KRIPTOGRAFI FILE CITRA MENGGUNAKAN ALGORITMA *TEA (TINY ENCRYPTION ALGORITHM)*

Mukti Qamal*

Abstract

The sharp growth of data communication recently raise the importance of data security and confidentiality. Cryptography is the art or science of maintaining the security of data or message that randomize the data or message. In addition to text-based documents, it is also common to send image through a network. This research investigates the way to secure the image from the cryptanalyst disorders by using TEA (Tiny Encryption Algorithm) algorith, which is a secret key cryptographic algorithm. The strength of this algorithm lies in the Feistel network and the number of delta derived from the golden number. It was found that the algorithm is appropriate for encrypting and decrypting image, because of stronger round quantity and key length, and does not need S-box and P-box in the process. The test showed that this algorithm with 32 round fits highly for securing images with optimum speed. It was also found that the bigger image size, the longer the encryption and decryption process takes time to run, with decrypted image has different bit depth value compared to original image.

Keywords : Image , Cryptography , TEA (Tiny Encryption Algorithm)

PENDAHULUAN

Masalah keamanan dan kerahasiaan data merupakan hal yang sangat penting dalam suatu organisasi. Terlebih jika arus data tersebut berada dalam suatu jaringan komputer yang terhubung ke jaringan publik atau internet. Banyak orang kemudian berusaha menyiasati bagaimana mengamankan informasi yang dikomunikasikannya, atau menyiasati bagaimana mendeteksi keaslian dari informasi yang diterimanya.

* Dosen Teknik Informatika Universitas Malikussaleh.

Citra digital telah digunakan secara luas dalam berbagai macam proses sehingga perlindungan citra digital dari pihak yang tidak memilki hak akses menjadi sangat penting. Banyak institusi telah menggunakan citra digital untuk menyimpan informasi penting, misalnya hasil pemeriksaan pasien (untuk rumah sakit), area geografi (untuk penelitian), posisi musuh (untuk militer), produk baru (untuk perusahaan swasta), dan status keuangan.

Dalam penelitian ini teknik yang akan digunakan untuk mengkriptografi file citra adalah algoritma Tiny Encription Algoritm. Algoritma ini merupakan algoritma penyandian *block cipher* yang dirancang untuk penggunaan memory yang seminimal mungkin dengan kecepatan proses yang maksimal. TEA diklaim mempunyai tingkat keamanan yang tinggi jika dibandingkan dengan algoritma kriptografi sejenis. Keunggulan utama dari TEA adalah keringanan prosesnya, operasi-operasi yang digunakan hanya berupa operasi bit biasa, tanpa substitusi, permutasi ataupun operasimatrik.

KEAMANAN DATA

Ada beberapa ancaman keamanan yang sering terjadi terhadap informasi yaitu :

1. *Interuption*: merupakan ancaman terhadap *availability* informasi, data yang ada dalam sistem komputer dirusak atau dihapus sehingga jika data atau informasi tersebut dibutuhkan maka pemiliknya akan mengalami kesulitan untuk mengaksesnya, bahkan mungkin informasi itu hilang.
2. *Interception*: merupakan ancaman terhadap kerahasiaan *(secrey)*. Informasi disadap sehingga orang yang tidak berhak dapat mengakses komputer dimana informasi tersebut disimpan.
3. *Modification*: merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu-lintas informasi yang sedang dikirim dan kemudian mengubahnya sesuai dengan keinginan orang tersebut.

4. *Fabrication*: merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru atau memalsukan informasi sehingga orang yang menerima informsi tersebut menyangka bahwa informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi.

Tabel 1. Ancaman Terhadap Komputer

System	Avability	Secrey	Integrity
Hardware	Dicuri atau dirusak		
Software	Program dihapus	Software di copy	Program dimodifikasi
Data	File dihapus atau dirusak	Dicuri, disadap	File dimodifikasi
Line komunikasi	Kabel diputus	Informasi disadap	Informasi dimodifikasi

Keamanan komputer meliputi beberapa aspek, antara lain :

1. *Authentication:* agar penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi.
2. *Integrity:* keaslian pesan yang dikirim melalui jaringan dan dapat dipastikan bahwa informasi yang dikirim tidak dimodifikasi oleh orang yang tidak berhak.
3. *Non-repudiation:* hal yang berhubungan dengan si pengirim. Pengirim tidak dapat mengelak bahwa dialah yang mengirim informasi tersebut.
4. *Authority:* Informasi yang berada pada sistem jaringan tidak dapat dimodifikasi oleh pihak yang tidak berhak untuk mengaksesnya.
5. *Confidentiality:* merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses.
6. *Privacy:* lebih ke arah data-data yang bersifat pribadi.
7. *Availability:* berhubungan dengan ketersediaan informasi ketika dibutuhkan.
8. *Access Control:* Aspek ini berhubungan dengan cara pengaturan akses ke informasi. Hal ini biasanya berhubungan dengan masalah

otentikasi dan privasi. Kontrol akses seringkali dilakukan dengan menggunakan kombinasi *user id* dan *password.*

DEFINISI KRIPTOGRAFI

Menurut Dony Arius (2008) Kriptografi berasal dari bahasa Yunani, *crypto*dan *graphia. Crypto*berarti *secret*(rahasia) dan graphia berarti *writing*(tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk mejaga keamanan peasn ketika pesan dikirim dari suatu tempat ke tempat lain .

Menurut Yusuf Kurniawan (2004) Kriptografi adalah ilmu yang mempelajari bagaimana suatu pesan atau dokumen kita aman, tidak bisa di baca oleh pihak yang tidak berhak.

Menurut Rinaldi Munir (2006) Kriptografi *(cryptography)* berasal dari bahasa Yunani: *"cryptos"* artinya *"secret"* (rahasia), sedangkan *"graphein"* artinya *"writing"* (tulisan rahasia). Ada beberapa definis Kriptografi yang telah dikemukakan di dalam berbagai literatur. Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikan ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definsi ini mungkin cocok pada masa lalu dimana kriptografi digunakan untuk keamanan komunikasi penting seperti dikomunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih dari sekedar *privacy,* tetapi juga untuk tujuan *data integrity, authentication, non-repudiation.*

TUJUAN KRIPTOGRAFI

Kriptografi bertujuan untuk memberi layanan keamanan (yang juga dinamakan aspek-aspek keamanan) sebagai berikut (Munir, Rinaldi 2006) :

- **Kerahasiaan** *(confidentiality)***,** adalah layanan yang ditunjukkan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
- **Integritas data** *(data integrity)* adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.

- **Otentikasi** *(authentication),* adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi *(user authentication* atau *entity authentication)* maupun mengidentifikasi kebenaran sumber pesan *(data origin authentication).*
- **Nirpenyangkalan** *(non-repudiation),* adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

KOMPONEN KRIPTOGRAFI

Pada dasarnya komponen kriptografi terdiri dari beberapa komponen, seperti :

1. *Enkripsi:* merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Pesan asli disebut *plaintext* (teks-biasa), yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode.
2. *Dekripsi:* merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.
3. *Kunci:* adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, yaitu kunci rahasia *(private key)* dan kunci umum *(public key).*
4. *Ciphertext:* merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak bisa dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti).
5. *Plaintext:* sering disebut dengan *cleartext.* Teks-asli atau teks-biasa ini merupakan peasn yang ditulis atau diketik yang memilki makna. Teks-asli inilah yang diproses menggunakan algoritma kriptografi untuk menjadi ciphertext (teks-kode).

6. *Pesan:* dapat berupa data atau informasi yang dikrim (melalui kurir, saluran komunikasi data, dsb) atau yang disimpan di dalam media perekaman (kertas, storage, dsb).
7. *Cryptanalysis:* Kriptanalisis bisa diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan teks-asli tanpa haru mengetahui kunci yang sah secara wajar. Jika suatu teks-kode berhasil diubah menjadi teks-asli tanpa menggunakan kunci yang sah, proses tersebut dinakaman *breaking code.* Hal ini dilakukan oleh para kriptanalisis. Analisis kode juga dapat menemukan kelemahan dari suatu algoritma kriptografi dan akhirnya dapat menemukan kunci atau teks-asli dari teks-kode yang dienkripsi dengan algoritma teretentu.

Konsep matematis yang mendasari algortima kriptografi adalah relasi antara dua buah himpunan, yaitu himpunan yang berisi elemen–elemen *plaintext* dan himpunan yang berisi *ciphertext*.Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen– elemen antara kedua himpunan tersebut. Misalnya P menyatakan plaintext dan C menyatakan ciphertext, maka fungsi enkripsi E memetakan P ke C :

$$E(P) = C$$

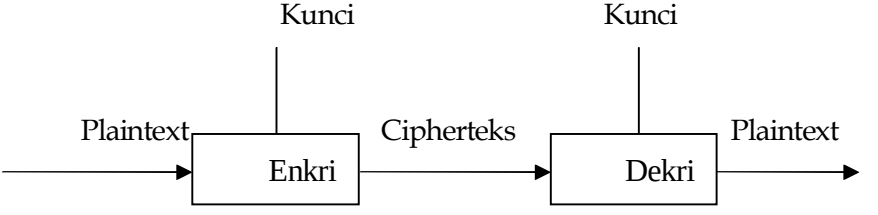
Dan fungsi dekripsi D memetakan C ke P :

$$D(C) = P$$

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan awal, maka kesamaan berikut harus benar :

$$D(E(P)) = P$$

Keamanan algoritma kriptografi sering diukur dari banyaknya kerja (*work*) yang dibutuhkan untuk memecahkan ciphertext menjadi plaintext-nya tanpa mengetahui kunci yang digunakan.



Gambar 1. Skema Enkripsi dan Dekripsi

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya:

1) Algoritma Simetri (menggunakan satu kunci untuk enkripsi dan dekripsinya)
2) Algoritma Asimetri (menggunakan kunci yang berbeda untuk enkripsi dan dekripsi)
3) *Hash Function*

ALGORITMA SIMETRI

Algoritma ini memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahukan kunci dari pesan tersebut agar bisa mendekripsikan pesan yang dikirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan. Algoritma yang memakai kunci simetri di atntaranya adalah :

1. Data Encryption Standard (DES),
2. RC2, RC4, RC5, RC6,
3. International Data Encryption Algorithm (IDEA),
4. One Time Pad (OTP),
5. A5,
6. Tiny Encryption Algorithm (TEA), dan lain sebagainya.

Kelebihan Kriptografi Simetri:

1. Algoritma kriptografi simetri dirancang sehingga proses enkripsi/dekripsi membutuhkan waktu yang singkat.
2. Ukuran kunci simetri relatif pendek.
3. Algoritma kriptografi simetri dapat digunakan untuk membangkitkan bilangan acak.
4. Algoritma kriptografi simetri dapat disusun untuk menghasilksn chiper yang lebih kuat.
5. Otentikasi pengirim pesan langsung diketahui dari chiperteks yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja.

Kelemahan Kriptografi Simetri:

1. Kunci simetri harus dikirim melalui saluran yang aman. Kedua entitas yang berkominikasi harus menjaga kerahasiaan kunci ini.'
2. Kunci harus sering diubah, mungkin pada setiap sesi komunikasi.

KATEGORI *CIPHER* KUNCI SIMETRI

Algoritma kunci-simetri mengacu pada metode enkripsi yang dalam hal ini baik pengirim maupun penerima memiliki kunci yang sama. Algoritma kunci-simetri modern beroperasi dalam mode bit dan dapat dikelompokkan menjadi dua kategori, yaitu:

1. Cipher Aliran (*Stream Cipher*)
   Algoritma kriptografi beroperasi dalam plaintexs/ciphertext dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsi/didekripsikan bit per bit. Chiper aliran mengenkripsi satu bit pada setiap kali proses enkripsi.
2. Cipher Blok (*Block Cipher*)
   Algoritma kriptografi beroperasi pada plaintexs/ciphertext dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi dalam blok-blok bit yang panjangnya sudah ditentukan sebelumnya. Misalnya panjang blok adalah 64 bit, maka itu berarti algoritma enkripsi memperlakukan 8 karakter setiap kali enkripsi (1 karakter = 8 bit dalam pengkodean ASCII). Cipher blok mengenkripsi satu blok bit setiap kali. (Munir, 2006).

MODE OPERASI CIPHER BLOK

Plaintext dibagi menjadi beberapa blok dengan panjang tetap. Beberapa mode operasi dapat diterapkan untuk melakukan enkripsi terhadap keseluruhan blok plaintext.

Empat mode operasi yang lazim diterapkan pada sistem blok chiper adalah :

1) Electronik Code Book (ECB)

Pada mode ini, setiap blok plainteks P_i dienkripsi secara individual dan independen menjadi blok cipherteks C_i .

2) *Cipher Blok Chaining (CBC)*

Mode ini menerapkan mekanisme umpan-balik *(feedback)*pada sebuah blok, yang dalam hal ini hasil enkripsi sebelumnya di-umpan-balikkan ke dalam enkripsi blok yang *current.* Caranya, blok plainteks yang *current* di-*XOR*-kan terlebih dahulu dengan blok cipherteks hasil enkripsi sebelumnya, selanjutnya hasil peng-*XOR*-an ini masuk ke dalam fungsi enkripsi. Dengan mode *CBC,* setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya.

Dekripsi dilakukan dengan memasukkan blok cipherteksnya yang *current* ke fungsi sebelumnya. Dalam hal ini, blok cipherteks sebelumnya berfungsi sebagai umpan-maju (*feedforward*) pada akhir proses dekripsi.

3) *Cipher-Feedback (CFB)*

Jika mode *CFB* yang diterapkan untuk transmisi data, maka enkripsi tidak dapat dilakukan bila blok plainteks yang diterima belum lengkap.

Secara umum, *CFB p*-bit mengengkripsi plainteks sebanyak p bit setiap kalinya, yang dalam hal ini mana $p \leq n$ (n = ukuran blok). Dengan kata lain, *CFB* mengenkripsi *cipher* blok seperti pada *cipher* aliran.

4) *Output-Feedback (OFB)*

Mode *OFB* ini mirip dengan mode *CFB*, kecuali *p*-bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan di antrian. Dekripsi dilakukan sebagai kebalikan dari proses enkripsi.

PRINSIP-PRINSIP PERANCANGAN *CIPHER* BLOK

Perancangan algoritma kriptografi yang berbasis blok mempertimbangkan beberapa prinsip berikut :

1. Prinsip *Confusion* dan *Diffusion* dari Shannon
   - *Confusion*

     Prinsip ini menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci. Prinsip *Confusion* akan membuat kriptanalis frustasi untuk mencari pola-pola statistik yang muncul cipherteks. *Confusion* yang bagus membuat

hubungan statistik antara plainteks, chiperteks, dan kunci menjadi sangat rumit.

- *Diffusion*
  Prinsip ini menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin cipherteks. Prinsip *Diffusion* juga menyembunyikan hubungan statistik antara plainteks, cipherteks, dan kunci dan membuat kriptanalisis menjadi sulit.

2. *Chiper* berulang (*Iteratid Cipher)*
   Fungsi transformasi sederhana yang mengubah plainteks menjadi cipherteks diulang sejumlah kali. Pada setiap putaran digunakan upa-kunci (*subkey*) atau kunci putaran (*round key*) yang dikombinasikan dengan plainteks.
3. Jaringan Feistel (*Feistel Network*)
   Hampir semua algoritma *chiper* blok bekerja dalam model jaringan Feistel. Karena model Jaringan Feistel bersifat *reverssible* untuk proses dekripsi dan enkripsi. Sifat *reverssible*ini membuat tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks. Jaringan Feistel ditemukan oleh Horst Feistel tahun 1970.
4. Kunci Lemah
   Kunci lemah adalah kunci yang menyebabkan tidak adanya perbedaan antara enkripsi dan dekripsi. Dekripsi terhadap cipherteks tetap menghasilkan plainteks semula, namun enkripsi dua kali berturut-turut terhadap plainteks akan menghasilkan kembali plainteksnya.
5. Kotak-S
   Kotak-S adalah matriks yang berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain. Pada kebanyakan algoritma *cipher* blok, kotak-S memetakan *m* bit masukan menjadi *n* bit keluaran, sehingga kotak-S tersebut dinamakan kotak *m x n* S-box.

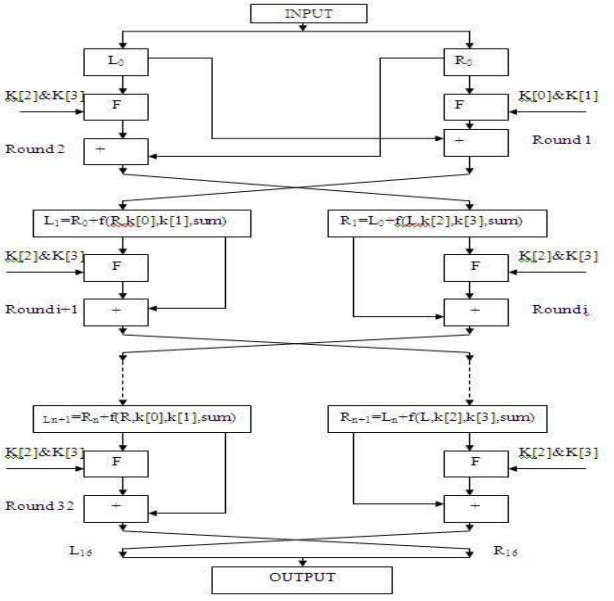
TINY ENCRYPTION ALGORITHM (TEA)

Tiny Encryption Algorithm (TEA) merupakan suatu algoritma sandi yang diciptakan oleh David J. Wheeler dan Roger M. Needham dari Cambridge University tahun 1994. Algoritma ini merupakan algoritma penyandian *block cipher* yang dirancang untuk penggunaan memori yang seminimal mungkin dengan kecepatan proses yang maksimal.

Sistem penyandian TEA menggunakan proses *feistel network* dengan menambahkan fungsi matematik berupa penambahan dan pengurangan sebagai operator pembalik selain XOR. Hal ini dimaksudkan untuk menciptakan sifat *non-linearitas*. Pergeseran dua arah (ke kiri dan ke kanan) menyebabkan semua bit kunci dan data bercampur secara berulang ulang.

TEA memproses 64-bit input sekali waktu dan menghasilkan 64-bit output. TEA menyimpan 64-bit input kedalam L_0 dan R_0 masing masing 32-bit, sedangkan 128-bit kunci disimpan kedalam k[0], k[1], k[2], dan k[3] yang masing masing berisi 32-bit. Diharapkan teknik ini cukup dapat mencegah penggunaan teknik *exshautive search* secara efektif. Hasil outputnya akan disimpan dalam L_{16} dan R_{16}.

Bilangan delta konstan yang digunakan adalah 9E3779B9, dimana bilangan delta berasal dari *golden number*, digunakan delta = $(\sqrt{5} -1)2^{31}$. Suatu bilangan delta ganda yang berbeda digunakan dalam setiap roundnya sehingga tidak ada bit dari perkalian yang tidak berubah secara teratur. Berbeda dengan struktur *feistel* yang semula hanya mengoperasikan satu sisi yaitu sisi sebelah kanan dengan sebuah fungsi F, pada algoritma TEA kedua sisi dioperasikan dengan sebuah fungsi yang sama. Struktur penyandian TEA adalah seperti dalam gambar berikut.

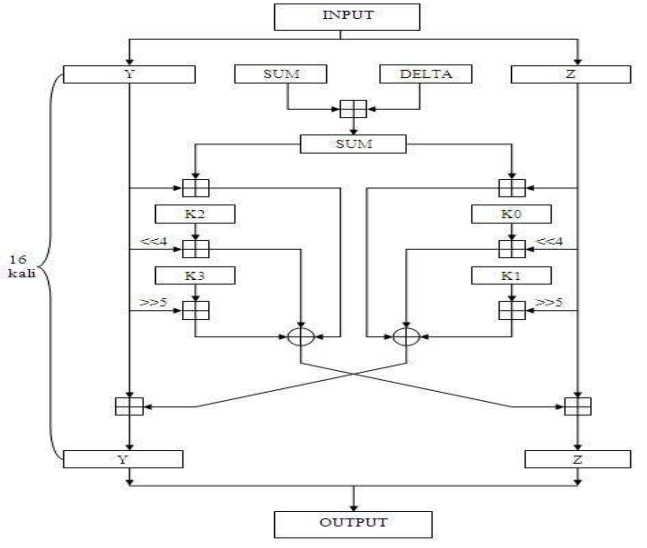


Gambar 2. Algoritma TEA

Untuk melakukan enkripsi, Proses diawali dengan *input-bit* teks sebanyak 64-bit, kemudian 64-bit teks tersebut dibagi menjadi dua bagian, yaitu sisi kiri (L_0) sebanyak 32-bit dan sisi kanan (R_0) sebanyak 32-bit. Setiap bagian teks akan dioperasikan sendiri-sendiri. R_0 (Z) akan digeser kekiri sebanyak empat (4) kali dan ditambahkan dengan kunci k[0], sementara itu Z ditambah dengan sum (delta) yang merupakan konstanta. Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Langkah selanjutnya di-XOR-kan dengan hasil penambahan antara Z yang digeser kekanan sebanyak lima (5) kali dengan kunci k[1]. Hasil tersebut kemudian ditambahkan dengan L_0 (Y) yang akan menjadi R_1.

Sisi sebelah kiri akan mengalami proses yang sama dengan sisi sebelah kanan. L_0 (Y) akan digeser kekiri sebanyak empat (4) kali lalu ditambahkan dengan kunci k[2], sementara itu, Y ditambah dengan sum (delta). Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Langkah selanjutnya di-XOR-kan dengan hasil penambahan antara Y yang digeser ke kanan sebanyak lima (5) kali dengan unci k[3]. Hasil tersebut kemudian ditambahkan dengan R_0 (Z) yang akan menjadi L_1.

Struktur dari penyandian dengan algoritma untuk satu cycle (dua round) dapat dilihat pada gambar 2.3 berikut:



Gambar 3. Satu Cycle TEA (dua round)

Langkah-langkah penyandian dengan algoritma TEA dalam satu cycle (dua round) :

1. Pergeseran *(shift)*
   Blok teks terang pada kedua sisi yang masing masing sebanyak 32-bit akan digeser kekiri sebanyak empat (4) kali dan digeser ke kanan sebanyak lima (5) kali.

2. Penambahan
   Langkah selanjutnya setelah digeser kekiri dan kekanan, maka Y dan Z yang telah digeser akan ditambahkan dengan kunci k[0]-k[3]. Sedangkan Y dan Z awal akan ditambahkan dengan sum (delta).

3. Peng-XOR-an
   Proses selanjutnya setelah dioperasikan dengan penambahan pada masing-masing register maka akan dilakukan peng-XOR-an dengan rumus untuk satu *round* adalah sebagai berikut:

```
y = y + (((z<<4)+k[0])^z+sum^((z>>5)+k[1]))
z = z + (((y<<4)+k[2]^y+sum^((y>>5)+k[3]))
```

Hasil penyandian dalam satu *cycle* satu blok teks terang 64-bit menjadi 64-bit teks sandi adalah dengan menggabungkan Y dan Z. Untuk penyandian pada *cycle* berikutnya Y dan Z ditukar posisinya, sehingga Y_1 menjadi Z_1 dan Z_1 menjadi Y_1 lalu dilanjutkan proses seperti langkah-langkah diatas sampai dengan *16 cycle (32 round).*

4. Key Schedule
   Algoritma TEA menggunakan *key schedule*-nya sangat sederhana. Yaitu kunci k[0] dan k[1] konstan digunakan untuk *round*ganjil sedangkan kunci k[2] dan k[3] konstan digunakan untuk *round* genap.

5. Dekripsi dan Enkripsi
   Proses dekripsi sama halnya seperti pada proses penyandian yang berbasis *feistel cip*her lainnya. Yaitu pada prinsipnya adalah sama pada saat proses enkripsi. Hal yang berbeda adalah penggunaan teks sandi sebagai *input* dan kunci yang digunakan urutannya dibalik. Proses dekripsi semua *round* ganjil menggunakan k[1] terlebih dahulu kemudian k[0], demikian juga dengan semua *round* genap digunakan k[3] terlebih dahulu kemudian k[2]. Rumus untuk enkripsi dan dekripsi seperti dibawah ini :

   Rumus Enkripsi:

   SUM = n * Delta

   $L_n = L_{(n-1)}+((R_{(n-1)}\text{Shl } 4)+K[0])\ \text{XOR}\ (R_{(n-1)}+\text{SUM})\ \text{XOR}\ ((R_{(n-1)}\ \text{Shr } 5)+K[1]))$

   $R_n = R_{(n-1)} + ((L_n\ \text{Shl } 4)+K[2])\ \text{XOR}\ (L_n+\text{SUM})\ \text{XOR}\ ((L_n\ \text{Shr } 5)+K[3])$

   Rumus Dekripsi:

   $R_n = R_{(n-1)} - ((L_{(n-1)}\ \text{Shl } 4)+k[2])\ \text{XOR}\ (L_{(n-1)}+\text{SUM})\ \text{XOR}\ ((L_{(n-1)}\ \text{Shr } 5)+k[3])$

   $L_n = L_{(n-1)} - ((R_n\ \text{Shl } 4)+k[0])\ \text{XOR}\ (R_n+\text{SUM})\ \text{XOR}\ ((R_n\ \text{Shr } 5)+k[1])\ )$

   Di mana : Delta = $(\sqrt{5}-1)2^{31} = (2654435770)_D = (\$9e3779b9)H$

   n = *Round* ke

Adapun beberapa keunggulan *Tiny Encryption Algorithm (TEA)* yaitu :

a. Pada Algoritma *Tiny Encryption Algorithm (TEA)* panjang kuncinya yaitu 128-bit, merupakan jumlah kunci yang cukup panjang untuk

algoritma kriptografi modern saat ini yang dapat menahan serangan kriptanalis.

b. Teknik yang digunakan TEA cukup baik, yaitu pada setiap prosesnya menggunakan jaringan feistel yang memuat operasi permutasi, substitusi dan modular arithmatic berupa XOR dan penambahan bilangan delta yang diharapkan dari operasi tersebut menciptakan efek difusi dan konfusi yang baik, karena semakin baik efek difusi dan konfusi yang dihasilkan suatu algoritma semakin baik pula tingkat keamanannya.
c. Ukuran blok input pada TEA yaitu 64-bit, sebuah jumlah yang cukup panjang untuk menghindari analisis pemecahan kode dan cukup kecil agar dapat bekerja dengan cepat.
d. Tidak membutuhkan S-Box dan P-Box dalam proses enkripsi dan dekripsinya, karena S-Box dan P-Box tersebut tidak dapat menjamin keamanannya dikarenakan sruktur dariS-Box dan P-Box tersebut hanya diketahui oleh NSA (National Security Agency) dan diubah menurut saran dari NSA, sehingga jika S-Box dan P-Box tersebut diubah maka sangat mungkin sekali algoritma yang digunakan akan lebih mudah dibobol. Selain itu, juga dapat meminimalkan penggunaan memory pada saat melakukan proses enkripsi dan dekripsi sehingga dapat memaksimlakan proses.
e. Algoritma TEA diketahui sangat kuat terhadap metode penyerangan berupa hanya chipertext yang diketahui, plaintext yang diketahui dan plaintext yang terpilih.

Sedangkan kelemahan dari Algoritma *TEA* ini adalah karena TEA ini termasuk ke dalam kelompok Algoritma Simetri, maka masih rentan untuk dibobol, karena dalam algoritma simetri masalah utama memang terletak dari segi pendistribusikan kunci yang akan digunakan.

FUNGSI HASH SHA-1

SHA-1 menerima masukan berupa pesan dengan ukuran maksimum 2^{64} bit (2. 147 .483 .648 *gigabyte*) dan menghasilkan *message digest* yang panjangnya 160 bit. Langkah-langkah pembuatan *message digest* dengan SHA-1 secara garis besar adalah sebagai berikut :

1. Penambahan Bit-bit Pengganjal

Penambahan bit-bit pengganjal (padding bits). Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512. Ini berarti panjang pesan setelah ditambahi bit-bit pengganjal adalah 64 bit kurang dari kelipatan 512. Panjang bit-bit pengganjal haruslah berada antara 1 hingga 512 bit. Hal tersebut menyebabkan pesan dengan panjang 448 tetap harus ditambahkan bit penyangga sehingga panjangnya akan menjadi 960 bit. Bit-bit pengganjal sendiri terdiri dari sebuah bit 1 diikuti dengan bit 0.

2. Penambahan Nilai Panjang Pesan Semula

Penambahan nilai panjang pesan semula. Pesan yang telah diberi padding bits selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Setelah ditambah dengan 64 bit, panjang pesan akan menjadi kelipatan 512 bit.

3. Inisialisasi Penyangga MD

SHA membutuhkan 5 buah penyangga (*buffer*) yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 5 x 32 = 160 bit. Kelima penyangga ini menampung hasil antara dan hasil akhir. Kelima penyangga tersebut diberi nama A, B, C, D, dan E. Setiap penyannga diinisialisasi dengan nilai-nilai (dalam notasi *HEX*) sebagai berikut :

A=67452301
B=EFCDAB89
C=98BADCFE
D=10325476
E = C3D2E1F0

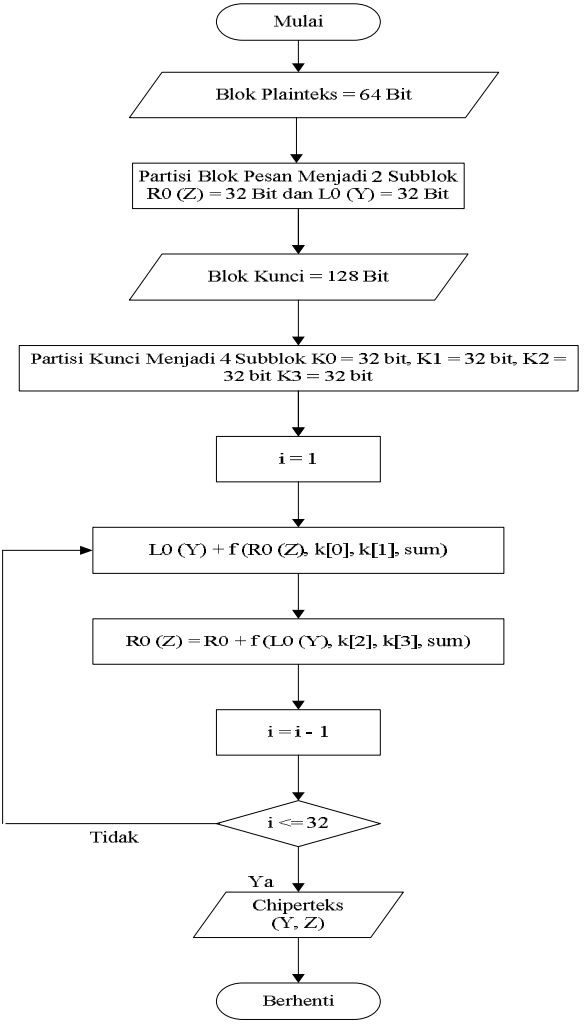
4. Pengolahan Pesan dalam Blok Berukuran 512 bit

Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit. Setiap blok 512-bit diproses bersama dengan buffer MD menjadi keluaran 128-bit. Proses ini disebut proses H_{SHA}.

SKEMA SISTEM

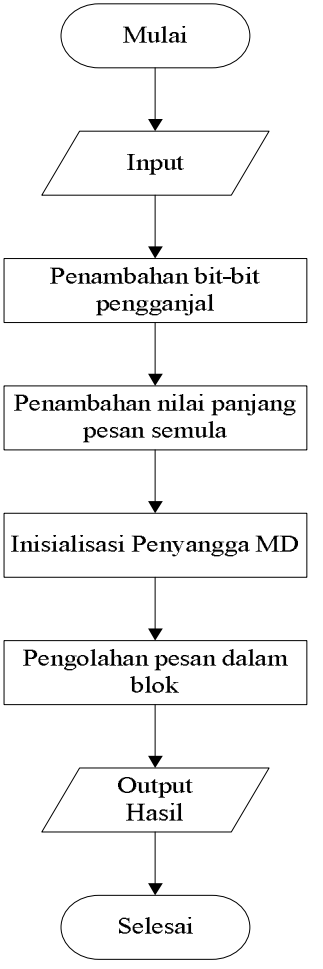
Diagram alir atau skema sistem kriptografi citra menggunakan algoritma TEA secara umum digambarkan sebagai berikut :

1. Skema Enkripsi Data



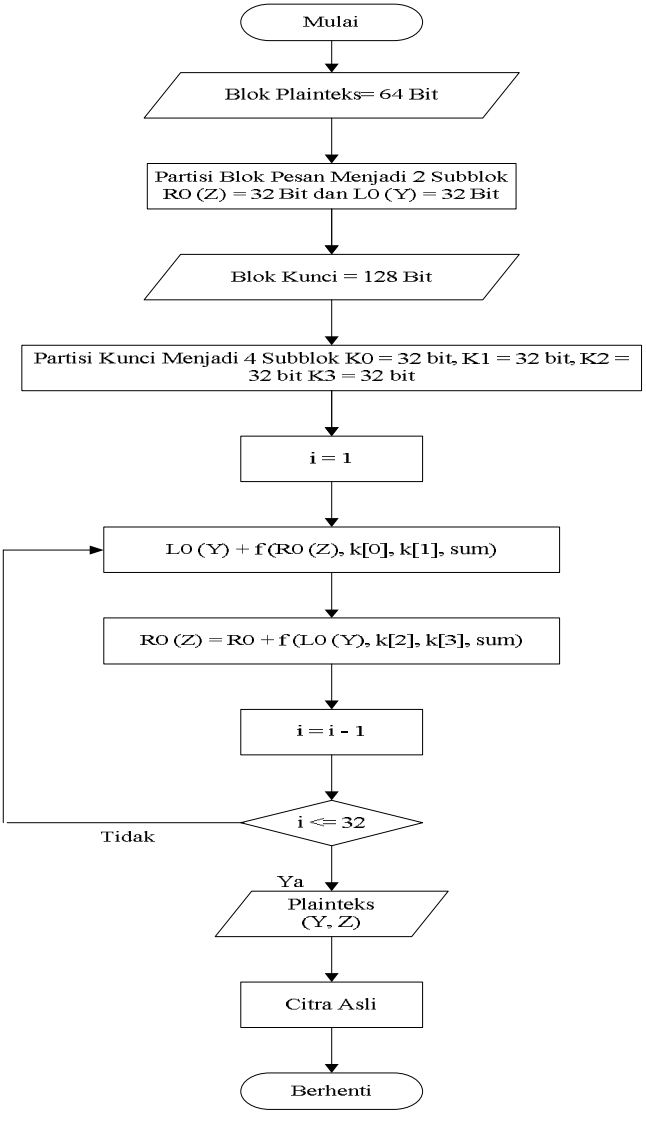
Gambar 4. Skema Enkripsi

2. Skema Fugsi Hash SHA-1



Gambar 5. Skema Fungsi Hash SHA-1

3. Skema Dekripsi Data



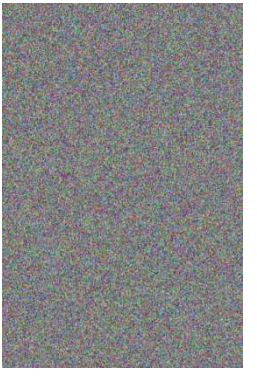
Gambar 6. Skema Dekripsi

PENGUJIAN SISTEM KRIPTOGRAFI CITRA

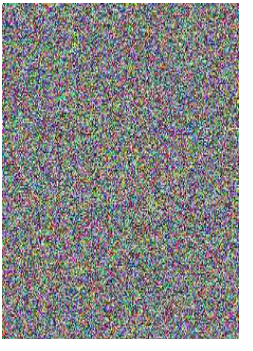
Citra untuk pengujian terdiri dari citra standar berupa file yang berekstensi bmp *(bitmap)*. Format JPEG tidak didukung, karena penyimpanan dengan format JPEG dapat menyebabkan kerusakan pada

data yang disembunyikan dan pada Delphi 7 gambar yang berformat JPEG kurang mendukung.

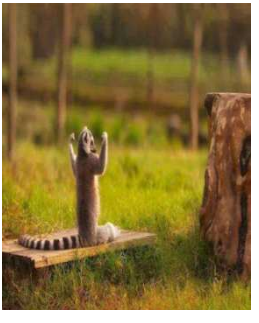
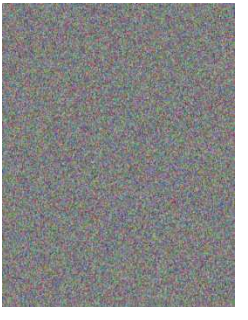
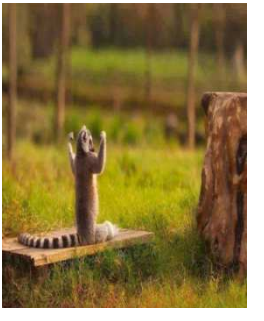
Pada bagian ini akan dilakukan pengujian pada 10 citra dengan kunci yang berbeda. Di bawah ini adalah data citra uji coba.


Citra Asli Hasil Enkripsi Hasil Dekripsi


Citra Asli Hasil Enkripsi Hasil Dekripsi

Citra Asli Hasil Enkripsi Hasil Dekripsi

Citra Asli Hasil Enkripsi Hasil Dekripsi

Citra Asli Hasil Enkripsi Hasil Dekripsi

Citra Asli Hasil Enskripsi Hasil Deskripsi

Gambar 7. Citra Yang Digunakan Dalam Uji Coba

Berdasarkan hasil pengujian di atas, dapat terlihat bahwa algoritma *TEA (Tiny Encryption Algorithm)* dapat bekerja dengan baik dalam proses enkripsi maupun dekripsinya. Semua citra yang diuji berhasil dienkripsi dan juga berhasil didekripsikan.

Tabel 2. Citra Asli

Nama File Citra	Pixel Asli	Bit Depth Asli
Dian Pelangi.*bmp*	427x640	24
Dewi Sandra.*bmp*	224x225	24
Hijau.*bmp*	650x431	24

Ijazah (1).*bmp*	1750x1275	24
Ijazah (2).*bmp*	842x595	24
Ijazah (3).*bmp*	967x647	24

Tabel 3. Hasil Pengujian

Nama File Citra	Pixel Setelah Di Enkripsi	Pixel Setelah Di Dekripsi	Bit Depth Setelah Di Enkripsi	Bit Depth Setelah Di Dekripsi
Dian pelangi.*bmp*	427x640	427x640	24	32
Dewi sandra.*bmp*	224x225	224x225	24	32
Hijau.*bmp*	650x431	650x431	24	32
Ijazah (1).*bmp*	1750x1275	1750x1275	24	32
Ijazah (2).*bmp*	842x595	842x595	24	32
Ijazah (3).*bmp*	967x647	967x647	24	32
Ungu.*bmp*	1024x768	1024x768	24	32

Berdasarkan hasil pengujian pada proses enkripsi dan dekripsi bit gambar awal (citra asli) yang bernilai 24 akan diubah menjadi 32 agar menghasilkan kualitas gambar yang maksimal. Maka dari itu nilai Bit Depth nya berbeda antara citra asli dan citra setelah di dekripsi.

KESIMPULAN

Dari penelitian dan hasil pengujian, Algoritma Kriptografi *TEA* aman digunakan untuk enkripsi dan dekripsi, karena jumlah round serta panjang kuncinya yang lebih kuat, serta tidak membutuhkan S-Box dan P-Box dalam proses enkripsi dan dekripsi. Di samping itu Algoritma *TEA* dengan 32 *round* sangat cocok untuk sistem keamanan file citra yang mengandalkan kecepatan proses yang optimal. Semakin besar ukuran citra semakin lama proses yang dilakukan untuk enkripsi dan dekripsinya. Citra asli dengan citra setelah didekripsikan akan berubah nilai Bit Depthnya.

REFERENSI

Aditya, Stefanus, dkk. "Implementasi Algoritma *Rijndael* Untuk Enkripsi dan Dekripsi Citra Digital". Journal Teknik Informatika. (15-16 Juni 2012). 1907-5052

Andem, Vikram Reddy. 2003. *A Cryptanalysis Of The Tiny Encryption Algorithm* [Thesis]. University Of Albama : Departement Of Computer Science.

Ariyus, Dony, 2008. *Pengantar Ilmu Kriptografi*, Penerbit Andi.Jakarta

Hidayat, A, 2008. "Kriptografi dan Stenografi Menggunakan Algoritma Vigenere dan TEA (Tiny Encryptin Algorithm)", Journal Matematika, Universitas Padjajaran, Bandung.

Komputer, Wahana. 2010. *The Best Encryption Tools.* Elex Media Komputindo. Jakarta.

Kurniawan, Yusuf. 2004. *Kriptografi Keamanan Internet dan Jaringan Komunikasi,* Informatika. Bandung.

Munir, Rinaldi, 2006. *Kriptografi,* Informatika. Bandung.

Putra, Darma, 2010. *Pengolahan Citra Digital.* Andi. Yogyakarta.

Williams, Derek. 2008. "The Tiny Encryption Algorithm (TEA)". Journal Columbus State University

William, K. 2000. "Studi Mengenai Tiny Encryption Algorithm (Tea)". Journal Teknik Informatika, ITB, Bandung