0100000010000000 \oplus 0100000010000000$

$= 0000000000000000$

Maka

$S_{0,1}$ = 0010100000110000 0000000000000000

i=1, j=1

$S_{0,1}^{[15..0]} = x_{2,1}^{[15..0]} \oplus x_{5,1}^{[31.16]}$

$=0001000011100000 \oplus 0010000011000000$

$=0011000000100000$

$S_{0,1}^{[31..16]} = x_{2,1}^{[31..16]} \oplus x_{5,1}^{[15..0]}$

$=0110000010100000 \oplus 0110000010100000$

$=0000000000000000$

$S_{1,1}$ = 0011000000100000 0000000000000000

i=1, j=2

$S_{2,1}^{[15..0]} = x_{4,1}^{[15..0]} \oplus x_{7,1}^{[31.16]}$

$= 0011000011010000 \oplus 0001000011100000$

$= 0010000000110000$

$S_{2,1}^{[31..16]} = x_{4,1}^{[31..16]} \oplus x_{7,1}^{[15..0]}$

$=0101000010010000 \oplus 0101000010010000$

$=0000000000000000$

$S_{2,1}$ = 0010000000110000 0000000000000000

i=1, j=3

$S_{3,1}^{[15..0]} = x_{6,1}^{[15..0]} \oplus x_{1,1}^{[31.16]}$

$=0000100011110000 \oplus 0011000011010000$

$=0011100000100000$

$S_{3,1}^{[31..16]} = x_{6,1}^{[31..16]} \oplus x_{1,1}^{[15..0]}$

$=0111000010110000 \oplus 0010000011000000$

$=0101000001110000$

$S_{2,1}$ = 0011100000100000 0101000001110000

Jika kunci (*key*) sebelumnya adalah '1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ', maka setelah melalui proses pembentukan kunci seperti yang diuraikan diatas, didapat 4 rangkaian bit acak semu sepanjang 32 bit yang nantinya akan digunakan untuk proses enkripsi dan dekripsi. Dimana 4 rangkaian bit acak semu tersebut adalah sebagai berikut :

$S_{j,i}$ = 0010100000110000 0000000000000000

0011000000100000000000000000000
0010000000110000000000000000000
0011100000100000010100000111000

Jika plainteks yang akan di enkripsi adalah sebagai berikut : P (*plaintext*) : STMIK BUDIDHARMA

Ubah plaintext ke bentuk biner :

‘S’ = 01010011
‘T’ = 01010100
‘M’ = 01001101
‘I’ = 01001001
‘K’ = 01001011
‘ ’= 00100000
‘B’ = 01000010
‘U’ = 01010101
‘D’ = 01000100
‘I’ = 01001001
‘D’ = 01000100
‘H’ = 01001000
‘A’ = 01000001
‘R’ = 01010010
‘M’ = 01001101
‘A’ = 01000001

hasil konversi plaintext ke biner :

P : 01010011 01010100 01001101 01001001 01001011 00100000
01000010 01010101 01000100 01001001 01000100 01001000
01000001 01010010 01001101 01000001

Selanjutnya lakukan XOR antara plainteks dengan nilai $S_{i,i}$, di mana jika panjang plainteks sama dengan $S_{i,i}$, maka XOR setiap bit plainteks dengan bit $S_{i,i}$, namun jika plainteks lebih panjang maka XOR bit plainteks dengan mengulang ke bit awal $S_{i,i}$, sedangkan jika plainteks lebih pendek maka XOR awal plainteks kembali sesuai dengan panjang $S_{j,i}$.

Plainteks : 01010011 01010100 01001101 01001001 01001011 00100000
01000010 01010101 01000100 01001001 01000100 01001000
01000001 01010010 01001101 01000001

$S_{j,I}$: 0010100000110000000000000000000
0011000000100000000000000000000
0010000000110000000000000000000
0011 1000001000000101 0 00001 110000

Maka hasil enkripsi (*cipherteks)* adalah sebagai berikut : Cipherteks : 123 100 77 73 123 64 66 85 116 120 68 72 121 114 29 49 { d M I { @ B U t x D H y r GS 1.

4. KESIMPULAN

Adapun kesimpulan yang penulis peroleh berdasarkan hasil perancangan aplikasi ini adalah sebagai berikut :

a. Aplikasi ini dapat digunakan untuk mengenkripsi data teks hingga ukuran 224 byte, sehingga efektif digunakan untuk enkripsi data berukuran besar.
b. Kecepatan sistem dalam melakukan enkripsi dan dekripsi dapat dikatakan sangat cepat sehingga dapat menghemat waktu dalam melakukan enkripsi data teks.
c. Sistem belum memiliki fasilitas untuk penyimpanan data hasil enkripsi dan dekripsi, sehingga data hasil enkripsi tidak bisa digunakan setiap saat dibutuhkan, serta belum bisa dijalankan dalam jaringan apapun.

DAFTAR PUSTAKA

[1] Adelia. 2004. Dasar-Dasar Pemrograman Microsoft Visual Basic 2008. Bandung. Penerbit PT. Sarana Tutorial Nurani Sejahtera.
[2] Sholiq. 2006. Panduan Singkat Bahasa Pemodelan Objek Standar. Yogyakarta. Penerbit Andi.
[3] Kendall. 2007. Analisis dan Perancangan Sistem. Jakarta. Penerbit Indeks.
[4] Nurdin Usman. 2002. Konteks Implementasi Berbasis Kurikulum. Jakarta. Penerbit PT. Indeks.
[5] Rifki Sadikin. 2012. Kriptografi Untuk Keamanan Jaringan.Yogyakarta. Penerbit Andi publisher.
[6] Andree Datta Adwitya. 2006. Studi Lengkap Mengenai Rabbit Cipher Institut Teknologi Bandung. Waktu Akses 30 Juni2014, 16:12 WIB.
[7] Mohamad Endhy. 2008. Implementasi Algoritma Stream Cipher Rabbit Pada Protokol Secure Socket Layer (SSL)Universitas Indonesia. Waktu Akses 26 Juni 2014, 21:15 WIB.
[8] Paramitha. 2006. Studi dan Analisis Mengenai Algoritma Cipher Aliran "Rabbit" STEI-ITB. Waktu Akses 26 Juni 2014, 23:18 WIB.
[9] Rinaldi Munir. 2003. Aplikasi Klien Surel Dengan Algoritma Rabbit Pada Ponsel Android Teknik Informatika ITB. Waktu Akses 26 Juni 2014, 23:40 WIB.