Penerapan Algoritma Secure Hash Algorithm (SHA) Keamanan Pada Citra

[1)]Hermansyah Sembiring
STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
E-Mail : hermansyah.sembiring@gmail.com

[2)]Fuzy Yustika Manik
STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
E-Mail : fuzy.yustika.manik@gmail.com

[3)]Tengkuzaidah
STMIK Kaputama, Jl. Veteran No. 4A-9A, Binjai, Sumatera Utara, Indonesia
E-Mail : tengkuzadiah@yahoo.com

ABSTRACT
Current information exchange is not only in the form of text, it can be an image or video. All of that can be done using computer network technology in the form of the internet. Through an internet connection, you can connect with many people. Cryptography aims for messages or images that cannot be seen by other parties who have no interest in the information. Messages or secured images can be data stored in secure computer memory or sent through computer networks. And can protect the confidentiality of images from various threats that arise. To maintain data security, SHA-256 is used when transforming data bytes into string hashes. The system built has been able to implement the SHA 256 algorithm which can change the original image file in the form of unknown files and encrypted images that are difficult to know the original image, unless read using an application that has been built. the image when encrypted and used again at the time of decryption with as many as 45 characters and may be in the form of numbers or letters. The SHA 256 process when encrypted in only one round, can be played with decryption.

Keywords: Secure Hash Algorithm, Security Algorithm, on Image.

PENDAHULUAN

Pertukaran informasi saat ini tidak hanya berupa teks, bisa juga berupa citra. Semua itu dapat dilakukan dengan menggunakan koneksi internet. Melalui koneksi internet, semua dapat terhubung dengan banyak orang.

Adanya koneksi internet ini semakin memudahkan seorang untuk melakukan pengambilan informasi rahasia citra yang terkirim melalui internet maupun yang tersimpan dalam media penyimpanan. Untuk itu diperlukan suatu usaha keamanan yang ketat supaya informasi citra digital tidak dibaca dan mengamankan keaslian citra tersebut agar tidak mudah diubaholeh orang yang tidak bertanggung jawab.

Kriptografi memegang peran penting dalam membangun keamanan citra. Kriptografi bertujuan agar citra tidak dapat dilihat oleh orang yang tidak berhak sehingga informasi baik yang disimpan dalam komputer aman maupun yang dikirim melalui koneksi internet. Serta dapat melindungi kerahasian citra dari berbagai ancaman yang muncul. Algoritma kriptografi yang dapat diterapkan untuk mengamankan citra adalah algoritma Secure Hash Algorithm (SHA).

Adanya koneksi internet ini semakin memudahkan seorang untuk melakukan pengambilan informasi rahasia citra yang terkirim melalui internet maupun yang tersimpan dalam media penyimpanan. Oleh sebab itu diperlukan suatu usaha keamanan yang ketat supaya informasi citra digitaltidak dibaca dan mengamankan keaslian citra tersebut agar tidak mudah diubaholeh orang yang tidak bertanggung jawab.

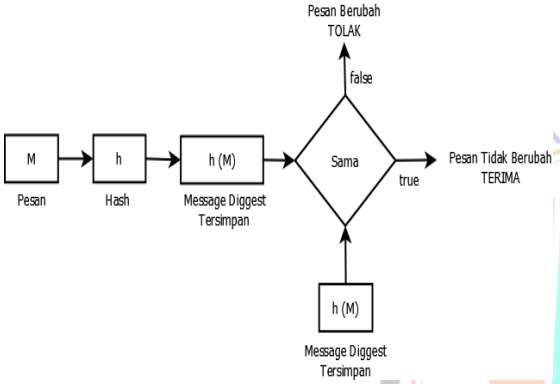
Kriptografi memegang peran penting dalam membangun keamanan citra [1][2]. Kriptografi bertujuan agar citra tidak dapat dilihat oleh orang yang tidak berhak sehingga informasi baik yang disimpan dalam komputer aman maupun yang dikirim melalui koneksi internet. Serta dapat melindungi kerahasian citra dari berbagai ancaman yang muncul. Algoritma kriptografi yang dapat diterapkan untuk mengamankan citra adalah algoritma *Secure Hash Algorithm* (SHA).

LANDASAN TEORI

2.1 Fungsi Hash

Fungsi hash adalah sebuah fungsi yang masukkannya adalah sebuah pesan ukuran sebuah sidik pesan (*message fingerprint*). Sidik pesan sering juga disebut (*message digest*) [3]. Fungsi hash dapat digunakan untuk mewujudkan beberapa layanan keamanan jaringan misalnya untuk keutuhan data dan otentikasi pesan.

Pengiriman pesan dan penerima pesan dapat digunakan untuk mewujudkan layanan keutuhan data [4]. Misalnya *M* merupakan pesan dan *h* adalah fungsi hash, maka *y= h(M)* disebut dengan sidik pesan *x* atau sering juga disebut dengan *message digest*. Sebuah message digest umumnya berukuran yaitu sekitar 512 bit.



Gambar 1. Pengujian Keutuhan Pesan Dengan Fungsi Hash
Sumber : Sadikin (2012, h. 310)

Dengan menggunakan sebuah fungsi *hash h*, sebelum pesan *M* disebarkan/ dikirimkan sebuah *message digesty lama= h(M)* disimpan sebagai acuan. Misalnya didapatkan kembali *M'* setelah disebarkan/ dikirim apabila ingin menguji *M = M'* hitung kembali *message digest* baru *ybaru= h(M)* disimpulkan pesan tidak berubah bila *ylama = ybaru*.

$$H^{(i)}=H^{(i+1)} + CM(i)(H^{(i-1)}) \quad \text{.....(2.1)}$$

dengan keterangan :
C = fungsi kompresi dari sha-256
+ = operasi penjumlahan modulo 2^{32}
H= hash dari *M*

Fungsi dari SHA-256 mengoperasikan 512 bit blok dan sebuah blok yang terdiri dari 256 bit *intermediate hash value*. *Intermediate hash value* adalah blok yang terdiri dari 256 bit yang dienkripsi menggunakan blok pesan sebagai kunci. Karena itu ada dua komponen utama yang harus dideskripsikan yaitu fungsi kompresi SHA-256 dan *message schedule* dari SHA-256.

2.2 *Secure Hash Algorithm*256 (SHA)

SHA-256 dirancang oleh *The National Institute of Standards and Technology* (NIST) pada tahun 2002. SHA-256 menghasilkan *message digest* dengan panjang 256 bit. SHA-256 merupakan salah satu fungsi *hash* satu arah, karena tidak mungkin menemukan pesan dari *message digest* yang dihasilkan [5][6].

Langkah-langkah pembuatan *message digiset* dengan SHA-256 secara garis besar sebagai berikut :

1. Penambahan bit-bit pengganjalan (*padding bits*).
2. Penambahan nilai panjang pesan semula.
3. Insialisasi nilai *hash* awal.
4. Pengelohan pesan dalam blok berukuran 512 bit.

SHA digunakan untuk menghitung *message diggest* dari pesan atau file data yang disediakan sebagai input. Pesan atau file dianggap sebagai kumpulan bit-bit. Panjang dari pesan adalah banyaknya bit didalam pesan (Pesan kosong memiliki panjang 0).

Jika banyaknya bit di dalam pesan merupakan kelipatan 8, untuk memudahkan pembacaan dapat ditampilkan dalam format hexadecimal.

Tujuan dari message padding adalah membuat panjang total dari isi pesan menjadi kelipatan 512 bit. SHA secara sekuensial memproses blok 512 bit ketika menghitung message diggest.

Pada message padding, tambahkan satu buah "1" , diikuti oleh m buah "0" diikuti oleh 64 bit integer pada akhir pesan untuk menghasilkan pesan dengan panjang 512 * n. 64 bit integer tersebut adalah panjang dari pesan asli sebelum message padding [7].

Proses yang digunakan dalam SHA-256 antara lain :

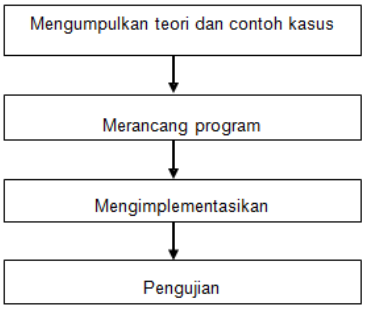
1. $Ch(x, y, z)=(x \wedge y) + (\sim x \wedge z)$
2. $Maj(x, y, z)=(x \wedge y) + (x \wedge z) + (y \wedge z)$
3. $\Sigma 0(x)=ROTR^{2}(x) + ROTR^{13}(x) + ROTR^{22}(x)$
4. $\Sigma 1(x)=ROTR^{6}(x) + ROTR^{11}(x) + ROTR^{25}(x)$
5. $\sigma 0(x)= ROTR^{7}(x) + ROTR^{18}(x) + SHR^{3}(x)$
6. $\sigma 1(x)= ROTR^{17}(x) + ROTR^{19}(x) + SHR^{10}(x)$

Keamanan pada citra menggunakan algoritma *Secure Hash Algorithm* (SHA) yang berguna untuk menjaga kerahasiaan citra pada saat dienkripsi dengan kunci yang diproses menggunakan algoritma SHA256.

ANALISIS DAN PERANCANGAN

3.1 Metodologi Penelitian

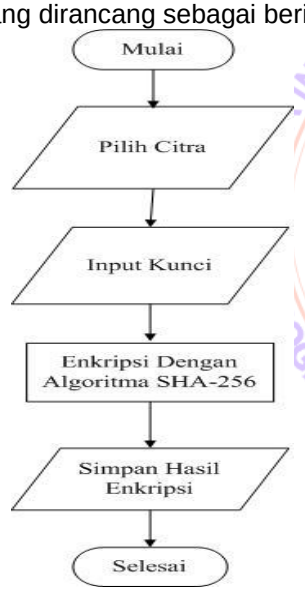
Beberapa langkah yang akan dilakukan dalam menyelasaikan masalah perancangan aplikasi enkripsi dan dekripsi pada citra (PNG) menggunakan algoritma *Secure Hash Algorithm* (SHA) terdiri dari tahapan sebagai berikut :



Gambar 2. Metodologi Penelitian

3.2 Flowchart Enkripsi

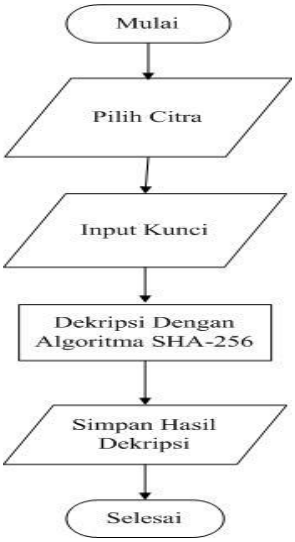
Adapun bentuk perancangan *flowchart* proses enkripsi yang dirancang sebagai berikut :



Gambar 3. Flowchart Enkripsi

3.3 Flowchart Dekripsi

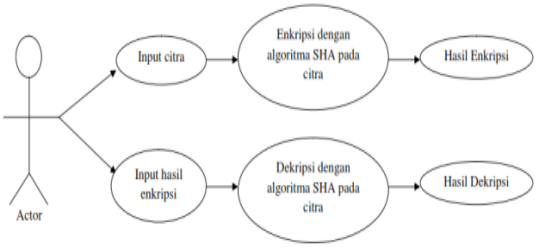
Adapun bentuk perancangan proses kerja dekripsi yang dirancang sebagai berikut :



Gambar 4. Flowchart Dekripsi

3.4 Use Case

Diagram *use case* digunakan untuk memberikan gambaran kebutuhan perangkat lunak. Use case diagram yang memiliki proses enkripsi dan dekripsi pada citra. Use case memilih penyampaian suatu informasi dengan citra kepada penerima.



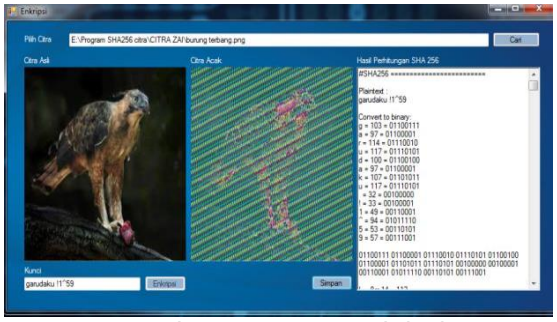
Gambar 5. Diagram Use Case

IMPLEMENTASI

Implementasi terhadap program dilakukan dengan menggunakan Visual Basic Net 2010 yang sudah ada ketika menginstall visual studio 2010. Visual Studio 2010 digunakan untuk membuat aplikasi didalam komputer dan menggunakan citra berformat PNG dan BMP.

4.1. Proses Enkripsi pada Citra

Pada proses ini adalah cara mengenkripsi file citra, dengan menekan tombol cari file citra yang akan dienkripsi, lalu akan mucul citra asli serta masukkan kunci, lalu tekan tombol enkripsi tunggu sebentar sampai hasil citra enkripsi muncul di citra acak, kemudian akan mucul juga perhitungan SHA 256 setelah itu simpan seperti tampilan di bawah ini :



Gambar 6. Proses Enkripsi

4.2. Pengujian Proses Dekripsi

Pada proses ini cara mendekripsikan (mengembalikan ke citra aslinya), denggan menekan tombol cari untuk file citra yang akan didekripsi ambil citra yang sudah di enkripsi lalu masukkan kunci yang sama dengan enkripsi klik tombol dekripsi kemudian akan mucul citra asli di citra acak serta perhitungan SHA 256 setalah itu simpan seperti di bawah ini.

Gambar 7. Proses Dekripsi

KESIMPULAN

Sistem yang dibangun telah dapat melakukan penerapan algoritma SHA 256 yang dapat mengubah file citra yang asli dalam bentuk file yang tidak dikenal dan citra yang telah di enkripsi tidak dapat diketahui, kecuali dibaca menggunakan aplikasi yang telah dibangun. Kunci yang dipakai untuk mengamankan citra pada saat dienkripsi dan di pakai lagi pada saat dekripsi dengan sebanyak 45 karakter serta boleh berbentuk angka maupun huruf.

DAFTAR PUSTAKA

[1] T. Limbong, “Pengujian kriptografi klasik caesar chipper menggunakan matlab,” no. September 2015, 2017.

[2] D. Ariyus, “PENGANTAR ILMU KRIPTOGRAFI Teori Analisis Dan Informasi, FI,” *Yogyakarta CV ANDI OFFSET*, 2008.

[3] R. Sadikin, “Kriptografi untuk keamanan jaringan,” *Penerbit Andi, Yogyakarta*, 2012.

[4] “Kriptografi untuk keamanan jaringan dan implementasinya dalam bahasa Jawa / Rifki Sadikin,” p. 2012, 2012.

[5] I. Wibowo, B. Susanto, and J. Karel T, “Penerapan algoritma Kriptografi Asimetris RSA untuk keamanan data di Oracle,” *J. Inform.*, vol. 5, no. 1, 2009.

[6] T. Sutabri, *Analisa Sistem Informasi*. Yogyakarta: Andi, 2012.

[7] R. Munir, “Kriptografi,” *Inform. Bandung*, 2006.